of regular differential chains that is equivalent to the input system (in this paper we can assume, with no loss of generality, that the output comprises a single system). The concept of a regular differential chain is a generalisation of the concept of a characteristic set. Both concepts are defined with respect to some ranking. The Rosenfeld–Gröbner algorithm was implemented in 1996 in the diffalg package of the MAPLE computer algebra software. Starting from MAPLE 14, it is replaced by the MAPLE Differential Algebra package, which relies on the BLAD libraries [9]. Differential elimination was applied [10] 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 parameters optimisation scheme. The overall method was implemented over the BLAD libraries [11].

Recently, we proposed a novel method for optimising the parameters [12–16], by using differential elimination. Differential elimination was used in previously [10, 17] in the context of a system identification based on physical laws. In general, the previous methods required much more computation time, making them less suitable to systems biology applications. In our method, we use part of a technique from a previous study [12], in which differential elimination is introduced into the parameter optimisation in a model including unmonitored variables. Instead of using differential elimination for estimating the initial values for the following parameter optimisation, as done in the previous study [12], the equations derived by differential elimination are directly introduced as the constraints into the objective function for the parameter optimisation. In previous reports [13–15], we validated the effectiveness of the constraint introduction in the models of differential equations to improve the estimation accuracy, in comparison with previous methods. Recently, we also designed a symbolic computation technique for analysing the complex model to improve the computational time, to accompany the constraint introduction [16].

In this review, we describe the methodology and the applications of our new method. In the methodological section, we will briefly explain the differential elimination [12] and the following simplification of the equivalent system derived by differential elimination into a reduced system [16]. We then outline the introduction of the system obtained by differential elimination into the objective function in the numerical optimisation method [13–15]. In particular, we present the implementations of the symbolic computation parts in Maple and the algorithms of the numeric computations in two appendices. In the application section, we will illustrate the validity of our method, using two models chosen from representative kinetic models for biological phenomena [18]. One is a simple model of three variables, analogous to a molecular reaction cascade, such as phosphorylation in signal transduction, and the other is a negative-feedback model of three variables with oscillation, analogous to an oscillatory response, such as Mitogen-activated protein kinase (MAPK) signalling pathways and circadian rhythms [18–20]. In the former model, the entire framework for introducing differential elimination is illustrated by the same model as that previously reported [13], but with larger different parameter values. In the latter model, the procedure to simplify equations by symbolic computation [16] is newly illustrated, to evaluate the reduction of the complexity of the equivalent system obtained by pure differential elimination. Finally, the merits and pitfalls of our method are further discussed, in terms of its limitations and extensions.

2 Methods

First, we will describe the symbolic parts, differential elimination and simplification, in our method. Second, we will define the idea of the introduction of new constraints by symbolic computation into the objective function in numerical optimisation. Finally, we will briefly describe the numerical part of the parameter optimisation techniques. In addition, we present the implementation in Maple for the symbolic part and the algorithms for the numeric part in Appendices 1 and 2, respectively.

2.1 Differential elimination

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. After explaining differential elimination, we will briefly describe the introduction of the constraints. Here, we provide an example of differential elimination, as shown below, according to Boulier [12].

Assume a model of two variables, x_1 and x_2, which is described by the following system of parametric ordinary differential equations

$$\begin{cases} \dot{x}_1 = -k_{12}x_1 + k_{21}x_2 - \dfrac{V_e x_1}{k_e + x_1} \\ \dot{x}_2 = k_{12}x_1 - k_{21}x_2 \end{cases} \tag{1}$$

where k_{12}, k_{21}, k_e and V_e are some constants. Here, one molecule, x_1, is assumed to be measured, and the binding between x_1 and x_2 and the degradation of x_1 are assumed to occur according to mass-action kinetics and Michaelis–Menten kinetics, respectively. The differential elimination by the Rosenfeld–Gröbner algorithm then produces the following two equations equivalent to the above system

$$\begin{cases} C_{1,t} = \dot{x}_1(k_e + x_1) + k_{21}x_1^2 + (k_{12} + V_e)x_1 \\ \qquad -k_{21}(k_e + x_1)x_2 = 0 \\ C_{2,t} = \ddot{x}_1(x_1 + k_e)^2 + (k_{12} + k_{21})\dot{x}_1(x_1 + k_e)^2 \\ \qquad +V_e\dot{x}_1 k_e + k_{21}V_e x_1(x_1 + k_e) = 0 \end{cases} \tag{2}$$

When we define the left sides of the above system as $C_{1,t}$ and $C_{2,t}$, $C_{2,t}$ is composed of x_1, its derivatives, and the parameters obtained by eliminating x_2, and $C_{1,t}$ is composed of x_1, its derivatives, the parameters and x_2. Note that x_2 in $C_{1,t}$ can be obtained analytically or estimated numerically by x_1, the parameters and the initial value of x_2. Then, the values of $C_{1,t}$ and $C_{2,t}$ can be calculated, if we have time series data of x_1, and they become zero, if all of the parameters were exactly estimated. Thus, $C_{1,t}$ and $C_{2,t}$ can be regarded as an objective function that expresses the difference between the monitored and estimated data.

2.2 Simplification of the equivalent system

In general, reducing the evaluation complexity (additions, multiplications) is a difficult problem and requires a large number of computer operations (a.k.a. a high algorithmic complexity). Moreover, the evaluation complexity of the

doi: 10.1049/iet-syb.2010.0051

Rosenfeld–Gröbner output tends to be exponential, in terms of the evaluation complexity of the input. Consequently, before directly applying techniques such as factorisation, Horner schemes, common subexpression detection, and so on, for reducing the evaluation complexity, one should try to use the existing knowledge available for the initial ODE system. Here, as an example, we will demonstrate the simplification for a preprocessing step [16], which speeds up the evaluation of $C_{1,t} + C_{2,t}$, as shown below.

The expressions of $C_{1,t}$ and $C_{2,t}$ in (2) are not the expressions originally computed by the Rosenfeld–Gröbner algorithm. Indeed, the Rosenfeld–Gröbner algorithm outputs expand expressions. Thus, using the Rosenfeld–Gröbner outputs, one has to evaluate the following expression $\bar{C}_{\mathrm{DE}}(= C_{1,t} + C_{2,t})$

$$\begin{aligned}\bar{C}_{\mathrm{DE}} = &|-k_{21}x_2\,k_e - k_{21}x_2x_1 + \dot{x}_1\,k_e + \dot{x}_1x_1 \\ &+ k_{12}k_ex_1 + k_{12}x_1^2 + V_ex_1| \\ &+ |k_{21}k_eV_ex_1 + 2k_ek_{12}x_1\dot{x}_1 + 2k_{21}k_e\dot{x}_1x_1 \\ &+ 2k_e\ddot{x}_1x_1 + k_{12}\dot{x}_1k_e^2 + k_eV_e\dot{x}_1 \\ &+ k_{12}x_1^2\dot{x}_1 + k_{21}k_e^2\dot{x}_1 + k_{21}x_1^2\dot{x}_1 \\ &+ k_{21}x_1^2V_e + \ddot{x}_1\,k_e^2 + \ddot{x}_1x_1^2| \end{aligned} \tag{3}$$

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

Since the expressions of $C_{1,t}$ and $C_{2,t}$ were computed from an ODE system involving the denominator $k_e + x_1$, originating from a Michaelis–Menten factor, the expression $k_e + x_1$ can probably be factorised. Introducing a new variable, $d_e = k_e + x_1$, and applying the substitution $k_e \rightarrow (d_e - x_1)$ in the previous expression of $\bar{C}_{\mathrm{DE}}$, yields

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

which requires 9 additions + 21 multiplications. Note that the last expression of $\bar{C}_{\mathrm{DE}}$ does not involve k_e anymore, which shows that the variable k_e only appears in $\bar{C}_{\mathrm{DE}}$ in the term $k_e + x_1$. This trick with the denominators has divided the number of operations by 2. For more complex systems, the benefit can be much greater. It is worth noting that the trick works 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 $k_i \rightarrow (f_i - d_i)$.

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

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

which requires 9 additions + 12 multiplications.

To finish, extra simplifications can be achieved by using the optimise command of the optimise package in the Computer Algebra software MAPLE. This last command tries to recognise common expressions to compute common subexpressions only once. This command is not very costly, as it is based on easy heuristics. In our case, it yields the sequence of the four following commands

$$\begin{aligned} t_7 &= |V_ex_1 - (k_{21}x_2 - \dot{x}_1 - k_{12}x_1)d_e| \\ t_8 &= V_e\dot{x}_1 \\ t_{19} &= |-t_8x_1 + (k_{21}V_ex_1 + t_8 + (\ddot{x}_1 + (k_{12} + k_{21})\dot{x}_1)d_e)d_e| \\ \bar{C}_{\mathrm{DE}} &= t_7 + t_{19} \end{aligned}$$

which need 9 additions + 11 multiplications + 4 assignments. The benefit here is only one multiplication, but it can be greater on bigger systems. All of the previous operations can be automated in MAPLE, and the C code is obtained by the C command of the optimise package (see also Appendix 1).

2.3 Introduction of constraints by differential elimination

The objective function in this study is composed of two terms [13–16]: one is the standard error function between the estimated and measured data, and the other is the constraints obtained by differential elimination. The error function is defined as follows: cuppose that $x_{i,t}^{\mathrm{c}}$ is the time-course data at time t of x_i calculated by using the estimated parameter values, and $x_{i,t}^{\mathrm{m}}$ represents the measured data at time t. The sum of the absolute values of the relative errors between $x_{i,t}^{\mathrm{c}}$ and $x_{i,t}^{\mathrm{m}}$ gives the averaged relative error over the numbers of monitored variables and time points, E, as a standard error function, that is

$$E = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|\frac{x_{i,t}^{\mathrm{c}} - x_{i,t}^{\mathrm{m}}}{x_{i,t}^{\mathrm{m}}}\right| \tag{6}$$

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

Next, we define the constraints obtained by differential elimination. In general, differential elimination rewrites the original system of differential equations into an equivalent system, which means that the number of equations is equal in both systems (both systems have the same solutions). Thus, we express the constraint by differential elimination, the DE constraint (C_{DE}), as the average of the linear combination of the equations in the equivalent system over the number of equivalent equations and time points, as follows

$$C_{\mathrm{DE}} = \frac{1}{LT}\sum_{l=1}^{L}\sum_{j=1}^{T}|C_{l,j}| \tag{7}$$

where L and T are the number of equivalent equations and time points, respectively.

Finally, we introduce C_{DE} into the objective function, F, in combination with E, as

$$F = \alpha E + (1 - \alpha)C_{\mathrm{DE}} \tag{8}$$

where α ($0 \leq \alpha \leq 1$) is the weight of two functions, and is approximately estimated by the slope of the Pareto-optimal

solutions [21] (see details in Figs. 2*b* and 5*b*) for E and C_{DE}, and then is manually modified (see also Section 2.4). As a result, our computational task is to determine a set of parameter values that minimise F.

2.4 Optimisation technique

The new objective function can be generally introduced into any type of optimisation techniques. Indeed, we previously illustrated the introduction into two evolutionary optimisation techniques [genetic algorithm (GA) and particle swarm optimisation (PSO)] and one Newton type of optimisation technique, and the new objective function worked well in the evolutionary optimisation techniques [14]. Here, the GA is used.

GA is a well-known parameter optimisation framework, which was inspired by the evolutionary process of biology [22, 23]. We apply an efficient computational technique based on RCGAs as a non-linear numerical optimisation method, by the combination of unimodal normal distribution crossover (UNDX) and minimal generation gap (MGG) [24, 25]. The generation alternations are repeated until either the value of the objective function E becomes less than a given threshold (we call this threshold the error allowance on real-coded genetic algorithm (RCGAs)) or the number of generation-alternation iterations reaches a given threshold of maximum generation counts.

In the parameter optimisation, two thresholds are set to stop the optimisation: the error function over time points, E, and the number of generations per optimisation. In this study, we perform the optimisation 200 times for the cascade model and 100 times for the negative feedback model. In each optimisation, the generation number is set to 2000, and the thresholds of E for the simple cascade model and the negative-feedback model with oscillation are set to 0.02 and 0.5, respectively. As a result, the numbers of successes for 200 trials are 200 in both methods for the simple cascade model, and 94 and 83 in the standard method and our method for the negative-feedback model, respectively (see also Appendix 2).

Note that only E is used to stop the optimisation by GA, but C_{DE} is not. This indicates that the estimated value of α obtained by the Pareto-optimal solution is large in this procedure: E is optimised as the GA operation progresses, whereas C_{DE} is calculated by the estimated parameter values without optimisation. In other words, the C_{DE} for the most effective optimisation in F of (8) is smaller than the calculated C_{DE} corresponding to E in the Pareto-optimal solution plot, and therefore the value of α is expected to be small (see also Appendix 2).

2.5 Software: implementation of differential elimination and simplification

All of the symbolic computations for the differential elimination are performed using the diffalg package of MAPLE 10. In the performance of differential elimination, the ranking of variables was chosen as follows: $P(\text{Pool}) > x_3 > x_2 > x_1$ for the cascade model, and $X > Y_P > R_P$ for the negative feedback model. The following simplification of the polynomial equations derived by differential elimination is also performed by MAPLE 14. Finally, we generate the Java or C code from the simplified system, by using the Code Generation feature in MAPLE14 (see also Appendix 1).

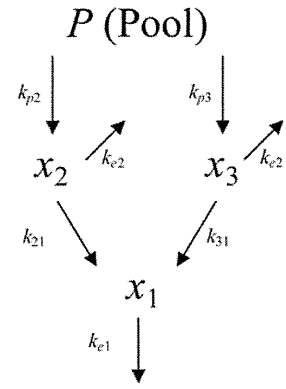


Fig. 1 *Schematic representation of a simple cascade model*
Model is composed of three variables, x_1, x_2 and x_3, with a pool, P. We assume that time series data for one of the variables, x_1, are obtained

3 Results and discussion

First, we will follow our procedure with a simple model, on the assumption that only one of the four variables in the model is measured. Second, we will illustrate the effectiveness of the symbolic technique for reducing the evaluation complexity in a negative-feedback model with oscillation. Finally, we will discuss the merits and pitfalls of our method, in terms of further method expansion.

3.1 Simple cascade model

A simple model of three molecules for cascade reactions [13], such as phosphorylation, is schematically depicted in Fig. 1, and the corresponding system of differential equations is expressed as follows

$$\begin{cases} \dfrac{dx_1}{dt} = k_{21}x_2 + k_{31}x_3 - k_{e1}x_1 \\ \dfrac{dx_2}{dt} = -k_{e2}x_2 + k_{p2}P - k_{21}x_2 \\ \dfrac{dx_3}{dt} = -k_{e2}x_3 + k_{p3}P - k_{31}x_3 \\ \dfrac{dP}{dt} = 0 \end{cases} \tag{9}$$

Here we assume that the time series of only one variable, x_1, can be observed.

For the above equations, the system equivalent to the original system was obtained by differential elimination, as follows

$$\begin{aligned} C_1 = \Big(&\frac{d^3x_1(t)}{dt^3} + (2k_{p3}k_{e2} + k_{e1} + k_{21} + k_{31})\frac{d^2x_1(t)}{dt^2} \\ &+ (k_{31}k_{p3}k_{e1} + 2k_{p3}k_{e2}k_{p3}k_{e1} + k_{21}k_{p3}k_{e1} + k_{21}k_{p3}k_{31} \\ &+ k_{21}k_{p3}k_{e2} + k_{e2}^2 + k_{31}k_{p3}k_{e2})\frac{dx_1(t)}{dt} + (k_{e2}^2k_{p3}k_{e1} \\ &+ k_{21}k_{p3}k_{31}k_{p3}k_{e1} + k_{31}k_{p3}k_{e2}k_{p3}k_{e1} + k_{21}k_{p3}k_{e2}k_{p3}k_{e1}) \\ &\times k_{p3}x_1(t)\Big)/(k_{p2}k_{p3}k_{31}k_{p3}k_{21} + k_{e2}k_{p3}k_{p2}k_{p3}k_{21} \\ &+ k_{p3}k_{p3}k_{21}k_{p3}k_{31} + k_{p3}k_{p3}k_{e2}k_{p3}k_{31}) - P \end{aligned}$$

IET Syst. Biol., 2011, Vol. 5, Iss. 5, pp. 281–292
doi: 10.1049/iet-syb.2010.0051

$$
\begin{aligned}
C_2 = \Big(& (-k_{p2}k_{21} - k_{p3}k_{31})\frac{d^3x_1(t)}{dt^3} \\
& - (k_{p2}k_{21}^2 - k_{p2}k_{e1}k_{21} - k_{31}^2k_{p3} - k_{e2}k_{p3}k_{31} \\
& - k_{e2}k_{p2}k_{21} - k_{e1}k_{p3}k_{31})\frac{d^2x_1(t)}{dt^2} \\
& - (k_{p2}k_{e1}k_{21}^2 - k_{31}^2k_{e1}k_{p3} - k_{e2}k_{p2}k_{21}k_{e1} \\
& - k_{e2}k_{e1}k_{p3}k_{31} - k_{31}^2k_{21}k_{p3} - k_{31}^2k_{p3}k_{e2} \\
& + k_{p3}k_{21}^2k_{31} + k_{e2}k_{21}k_{p3}k_{31})\frac{dx_1(t)}{dt} \\
& - (k_{e1}k_{31}^2k_{21}k_{p3} - k_{e1}k_{31}^2k_{p3}k_{e2} + k_{e1}k_{e2}k_{21}k_{p3}k_{31} \\
& + k_{p3}k_{21}^2k_{31}k_{e1})k_{p3}x_1(t))/(k_{31}(-k_{p2}k_{31}^2k_{21} + k_{p2}k_{31}k_{21}^2 \\
& + k_{e2}k_{p2}k_{21}^2 - k_{p2}k_{31}k_{21}k_{e2} - k_{31}^2k_{21}k_{p3} \\
& - k_{31}^2k_{p3}k_{e2} + k_{e2}k_{p2}k_{p3}k_{31} + k_{p3}k_{21}^2k_{31})\Big) - x_3(t)
\end{aligned}
$$

$$
\begin{aligned}
C_3 = \Big(& (-k_{p2}k_{21} - k_{p3}k_{31})\frac{d^3x_1(t)}{dt^3} \\
& + (k_{p2}k_{21}^2 + k_{p2}k_{e1}k_{21} + k_{31}^2k_{p3} + k_{e2}k_{p3}k_{31} \\
& + k_{e2}k_{p2}k_{21} + k_{e1}k_{p3}k_{31})\frac{d^2x_1(t)}{dt^2} \\
& + (k_{p2}k_{e1}k_{21}^2 + k_{p2}k_{31}k_{21}^2 + k_{e2}k_{p2}k_{21}^2 - k_{p2}k_{31}k_{21}k_{e2} \\
& + k_{31}^2k_{e1}k_{p3} + k_{e2}k_{p2}k_{21}k_{e1} + k_{e2}k_{e1}k_{p3}k_{31} \\
& - k_{p2}k_{31}^2k_{21})\frac{dx_1(t)}{dt} \\
& + (-k_{p2}k_{31}k_{21}k_{e2}k_{e1} - k_{p2}k_{31}^2k_{21}k_{e1} \\
& + k_{p2}k_{31}k_{e1}k_{21}^2 + k_{e2}k_{p2}k_{e1}k_{21}^2)x_1(t))/ \\
& (k_{21}(-k_{p2}k_{31}^2k_{21} + k_{p2}k_{31}k_{21}^2 + k_{e2}k_{p2}k_{21}^2 \\
& - k_{p2}k_{31}k_{21}k_{e2} - k_{31}^2k_{21}k_{p3} - k_{31}^2k_{p3}k_{e2} \\
& + k_{e2}k_{p2}k_{p3}k_{31} + k_{p3}k_{21}^2k_{31})\Big) - x_2(t)
\end{aligned}
$$

$$
\begin{aligned}
C_4 = & -\frac{d^4x_1(t)}{dt^4} + (-k_{e1} - k_{31} - 2k_{e2} - k_{21})\frac{d^3x_1(t)}{dt^3} \\
& + (-k_{21}k_{31} - k_{e2}k_{21} - k_{e1}k_{21} - k_{e2}^2 \\
& - k_{e1}k_{31} - k_{e2}k_{31} - 2k_{e1}k_{e2})\frac{dx_1^2(t)}{dt^2} \\
& + (-k_{21}k_{31}k_{e1} - k_{e2}k_{21}k_{e1} \\
& - k_{e2}k_{e1}k_{31} - k_{e1}k_{e2}^2)\frac{dx_1(t)}{dt}
\end{aligned} \tag{10}
$$

Since the above system was not so complicated, we directly introduced it into the objective function as the DE constraints.

According to the model in Fig. 1, the reference curve of one variable, x_1, was generated in Fig. 2*a*. Among the parameters in the model, the values of two parameters, k_{21} and k_{31}, were estimated, and the values of the remaining parameters were the same as those used in the simulation of Fig. 2*a*. The Pareto-optimal solution plot for the reference curve is shown in Fig. 2*b*. From the Pareto-optimal solution plot, α was estimated to be 0.999892, which was large in the present procedure, as described in Section 2.4. As for each contribution of E and C_{DE} in F of (8), the orders of αE and $(1-\alpha)C_{DE}$ are approximated to 10^0 and 10^2, respectively, according to (8) and the above value of α. Thus, although the α value is almost equal to 1, the DE constraint exerts a large influence on the parameter optimisation.

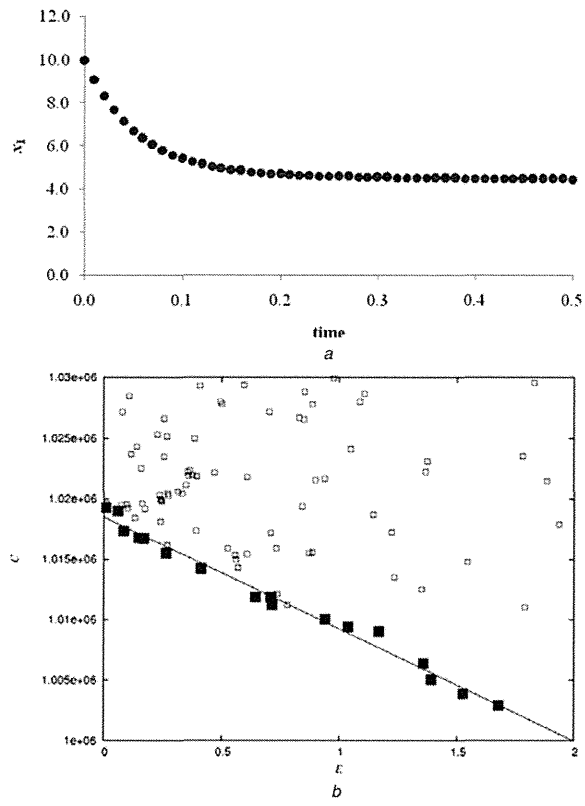


Fig. 2 *Reference curve simulated based on the cascade model and its Pareto-optimal solution plot*

a According to the kinetics of the model (9), a reference curve of one variable, x_1, was generated for $0 \leqq t \leqq 0.5$ with intervals of 0.01, under the following conditions: $x_1(0) = 10.0$, $x_2(0) = 20.0$, $x_3(0) = 30.0$, $P = 30.0$, $k_{21} = 10.0$, $k_{31} = 3.0$, $k_{p2} = 23.0$, $k_{p3} = 4.0$, $k_{e1} = 15.0$ and $k_{e2} = 4.0$

b The empty square (□) indicates the set of evaluated values, E and C. The filled square (■) shows the Pareto-optimal solutions, and the line represents the fitted line for Pareto-optimal solutions

The introduction of the DE constraint into the objective function was quite effective in the comparison with the distributions of the parameter values estimated with and without the DE constraint (Fig. 3). Indeed, the distribution of the estimated k_{21} and k_{31} values was highly concentrated around the correct values by the estimation with the DE constraint (Fig. 3*a*), whereas the distribution of the two parameter values was broader in the larger space by the estimation without the DE constraint (Fig. 3*b*). In other words, the values with the DE constraint were distributed in a relatively narrow range, whereas those without the DE constraint were distributed over a wide range. As a result, the parameter accuracy was improved by the new objective function with the introduction of the DE constraint, in the model of Fig. 1.

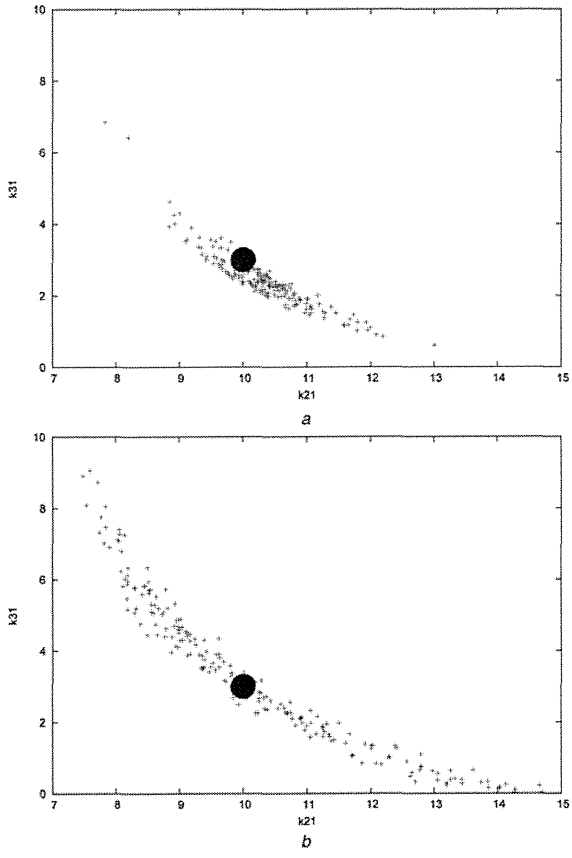


Fig. 3 *Comparison of parameter value distribution estimated for the reference curve in Fig. 2*

a With DE constraint
b Without DE constraint
The given values of k_{21} and k_{31} were 10.0 and 3.0, respectively

3.2 Feedback loop model with oscillation

A negative-feedback model with oscillation [18] is schematically depicted in Fig. 4, and the system of differential equations is expressed as follows

$$\begin{cases} \dfrac{dX}{dt} = k_0 + k_1 S - k_2 - X - k_2' R_P X \\ \dfrac{dY_P}{dt} = \dfrac{k_3 X(Y_T - Y_P)}{K_{m3} + Y_T - Y_P} - \dfrac{k_4 Y_P}{K_{m4} + Y_P} \\ \dfrac{dR_P}{dt} = \dfrac{k_5 Y_P(R_T - R_P)}{K_{m5} + R_T - R_P} - \dfrac{k_6 R_P}{K_{m6} + R_P} \end{cases} \quad (11)$$

Here we assume that the time series of three variables, X, Y_P and R_P, can be observed. For these equations, the system equivalent to the original system was obtained by differential elimination, but it was huge in size (7.4 MB in file size). Therefore the we simplified the equivalent system into a reduced system (0.1 MB, data not shown) [16] was performed. Then, the reduced system was introduced into the objective function as the DE constraints.

The above system was used to generate the reference curve for parameters optimisation, as shown in Fig. 5*a*. The task is to estimate the values of 6 parameters, k_1, k_{2p}, k_3, k_4, k_5 and k_6, among the 12 parameters in the

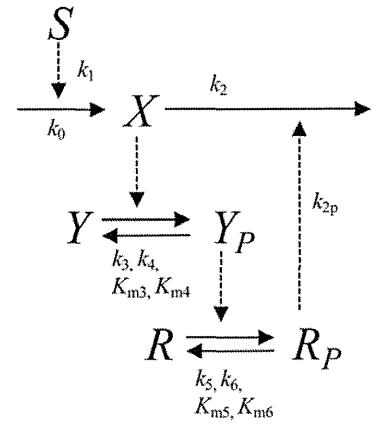


Fig. 4 *Schematic representation of the negative-feedback model*

The model is composed of three variables, x_1, x_2 and x_3, with a pool, P. We assume that time series data for one of the variables, x_1, are obtained

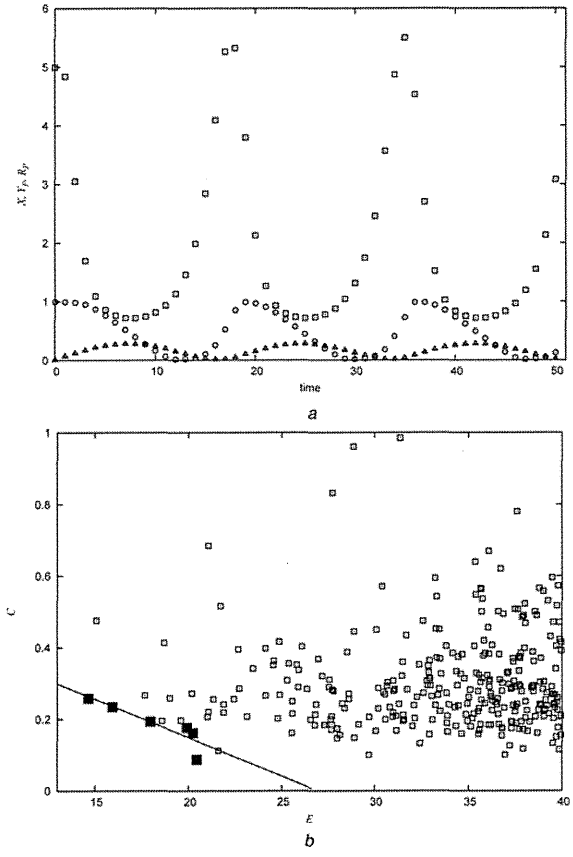


Fig. 5 *Reference curve simulated based on the negative-feedback model and its Pareto-optimal solution plot*

a According to the kinetics of the model (11), a reference curve of variables, X(○), Y_P(□) and R_P(△), was generated for $0 \leqq t \leqq 50$ with intervals of 1.0, under the following conditions: $X(0) = 5.0$, $Y_P(0) = 1.0$, $R_P(0) = 0.01$, $k_0 = 0.0$, $k_1 = 1.0$, $k_2 = 0.01$, $k_3 = 0.1$, $k_4 = 0.2$, $k_5 = 0.1$, $k_6 = 0.05$, $k_{2p} = 10.0$, $Y_T = R_T = 1.0$, $K_{m3} = K_{m4} = K_{m5} = K_{m6} = 0.01$ and $S = 2.0$
b The empty square (□) indicates the set of evaluated values, E and C. The filled square (■) shows Pareto-optimal solutions, and the line represents the fitted line for Pareto-optimal solutions

doi: 10.1049/iet-syb.2010.0051

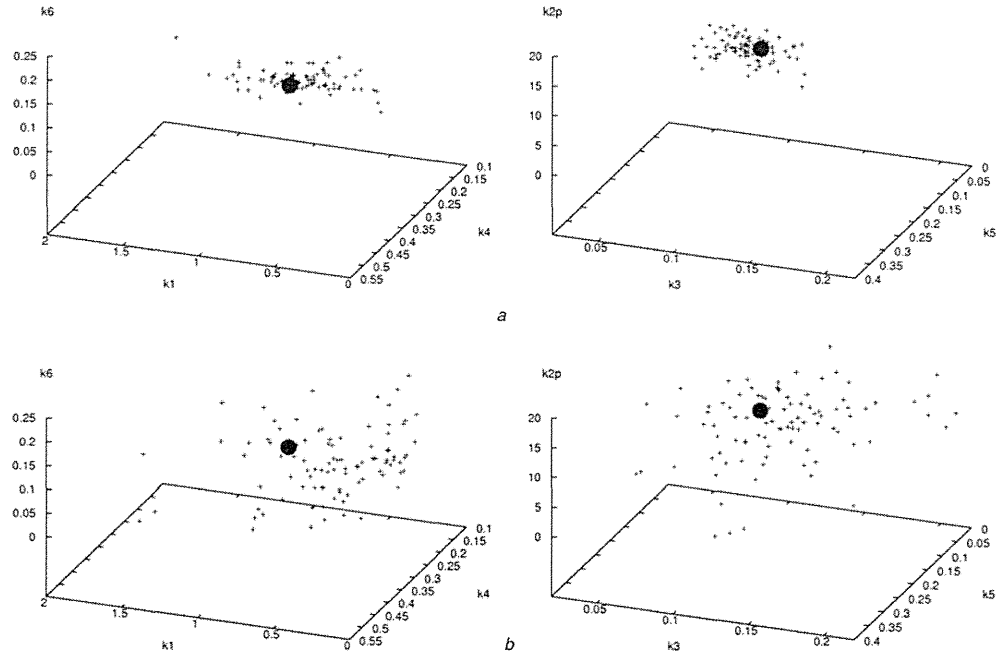


Fig. 6 *Comparison of parameter value clouds estimated by our method (a) and the standard method (b)*
The given values are as follows: $k_1 = 1.0$, $k_3 = 0.1$, $k_4 = 0.2$, $k_5 = 0.1$, $k_6 = 0.05$, $k_{2p} = 10.0$

model. The values of the remaining parameters were the same as those used in the simulation of Fig. 5*a*. This is because one can experimentally measure biochemical properties, such as the synthesis and degradation rates of a protein (k_0 and k_2) or the Michaelis–Menten constants (K_{m3}, K_{m4}, K_{m5} and K_{m6}). The Pareto-optimal solution plot for the reference curve is shown in Fig. 5*b*, andα was estimated to be 0.002665. Since the value of α was very small and was estimated to be large in the present procedure, α was set to 0.0 in this case. Thus, E was used only as the threshold for the stopping rule in GA, and the objective function for parameter optimisation was composed of only C_{DE}. Note that the magnitude of the α value depends on the relative magnitudes between E and C_{DE} simulated in Figs. 2*b* and 5*b*. Indeed, the order of $(1 - \alpha)C_{DE}$ is larger than that of αE, indicating the strong influence of the DE constraints on the parameter optimisation, as in Section 3.1.

Our method was also highly effective for the negative-feedback model with oscillation, in terms of improving the estimation accuracy (Fig. 6).The clouds of the six parameter values were clearly smaller in the estimation with the DE constraint (Fig. 6*a*), in comparison with the parameter clouds in the estimation with only the error function (Fig. 6*b*). Note that the objective function in our method comprises only the DE constraint in this case, because of the small influence of the error function on the objective function through the value of α, in the estimation by the slope of the Pareto-optimal solutions. Interestingly, this means that the error function that measures the difference between the sample and the estimated data is not effective for parameter estimation. Instead of the error function, the new constraints including the derivatives also include the information on the curve form. Thus, the new constraints may be useful for estimating the parameter values, especially in models with characteristic curve forms, such as oscillation. Note that all of the variables are observed in the above model. Even in the complete set of data for all variables, it may be difficult to precisely estimate the parameter values in the oscillation model, since the error function estimates a similar degree of differences between the measured and simulated values, which frequently emerge at different time points.

We also compared the fitness between the curves with the parameters estimated with and without the DE constraint. Since the error range for the error function, E, was set to 50%, the fitness is also regarded as a measurement of the estimation accuracy in the negative-feedback model, in comparison with the case of 2% in the cascade model. The averages and the standard deviations of the three variables were plotted at each time point in Fig. 7. Since all of the parameters were attained with error values within 50%, most of the values of the reference curve at each point ranged within the average $\pm$ the standard deviation. However, as easily seen in the figure, the points calculated with the parameters estimated by our methods (Fig. 7*a*) fit better than those obtained by the standard method (Fig. 7*b*). Indeed, most of the averages at each time point in Fig. 7*a* were approximately the same values as those in the reference curve. In contrast, the averages were relatively different from the values of the reference curve, and in particular, the difference widened as the amplitude of the curve increased, in a comparison between the three curves. Thus, our method clearly improved the estimation accuracy, in terms of the fitness of the estimated curve.

3.3 Limits and further extensions

As expected, the new objective function requires more computational time, in comparison with an objective function with only a standard error function, because of the increased number of functions in the DE constraint. Indeed, the computational time of our method was longer than that

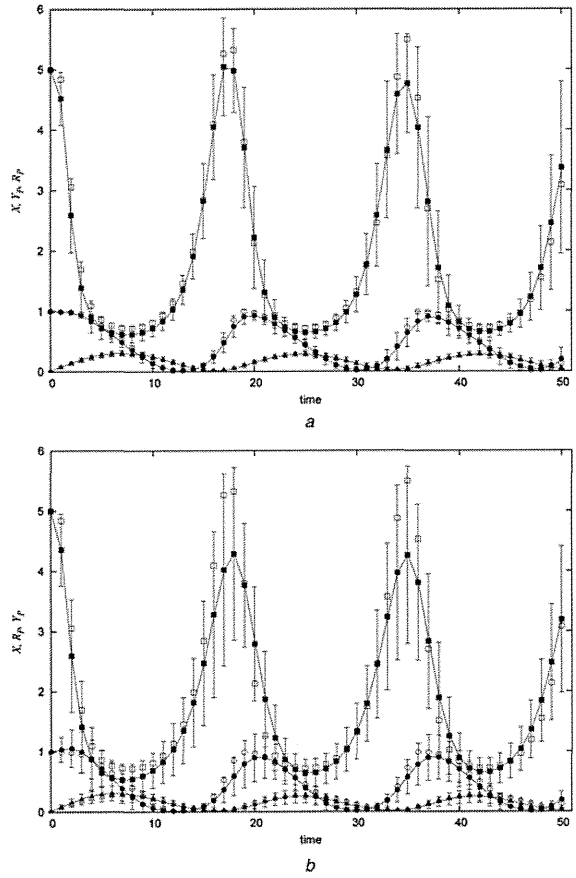


Fig. 7 *Comparison of fitness estimated by the standard method and our method*

a With DE constraint
b Without DE constraint
The empty circles (○), squares (□) and triangles (△) indicate the reference curves for X, Y and R_P, respectively. The filled circles (●), squares (■) and triangles (▲) indicate the averaged values of the fitted curves. The error bars show standard deviations

of the standard method in Models 1 and 2; the respective computational times for the standard method and our method were 0.0011 and 0.015 h in Model 1, and 0.8 and 4.9 h in Model 2 [128 CPUs of Intel(R) Xeon(R) X5550 2.67 GHz]. In addition to the computational time, a pitfall of our method is the model size or the equation size of the DE constraint. In the equivalent systems, the number of terms frequently increases, and this may make the application of our method to a complex or large model more difficult. Indeed, the parameters in the feedback model with the DE constraint could not be estimated without the simplification. The memory required for the DE constraint after the equation simplification by symbolic computation was reducedby 740-fold in the negative-feedback model with oscillation, as compared to that needed for the pure differential elimination. Thus, the simplification by symbolic computation [16] is prerequisite to the present procedure, especially for complex models.

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. As discussed in Section 3.2 of the feedback loop model, the DE constraints reflect well the form of the data curve. Indeed, the effects of lower order derivatives to fit the curve, such as the slope and the inflection point, can be geometrically interpreted, whereas those of higher order derivatives seem difficult to be intuitively interpreted. Although our method was useful, even for noisy data in a simple model [15], the estimated values of the higher order derivatives for noisy data may become more unstable. 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.

Note that the complexity of the DE constraints generally depends on the mathematical model and the corresponding system of differential equations. When the model and the corresponding system are simple, the DE constraint also has a simple form, and the computational time required by our method is reasonable. Indeed, we successfully analysed the cascade model of ten molecules within a reasonable time (data not shown). Since the modelling depends on biological knowledge, rather than the computational convenience, we should continuously reduce the computational time.

4 Conclusions

The introduction of the DE constraint, derived from the original system of differential equations into the objective function, clearly improved the parameter accuracy. The performance of the new method is illustrated by simple and complex models with small sizes, which are analogous to two of the molecular reactions responsible for biological phenomena. One of the features of the DE constraint is that it includes the derivatives of the original system for the model. Since the derivatives generally contain the curve form information of the measured time-series data, such as slope, extremal point and inflection point, the new objective function estimates the difference of not only the values but also the comprehensive forms between the measured and estimated data, whereas the standard 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 in parameter optimisation for improving the parameter accuracy.

As described in this review, symbolic computation played a central role in improving the estimation accuracy in our method, and a computer algebra system based on symbolic computation was used to implement it. The well-known Gröbner base, which was found by Buchberger [26, 27], is an underlying principle for symbolic computation, and for various fields of mathematics and computer science [28, 29]. Indeed, we have applied symbolic computation techniques to solve issues in systems biology [30, 31], as encouraged by stimulating discussions in the 'Algebraic Biology' conference series [32–35]. We would like to name the feasibility of symbolic computation implemented in the computer algebra system as 'Bruno force', after his discovery of the Gröbner base, in contrast to that of numeric computation depending on computer performance, which is frequently referred to as 'Brute force'. The present study is an example of the happy union between Brute force and 'Bruno force', and further progress is expected in

doi: 10.1049/iet-syb.2010.0051

various fields of systems biology by the amalgam of rational mathematical manipulation by 'Bruno force' and powerful numerical computation by Brute force.

5 Acknowledgments

We would like to dedicate this review to Prof. Bruno Buchberger, of Johannes Kepler University of Linz, Austria, for his continuous encouragement of our application of symbolic computation techniques to issues in systems biology. This work was partly 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), Japan.

6 References

1 Kitano, H.: 'System biology: a brief overview', *Science*, 2002, **295**, pp. 1662–1664
2 Nocedal, J., Wright, S.J.: 'Numerical optimization' (Springer-Verlag, New York, 1999)
3 Ritt, J.F.: 'Differential algebra' (Dover Publications Inc., New York, 1950)
4 Kolchin, E.R.: 'Differential algebra and algebraic groups' (Academic Press, New York, 1973)
5 Seidenberg, A.: 'An elimination theory for differential algebra' (Univ. California Publ. Math. (New Series), 1956), vol. 3), pp. 31–65
6 Wu, W.T.: 'On the foundation of algebraic differential geometry', *Mechanization Math. (Res. Preprints)*, 1989, **3**, pp. 2–27
7 Boulier, F., Lazard, D., Ollivier, F., Petitot, M.: 'Representation for the radical of a finitely generated differential ideal'. Proc. 1995 Int. Symp. on Symbolic and Algebraic Computation, ISSAC'95, New York, NY, USA, 1995, pp. 158–166
8 Boulier, F., Lazard, D., Ollivier, F., Petitot, M.: 'Computing representations for radicals of finitely generated differential ideals', *J. AAECC*, 2009, **20**, (1), pp. 73–121
9 Boulier, F.: 'The BLAD libraries', http://www.lifl.fr/~boulier/BLAD, 2004
10 Lilianne, D.-V., Ghislaine, J.-B., Celine, N.: 'System identifiability (symbolic computation) and parameter estimation (numerical computation)', *Numer. Algorithms*, 2003, **34**, pp. 282–292
11 Boulier, F., Denis-Vidal, L., Henin, T., Lemaire, F.: 'LÉPISME'. Proc. of the ICPSS Conf., 2004, http://hal.archives-ouvertes.fr/hal-00140368
12 Boulier, F.: 'Differential elimination and biological modelling' (Johann Radon Institute for Computational and Applied Mathematics (RICAM) Book Series, 2007), vol. 2), pp. 111–139
13 Nakatsui, M., Horimoto, K.: 'Parameter optimization in the network dynamics including unmeasured variables by the symbolic-numeric approach'. Proc. Third Int. Symp. on Optimization and Systems Biology (OSB'09), 2009, pp. 245–253
14 Nakatsui, M., Horimoto, K., Okamoto, M., Tokumoto, Y., Miyake, J.: 'Parameter optimization by using differential elimination: a general approach for introducing constraints into objective functions' (BMC Systems Biology), 2010, vol. 4 (suppl 2), p. 59
15 Nakatsui, M., Horimoto, K.: 'Improvement of estimation accuracy in parameter optimization by symbolic computation'. Proc. IEEE Multi-Conf. on Systems and Control, 2010, pp. 1720–1724
16 Nakatsui, M., Sedoglavic, A., Lemaire, F., Boulier, F., Urguplu, A., Horimoto, K.: 'A general procedure for the accurate parameter estimation in dynamic systems using new estimation errors', in Horimoto, K., Nakatsui, M., Popov, N. (Eds.): 'Algebraic and numeric biology' (*LNCS*, Springer, Heidelberg), in press
17 Ljung, L., Glad, T.: 'On global identifiability for arbitrary model parametrizations', *Automatica*, 1994, **30**, pp. 265–276
18 Tyson, J.J., Chen, C.K., Novák, B.: 'Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signalling pathways in the cell', *Curr. Opin. Cell Biol.*, 2003, **14**, (2), pp. 221–231
19 Kwon, Y.K., Cho, K.H.: 'Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics', *Bioinformatics*, 2008, **24**, (7), pp. 987–994
20 Novák, B., Tyson, J.J.: 'Design principles of biochemical oscillators', *Nat. Rev. Mol. Cell Biol.*, 2008, **9**, (12), pp. 981–991
21 Kung, H.T., Luccio, F., Preparata, F.P.: 'On finding the maxima of a set of vectors', *J. ACM*, 1975, **22**, pp. 469–476
22 Holland, J.H.: 'Adaptation in natural and artificial systems' (The University of Michigan Press, Ann Arbor, MI, 1975)
23 Goldberg, D.D.: 'Genetic algorithms in search, optimization and machine learning' (Addison-Wesley Longman Publishing Co. Inc., Boston, MA, 1989)
24 Ono, I., Kobayashi, S.: 'A real-coded genetic algorithm for function optimization using unimodal distribution crossover'. Proc. Seventh ICGA, 1997, pp. 249–253
25 Satoh, H., Ono, I., Kobayashi, S.: 'A new generation alternation model of genetic algorithm and its assessment', *J. Jpn. Soc. Artif. Intell.*, 1997, **15**, (2), pp. 743–744
26 Buchberger, B.: 'EinAlgorithmuszumAuffinden der Basiselemente des RestklassenringesnacheinemnulldimensionalenPolynomideals'. PhD thesis, University of Innsbruck, Austria, Math. Institute, Austria, 1966 English translation in [24]
27 Buchberger, B.: 'An algorithm for finding the basis elements in the residue class ring modulo a zero dimensional polynomial ideal', *J. Symbol. Comput. (Special Issue on Logic Math. Comp. Sci. Interact.)*, 2006, **41**, pp. 475–511
28 Cox, D.A., Little, J.B., O'Shea, D.: 'Using algebraic geometry' (Springer-Verlag, New York, 1998)
29 Buchberger, B., Affenzeller, M., Ferscha, A. *et al.* (Eds.): 'Hagenberg research' (Springer, Heidelberg, 2009)
30 Anai, H., Orii, S., Horimoto, K.: 'Symbolic-numeric estimation of parameters in biochemical models by quantifier elimination', *J. Bioinfo. Comput. Biol.*, 2006, **4**, pp. 1097–1117
31 Yoshida, H., Nakagawa, K., Anai, H., Horimoto, K.: 'An algebraic-numeric algorithm for the model selection in kinetic networks'. Proc. Tenth CASC 2007, 2007 (*LNCS*, **4770**), pp. 433–447
32 Anai, H., Horimoto, K. (Eds.): 'Algebraic biology – computer algebra in biology' (Universal Academy Press, Tokyo, 2005)
33 Anai, H., Horimoto, K., Kutsia, T. (Eds.): 'Algebraic biology' (*LNCS*, **4545**, Springer, Heidelberg, 2007)
34 Horimoto, K., Regensburger, G., Rosenkranz, M., Yoshida, H. (Eds.): 'Algebraic biology' (*LNCS*, **5147**, Springer, Heidelberg, 2008)
35 Horimoto, K., Nakatsui, M., Popov, N. (Eds.): 'Algebraic and numeric biology' (*LNCS*, Springer, Heidelberg), in press

7 Appendix 1: implementation of differential elimination and simplification in Maple 14

All procedures for deriving the DE constraints and for simplifying the resulting equations are described as the commands of the new Differential Algebra package in Maple 14 as given in Fig. 8.

8 Appendix 2: algorithms of numerical parts

Algorithms of the estimation of α value by using Pareto-optimal solutions (Algorithms 1-1 and -2 given in Figs. 9 and 10) and those of the real-coded GAs by the combination of UNDX with MGG (Algorithms 2-1, -2 and -3 given in Figs. 11–13) are described, respectively. Note that Algorithms 2-1, -2, and -3 are used in line 1 of Algorithm 1-2.

```
> 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
k21x2x1+ k21 x2ke -x1[t]x1 -x1[t]ke -k12x1 -k12x1ke -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
```

Fig. 8 *Commands of the new Differential Algebra package in Maple 14*

doi: 10.1049/iet-syb.2010.0051

```
> 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
```

Fig. 8 *(Continued)*

Algorithm 1-1

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

1: $P \leftarrow \phi$
2: $EV \leftarrow \phi$
3: $CV \leftarrow \phi$
4: n size of R
5: **for** $i = 0$ to n **do**
6: $EV \leftarrow EV$ union $E(R_i)$
7: $CV \leftarrow CV$ union $C(R_i)$
8: **end for**
9: **for** $i = 0$ to n **do**
10: Flag lp = true
11: **for** $j = 0$ to n **do**
12: **if** !($EV_i \leq EV_j$ and $CV_i \leq CV_j$) **then**
13: $lp \leftarrow$ false
14: **end if**
15: **end for**
16: **if** lp **then**
17: $P \leftarrow P$ union R_i
18: **end if**
19: **end for**
20: **return** P

Fig. 9 *Pareto-optimal solutions*

Algorithm 1-2

Function : estimate_alpha(δ, n, $\alpha = 1$, pop, gen, $trials$)
Input : error tolerance δ, number of trials n, population size of GA pop, maximum generation counts gen, trial number of GA $trials$, and tentative value of $\alpha = 1$
Return : estimated value of weighting factor α

1: $RES \leftarrow$ compute_parameter_set($\alpha = 1$, δ, pop, gen, $trials$)
2: $P \leftarrow$ select_pareto_optimal_solutions(RES)
3: $EV \leftarrow \phi$
4: $CV \leftarrow \phi$
5: n size of P
6: **for** $i = 0$ to n **do**
7: $EV \leftarrow EV$ union $E(P_i)$
8: $CV \leftarrow CV$ union $C(P_i)$
9: **end for**
10: fit $CV_i = -aEV_i + b$ from EV and CV by using least square method
11: **return** $a/(a + 1)$

Fig. 10 *Estimation of weighting factor α*

Algorithm 2-1

Function compute_next_generation(α, K)
Input : the weighting factor α, a parameter set K

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

Fig. 11 *Modification of the parameter set K by computing the next generation*

Algorithm 2-2

Function : compute_one_parameter_set(α, δ, *pop*, *gen*)
Input : the weighting factor α, the error tolerance δ for function F, the population size of GA *pop*, the maximum generation counts *gen*
Return : a set containing zero or one parameter set

1: create a set K containing *pop* random parameter sets
2: **for** $i = 1$ to *gen* **do**
3: compute_next_generation(α, K)
4: **if** an element k in K satisfies $E(k) \leq \delta$ **then**
5: **return** k
6: **end if**
7: **end for**
8: **return** ϕ

Fig. 12 *Optimisation process*

Algorithm 2-3

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

1: $RES \leftarrow \phi$
2: **for** $i = 1$ to *trials* **do**
3: $RES \leftarrow RES$ union compute_one_parameter_set(α, δ, *pop*, *gen*)
4: **end for**
5: **return** RES

Fig. 13 *Generation of a list of estimated parameter sets*

IET Syst. Biol., 2011, Vol. 5, Iss. 5, pp. 281–292
doi: 10.1049/iet-syb.2010.0051

Discovery of Chemical Compound Groups with Common Structures by a Network Analysis Approach (Affinity Prediction Method)

Shigeru Saito,[†,§] Takatsugu Hirokawa,[†] and Katsuhisa Horimoto*[,†,‡]

Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan, Chem & Bio Informatics Department, INFOCOM Corporation, Sumitomo Fudosan Harajuku Building, 2-34-17 Jingumae, Shibuya-ku, Tokyo, 150-0001, Japan, and and Institute of Systems Biology, Shanghai University, 99 Shangda Road, Shanghai 200444, China

Received July 9, 2010

We developed a method in which the relationship between chemical compounds, characterized by the secondary dimensional descriptors by a standard method, is first determined by network inference, and then the inferred network is divided into the compound groups by network clustering. We applied this method to 279 active inhibitors of factor Xa found by the first screening. A large network of 266 active compounds connected with 408 edges emerged and was divided into 10 clusters. Surprisingly, the chemical structures that were common within the clusters, but diverse between them, could be extracted. The activity differences between the clusters provide rational clues for the systematic synthesis of derivatives in the lead optimization process, instead of empirical and intuitive inspections. Thus, our method for automatically grouping the chemical compounds by a network approach is useful to improve the efficiency of the drug discovery process.

1. INTRODUCTION

Novel computational approaches and methodologies are increasing the efficiency of drug discovery, which involves numerous processes.[1] Indeed, various computational approaches in virtual screening are utilized to predict the activity of hypothetical compounds, based on the quantitative structure−activity relationship (QSAR).[2−5] In particular, the selection of compounds from a library or database of compounds is widely used to identify those that are likely to possess a given activity, when a single bioactive reference structure is available.[6−8] In this approach, fingerprint-based similarity searching is performed to identify the database molecules that are most similar to a user-defined reference structure.[9] Furthermore, the support vector machine is utilized to predict the activity of newly synthesized compounds with high accuracy.[10] The principal component analysis (PCA) also presents the relationship between the compounds, to allow a visual investigation of their activities in the principal component space. In particular, it generates a concept for the distribution of chemical compounds, named the chemical space, where different chemical compounds are reasonably distributed, depending on their corresponding origins.[11]

In spite of the popularity of computational approaches, empirical and intuitive approaches are still employed in drug discovery processes.[1] One reason for retaining the empirical and intuitive approaches is that after the first screening, the active compounds are usually compared in terms of the relationship between the chemical structure and its activity, before the next step of synthesizing the derivatives for selecting the ultimate lead. Unfortunately, this step partially depends on the empirical selection of the candidates for the chemical synthesis of the drug target, with reference to the chemical structures of the active and inactive compounds obtained by the first screening. Indeed, the structural information on the active compounds after the first screening is not fully utilized for selecting candidates of seeds for the derivative synthesis. Thus, the extraction of useful information about chemical structure and activity, in an automatic and visual manner, is desirable to systematically and efficiently synthesize derivatives for drug discovery.

We now propose an automatic method to visually group chemical compounds based on their structures, by using two types of network analysis methods. One is the network inference method. In the present study, we use the path consistency algorithm,[12] one of the graphical models from the family of probability models simplified by the conditional independences inherent in the graph,[13] which can visually infer the relationships between variables in a network form. Another is the network clustering method. This is a method to extract one property, named the "community structure", which indicates that the vertices in networks are often clustered into tightly knit groups, with a high density of within-group edges and a lower density of between-group edges.[14] This method is useful to automatically group the variables into some clusters from the connected network structure. Here we utilized the two network analysis methods to assess the relationship between chemical compounds. The utility of the present method is demonstrated by a set of chemical compounds after the first screening. The merits and pitfalls of the present method are also discussed, in terms of the previous computational methods.

* Corresponding author. E-mail: k.horimoto@aist.go.jp.. Telephone: +81 3 3599 8711.
† National Institute of Advanced Industrial Science and Technology (AIST).
§ INFOCOM Corporation.
‡ Shanghai University.

10.1021/ci100262s
Published on Web 12/09/2010

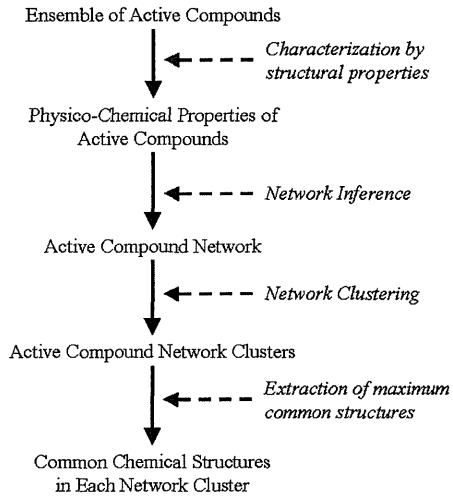


Figure 1. Workflow of the present method. The present method is schematically described in four steps.

2. MATERIALS AND METHODS

2.1. Overview of the Present Method. An overview of our method is schematically described in Figure 1. First, the chemical compounds selected by the first screening, in terms of drug activity, are characterized by their secondary structure properties, by a standard procedure. Second, the relationships between the compounds are investigated by a network inference method, the path consistency algorithm.[12] Third, the inferred network structures are divided into groups by a network clustering method, the Newman algorithm.[14] Fourth, the maximum common structures of the compounds are extracted in each cluster by a standard method. Thus, the characteristic features of the chemical structures hidden in the active chemical compounds are revealed visually and automatically by network analysis methods. The details of each step are described below.

2.2. Data Set. The data set contains a wide series of inhibitors of factor Xa extracted from the literature, all sharing a benzamidine moiety.[15] The considered data set contains 279 very active compounds (K_i lower than 10 nM) among a total of 435 chemical compounds, also including 156 low-activity compounds (K_i higher than 1 μM).

2.3. Descriptors. The calculated 2D descriptors were derived from the commercially available software, MOE, by Chemical Computing Group Inc. (http://www.chemcomp.com/). As a preprocessing step for the following analyses, the values of each descriptor were standardized by their averages and standard deviations. In this step, the number of descriptors was reduced, by leaving only the continuous values of the descriptors. Finally, 158 descriptors were used.

2.4. Network Inference by Path Consistency (PC) Algorithm with Modifications. The path consistency (PC) algorithm is a network inference method based on the graphical model.[12] The original PC algorithm is composed of two parts: the undirected graph inference by the partial correlation coefficient and the following directed graph generated by using the orientation rule. The present method partially exploits the first part of the PC algorithm, because the aim of the present application of the network inference method is to scrutinize the relationships between the chemical compounds, without the causality.

The algorithm for the first part is simple. The relationship between two variables is tested from the lower partial correlation coefficient to the higher one. For example, the relationship between the two variables is first tested by the zero-th partial correlation coefficient. If the null hypothesis is accepted, i.e., no association between the two variables, then no further test is performed for the higher order of the partial correlation coefficient. If it is rejected, then the relationship between the two variables is tested by the first partial correlation coefficient. In general, the $(m - 2)$-th order of the partial correlation coefficient is calculated between two variables, given $(m - 2)$ variables, i.e., $r_{ij,\mathrm{rest}}$, between X_i and X_j, given the 'rest' of the variables, $\{X_k\}$ for $k = 1, 2, ..., m$, and $k \neq i, j$, and after calculating the $(m - 2)$-th order of the partial correlation coefficient, the algorithm naturally stops. However, the algorithm does not usually request the $(m - 2)$-th order of the partial correlation coefficient for the natural stop. This is because no adjacent variables will be found after excluding the variables, even in the calculation of the lower order of the partial correlation coefficient. We provide the pseudocode of the algorithm in Figure 2.

In the sample data, the zero-th order (i.e., the condition where subset S is empty) of the partial correlation coefficient is calculated by Pearson's correlation coefficient, $r_{ij|S=\phi}$, expressed by

$$r_{ij|S=\varphi} = \frac{\mathrm{cov}(X_i, X_j)}{\sqrt{\mathrm{var}(X_i)\,\mathrm{var}(X_j)}}$$

where $\mathrm{cov}(X_i, X_j)$ and $\mathrm{var}(X_i)$ are the covariance between X_i and X_j and the variance of X_i. The higher order of the partial correlation coefficients, $r_{ij|S}$, expressed by

$$r_{ij|S} = \frac{-r^{ij}}{\sqrt{r^{ii} \cdot r^{jj}}}$$

where $ij|S$ means $S=\{1, 2, ..., p\}\backslash\{i,j\}$, and r^{ij} is the i-j element of the inverse correlation coefficient matrix.[13] Note that the dimensions of the correlation coefficient matrix are related to the orders of the partial correlation coefficients. The m-th order partial correlation coefficient is calculated from the $(m + 2)$ dimension of the correlation coefficient matrix. The partial correlation coefficient is statistically tested by using the Z-statistic.[16] First, z-transforms of the partial correlation coefficients are calculated, by the following equation:

$$z_{ij} = \frac{1}{2}\ln\left(\frac{1 + |r_{ij|S}|}{1 - |r_{ij|S}|}\right)$$

Then, the z-statistic is obtained from the following equation:

$$Z = \frac{z_{ij}}{\sqrt{1/n - 3 - p}}$$

where n is the number of samples and $p = |S|$ is the conditioning order of the partial correlation coefficient. The z-statistic follows the standard normal distribution, $N(0,1)$, and the significance probability can be set according to this distribution; i.e., we reject the null hypothesis H_0:$r_{ij|s} = 0$, if $Z > Z_{\alpha/2}$with significance level α. If H_0 is not rejected, then

Let $Adj(G,X_i) \setminus \{X_j\}$ be the set of nodes (variables) adjacent to X_i, except for X_j, in the undirected graph G.

Let p be the degree of conditioning.

1: $G \leftarrow$ complete undirected graph
2: $p = 0$
3: **repeat**
4: **for all** X_i such that $|Adj(G,X_i)| - 1 \geq p$ **do**
5: **for all** $X_j \in Adj(G,X_i)$ **do**
6: **for all** subset $S \subseteq Adj(G,X_i) \setminus \{X_j\}$ such that $|S| = p$ **do**
7: **if** $X_i \_\|\_ X_j \mid S$ **then**
8: delete edge between X_i and X_j in G
9: **end if**
10: **end for**
11: **end for**
12: **end for**
13: $p = p + 1$
14: **until** $|Adj(G,X_i)| - 1 \leq p$, $\forall X$
15: **return** G

Where "$X_i \_\|\_ X_j \mid S$" means X_i and X_j are conditionally independent on S; i.e., there is no edge between X_i and X_j.

Figure 2. Pseudocode of the modified path consistency algorithm. A pseudocode of the modified PC algorithm is described. In line seven, statistical hypothesis testing for the partial correlation between X_i and X_j conditioning on S is used to determine whether X_i and X_j are conditionally independent (for details, see text). If the partial correlation cannot be calculated, due to the multicollinearity, then we consider that X_i and X_j are always conditionally dependent on any other variables.

we consider $r_{ijls} = 0$, and we judge the i-th and j-th nodes as being conditionally independent of S.

The key point in the present network inference is the two modifications of the original PC algorithm, for application to the chemical compounds. The first modification is the correction of the algorithm in the calculation of the partial correlation coefficient. Since many compounds frequently show very similar descriptor values, the difficulty increases in the numerical calculation of the partial correlation coefficients, due to the multicolinearity between the variables. The original PC algorithm accidentally stops if only one partial correlation between a pair of variables violates the numerical calculation, against the high similarity of the descriptors. To avoid the accidental stops by the highly associated compound pairs, the original PC algorithm is modified as follows: If the calculation of any order of the partial correlation coefficient between the variables is violated, then the corresponding pair of variables is regarded as being dependent. The second modification is the correction of the output by the algorithm. The network inference outputs the edges with positive and negative correlations. The edge with a positive correlation in the network can be interpreted as a relationship with direct similarity between the properties of the chemical compound structures, while the edge with a negative correlation indicates a relationship with dissimilarity in a linear fashion. Thus, the edges with the positive correlation are adopted, and those with the negative correlation are excluded from the inferred network.

2.5. Grouping of Chemical Compounds by Network Clustering. In networks, the vertices are often clustered into tightly knit groups, with a high density of within-group edges and a lower density of between-group edges. This property is called a "community structure", and the computer algorithms for identifying the community structure are based on the iterative removal of edges with high "betweenness" scores, which identify such structures with some sensitivity. Here, we applied one of these algorithms to group the chemical compounds in the inferred network.[14]

This method is based on the modularity that is measured by a parameter, the Q-value. The Q-value is defined as follows:

$$Q = \sum (e_{ii} - a_i^2)$$

where e_{ii} means the fraction of edges in cluster i with respect to all edges in the network, and a_i means the fraction of the number of edges that end in cluster i. First, this method considers each node as a cluster. In each subsequent step, two clusters are combined to maximize the increment of the Q-value, ΔQ. ΔQ is calculated as follows:

$$\Delta Q = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j)$$

where e_{ij} means the half of the fraction of edges between clusters i and j with respect to all edges in the network. In addition to the above definition, e_{ij} is commutative, $e_{ij} = e_{ji}$, in the undirected graph. The complexity of the calculation is on the order $O(N)$, where N is the number of nodes in the network, and we combine two clusters at most $(N - 1)$ times; therefore, in sparse networks, the clustering is complete after $O(N^2)$ times.

2.6. Maximum Common Structures of Clusters. The maximum common structure within the constituent compounds belonging one cluster was obtained by using ChemAxon JKlustor libMCS.[17]

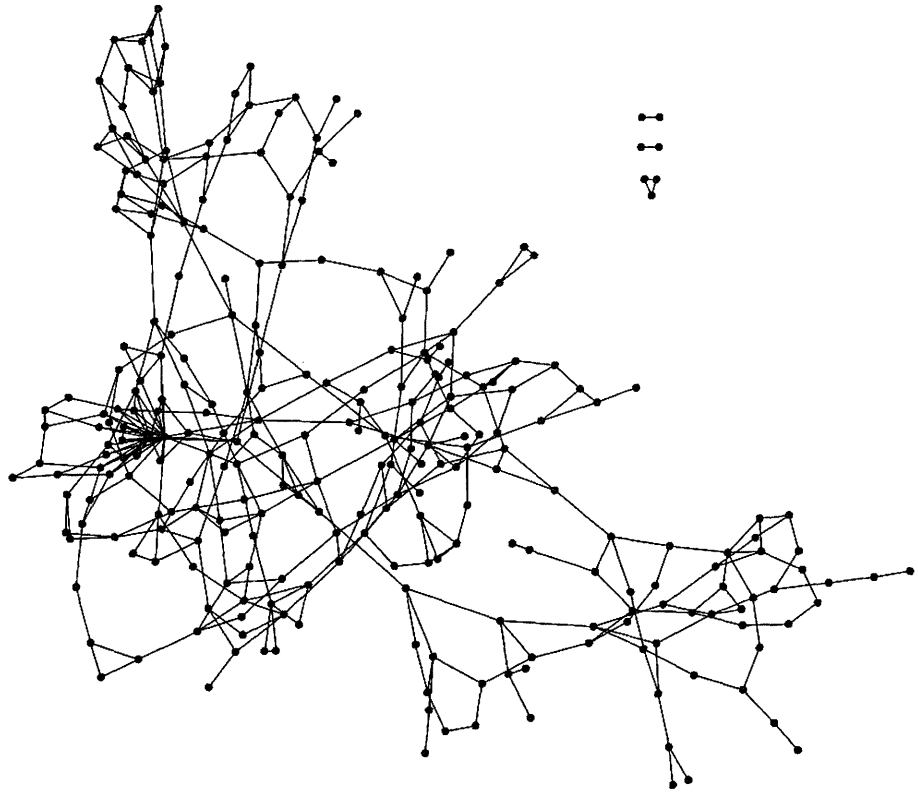


Figure 3. Chemical compound network inferred by path consistency algorithm. A large network of 266 compounds, inferred by the path consistency algorithm with 5% significance probability,[12] is described. The compounds and the established edges between compounds are denoted by open circles and straight lines, respectively.

3. RESULTS AND DISCUSSION

3.1. Chemical Compound Network. The relationships between the 279 active compounds were inferred by the PC algorithm. By the network inference, a large network containing 266 of the 279 active compounds emerged, as shown in Figure 3. Only seven compounds remained apart from the large network, and among them, five edges of the seven compounds were established. The emergence of a large network seems natural, because all of the compounds analyzed in this study share similar physicochemical properties, in terms of drug activity.

The large network contained 408 edges between compounds, and the average connectivity ([number of edges]/[$n(n-1)/2$], where n is the number of nodes) was about 0.0116. As shown in Figure 3, the inferred network was relatively sparse, in terms of edge connectivity. Although several hubs were observed in the network, it seems difficult to identify clear relationships between the compounds by visual inspection.

3.2. Chemical Compound Network Clusters. To scrutinize the compound relationships, we applied a network clustering method to rationally rearrange the connectivity in the inferred network of Figure 3. In Figure 4, 10 clusters naturally emerged from the entire connectivity in the inferred large network. Thus, the large, complicated network was transformed into distinctive clusters, with the number of compounds in each cluster ranging from 14 to 41. The emergence of the clusters indicates that some distinctive compound groups with similar structural properties exist in the network. The following step involves the investigation of the constituent compounds of each cluster that emerged by two network analyses, in terms of chemical structure and activity.

3.3. Common Structures of Chemical Compound Network Clusters. We surveyed the structural relationship between the constituent chemical compounds that belong to each cluster in the active network. Interestingly, the structures of the constituent compounds were common within each cluster, and they were diverse between the clusters.

The common structures of the member compounds in the clusters of the active network are shown in Figure 5A. It is readily apparent that common structures were found for all of the clusters, and high densities of the constituent compound structures were present in all of the clusters. Indeed, on average, ca. 63.7% of the compounds shared common structures: the highest and lowest share rates were 100.0% in cluster 6 and 35.5% in cluster 3. In addition, the average density of heavy atoms over all constituent compounds in each cluster was high: 9 of the 10 clusters showed more than 50% of the average density, and the exceptional cases were found in cluster 8. Furthermore, the common structures of each cluster were distinctive between them, as seen in Figure 5A. To estimate the differences between the common structures, the Tanimoto coefficients were calculated between them, as follows:

$$T_{ij} = \frac{\sum_k (X_{ik}X_{jk})}{\sum_k X_{ik}^2 + \sum_k X_{jk}^2 - \sum_k (X_{ik}X_{jk})}$$

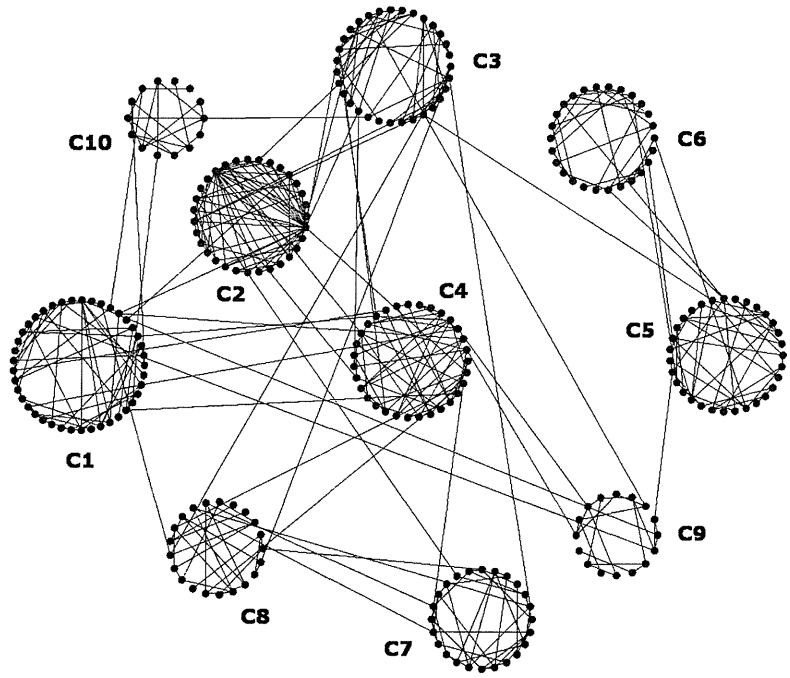


Figure 4. Network clusters of a large network of active chemical compounds. The network clusters estimated for the large network of active compounds in Figure 3 are depicted. Ten clusters emerged, and they are numbered in the order of the numbers of constituent compounds within the clusters.

As shown in Table 1, the Tanimoto coefficients for all pairs of common structures were much less than 0.85, a value that is generally considered to reflect similarity to each other. All of the coefficients were less than 0.4, except for only 0.688 between the common structures of clusters 5 and 6.

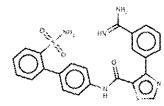
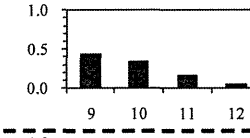
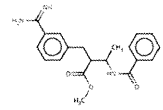
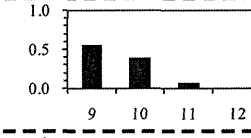
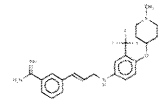
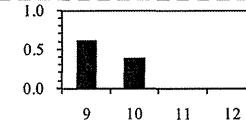
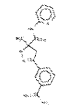
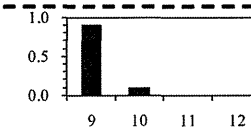
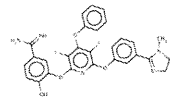
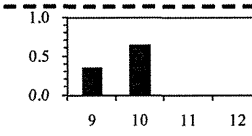
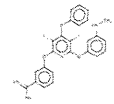
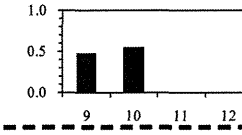
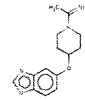
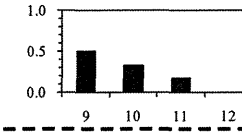
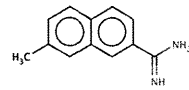
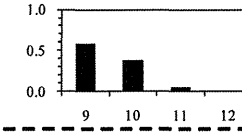
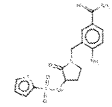
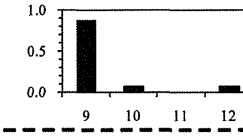
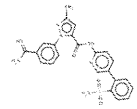
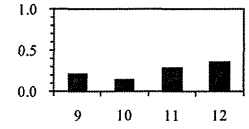
The structures common within clusters and diverse between clusters were further investigated in terms of the activity distribution, expressed by the $-\log(IC_{50})$ histogram of the constituent compounds. For reference, the histogram of all compounds was also drawn in Figure 5B, and in the histogram, the compounds with a $-\log(IC_{50})$ value of less than 9 (10 nM), which is generally regarded as the lead compound, were frequently included (29.3% of compounds). Subsequently, the compounds with IC_{50} values less than 10 nM were frequently observed in the histograms of each cluster in Figure 5A. Interestingly, some exceptions were also observed. A statistical difference between the total IC_{50} distribution in Figure 5B and the distributions in Figure 5A was found in several clusters. In the distributions of clusters 5 and 10, the frequency of observing an IC_{50} value than 10nM was relatively high, in comparison with the total distribution. This indicates that the common structures in clusters 5 and 10 may show a robust IC_{50} for any chemical modification. Thus, the common structure may be a candidate for lead optimization. In contrast, the frequencies of IC_{50} values less than 10 nM in clusters 4 and 9 were much lower than that in the total IC_{50} distribution. This indicates the possibility that many compounds with an IC_{50} activity of less than 10 nM can be synthesized from the common structures of the two clusters. Thus, the correspondence between the common structures and the IC_{50} distributions of each cluster provides some clues for the synthesis of new compounds in the lead optimization process.

3.4. Related Methods. For comparison with the performance of the present method, the PCA was performed for the same data. Figure 6 shows the projection of the cluster members of the active network in Figure 4 into the principal component space. As easily seen in the figure, the cluster members with each common structure are scattered in the space. Indeed, the constituent compounds in each cluster were projected into some duplicated spaces, while the compounds of clusters 5 and 6 were relatively separated from the other clusters in the projected space. As indicated in the preceding subsection, the Tanimoto coefficient between the common structures of clusters 5 and 6 was exceptionally large, and this similarity reflects the common configuration of the constituent compounds in the two clusters in the principal component space. In contrast, the Tanimoto coefficients between the common structures of the other clusters were small, and therefore the compounds were not clearly discriminated in the space. Thus, the PCA may be a low-resolution method to clearly detect the groups with common chemical structures in the data, followed by the first screening.

The fingerprint approach is well-known as another method to detect common structures in an ensemble of compounds.[9] Actually, we used this approach to identify the common structure of the constituent compounds in the respective clusters. As a trial, we applied the fingerprint approach to all of the compounds but were unable to find the common structure (data not shown). We expected this failure from the fact that the common structures of each cluster show much less similarity, as depicted in Figure 5A. In contrast to the PCA, therefore, the fingerprint method may be a high-resolution technique to detect the distinctive groups with common chemical structures.

Note that there are two reasons why we use the Newman method, instead of the standard hierarchical clustering by using the partial correlation matrix as a distance measure. One reason is that the edges in the inferred network are established by considering the higher order of correlation between multiple variables, instead of the distance between

A

Cluster No.	MCS	No. of Compounds	Density	Distribution
1.		41(18).	0.969.	
2.		31(23).	0.767	
3		31(11).	0.880	
4**		31(17).	0.648	
5*		31(18).	0.953	
6.		26(26).	0.739	
7.		24(21).	0.542	
8		21(16).	0.376	
9*		16(8).	0.809	
10**		14(8).	0.967	

B

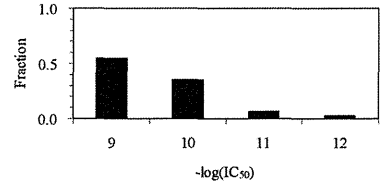


Figure 5. Maximum common structures of the active compound network clusters, together with IC_{50} histograms of constituent compounds. (A) The numbers of clusters in the first column are those described in Figure 4. The common structures of the 10 clusters in the second column were extracted by using ChemAxon JKlustor libMCS.[17] The total number of constituent compounds in each cluster is denoted in the third column, and the number of compounds sharing the corresponding common structures is also denoted in parentheses. In the fourth column, the average densities of heavy atoms in the common structures over the structures of all compounds are denoted. In the fifth column, the histograms of the IC_{50} values of the constituent compounds are depicted: the vertical and horizontal axes are the frequency of the compounds and the $-\log(IC_{50})$ values, respectively. In addition, the differences between each histogram of IC_{50} values for the respective clusters and that for the total active compounds (B) were tested by Fisher's exact test. The significance of the differences between the histograms is indicated at the cluster number in the first column: 5%, '**'; and 10%, '*'.

Table 1. Tanimoto Coefficients between Maximum Common Structures in Respective Active Compound Clusters

cluster no.	1	2	3	4	5	6	7	8	9	10
1	–									
2	0.357	–								
3	0.243	0.254	–							
4	0.261	0.304	0.244	–						
5	0.179	0.197	0.199	0.202	–					
6	0.160	0.166	0.226	0.232	0.686	–				
7	0.187	0.193	0.192	0.123	0.132	0.124	–			
8	0.137	0.224	0.176	0.152	0.126	0.150	0.073	–		
9	0.254	0.271	0.234	0.229	0.213	0.198	0.151	0.157	–	
10	0.213	0.165	0.228	0.274	0.222	0.246	0.142	0.095	0.246	–

pairs of variables in the clustering, The clustering technique in the present study is therefore suitable for keeping the inferred relationships between variables. The other reason is that the Newman method can automatically determine the number of clusters in terms of the network structure. In contrast, the number of clusters is determined by setting a threshold, as in hierarchical clustering, or the cluster number is done before the clustering, as in a self-organization map (SOM).

In summary, the PCA provides a coarse-grinning relationship between compounds from the macroscopic resources, and the fingerprint approach provides a fine relationship between limited ensembles of compounds. With these situations in mind, our procedure provides a medium relationship between compounds, to enrich the selection of molecules with a desired activity. Thus, it bridges the gap between the two methods, by finding the groups of common structures in the step after the first screening, during the process of the lead optimization.

3.5. Merits and Pitfalls of the Present Method. One of the merits of the present method is that it simply detects the structural similarity relationships between active compounds. Indeed, only one parameter, the significance probability in the path consistency algorithm, is set in the network analyses. Thus, the present method is highly automatic and visual, to help reveal a rational synthesis route of chemical compounds for new drug discovery.

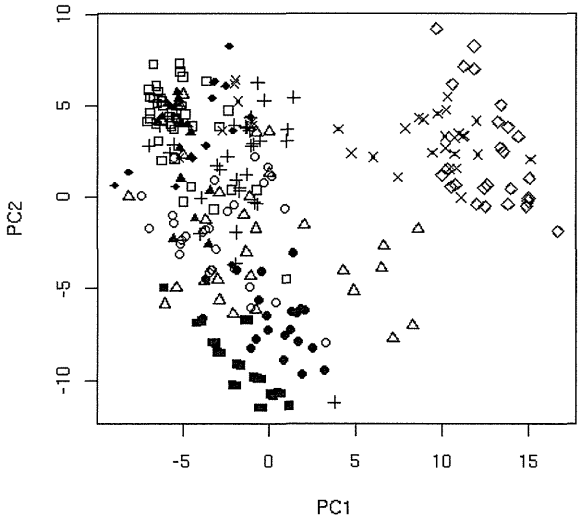


Figure 6. Distribution of members of network clusters in principle component space. The 266 active compounds in the network of Figure 3, which were characterized by the same number of descriptors (158 descriptors) as in the present analysis, were subjected to the PCA. The inertias of the first and second principal components (PC1 and PC2 in the figure) were 0.309 and 0.198, respectively. The constituent compounds of the 10 clusters in Figure 4 are indicated by the following symbols: cluster 1, □; 2, ○; 3, Δ; 4, +; 5, ×; 6, ◇; 7, ■; 8, ●; 9, ▲; and 10, ◆.

One of the key points of our method is the application of network inference, based on the graphical model, to the chemical compounds. Among the similar chemical structures, the present network inference detects the 'well-balanced' similarity, by using the partial correlation coefficient. In general, the graphical model distinguishes between real correlation and pseudocorrelation, based on the calculation of a partial correlation coefficient that realizes the concept of conditional independence.[13] The merit of this graphical model is that it only establishes the connection between the compounds with common structures and not between those lacking common structures. This discriminative ability is useful for classifying a large number of active compounds into various groups with different common structures in a rational manner.

In the present analysis, one large network was inferred, and 10 clusters emerged. The numbers of networks and clusters naturally depend on the user-defined descriptors and one parameter in the network inference. In the present analysis, the chemical compounds were characterized by as many secondary structure descriptors as possible. In general, the kinds of descriptors in the analysis may be changed, according to the analyzed data and the analysis aim. Fortunately, the quantification of chemical compounds by descriptors can be easily and quickly performed, due to recent advances in high-performance computing. Although the heuristic choice of descriptors is important to characterize the compound set, the descriptor optimization responsible for the compound set can be included as a preprocessing step in the present work. Furthermore, the size of the network and the following cluster numbers can be controlled by the user-defined significance probability in the network inference. For example, if one chooses a more significant probability than that of the present study, then a smaller network and fewer clusters will be obtained, in which more similar common structures will be found. In addition, the computational time for the present data in the two network analyses was about 5 s, using a personal computer (one CPU with a 2.4 GHz Pentium IV processor and 1GB of memory, under the Linux system). At any rate, the easy manipulation of the data, using only one user-defined parameter, may promote the use of the present method in applications to discriminate between various active compounds in drug discovery.

4. CONCLUSIONS

We have proposed a novel method to group active chemical compounds, by first screening with a combination of two network analysis methods. The scrutinization of active inhibitors of factor Xa by our method revealed reasonable grouping in terms of chemical structure and significant differences between each group in terms of activity. The present results illustrate the possibility that our method will bridge the gap between the compound activity test by the first screening and the following synthesis of lead derivatives.

ACKNOWLEDGMENT

K.H. was partly supported by a Grant-in-Aid for Scientific Research on Priority Areas "ystems Genomics" (grant

20016028) and by a Grant-in-Aid for Scientific Research (A) (grant 19201039), from the Ministry of Education, Culture, Sports, Science, and Technology of Japan.

REFERENCES AND NOTES

(1) Lipinski, C.; Hopkins, A. Navigating chemical space for biology and medicine. *Nature* **2004**, *432*, 855–861.
(2) Todeschini, R.; Consonni, V. The Handbook of Molecular Descriptors, in the Series of Methods and Principles in Medicinal Chemistry; Mannhold, R., Kubinyi, H., Timmerman, H., Eds.; Wiley-VCH: New York, 2000; Vol. 11.
(3) Katritzky, A. R.; Gordeeva, E. V. Traditional topological indexes vs electronic, geometrical, and combined molecular descriptors in QSAR/QSPR research. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 835–857.
(4) Karelson, M.; Lobanov, V. S.; Katritzky, A. R. Quantum-chemical descriptors in QSAR/QSPR studies. *Chem. Rev.* **1996**, *96*, 1027–1043.
(5) Devillers, J.; Balaban, A. T. *Topological Indices and Related Descriptors in QSAR and QSPR*; Gordon and Breach: Amsterdam, The Netherlands, 1999.
(6) Bajorath, J. Integration of virtual and high throughput screening. *Nat. Rev. Drug Discovery* **2002**, *1*, 882–894.
(7) Bajorath, J. Virtual screening: methods, expectations, and reality. *Curr. Drug Discovery* **2002**, *2*, 24–28.
(8) Sheridan, R. P.; Kearsley, S. K. Why do we need so many chemical similarity search methods. *Drug Discovery Today* **2002**, *7*, 903–911.
(9) Hert, J.; Willett, P.; Wilton, D. J. Comparison of Fingerprint-Based Methods for Virtual Screening Using Multiple Bioactive Reference Structures. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1177–1185.
(10) Jorissen, R. N.; Gilson, M. K. Virtual Screening of Molecular Databases Using a Support Vector Machine. *J. Chem. Inf. Model.* **2005**, *45*, 549–561.
(11) Dobson, C. M. Chemical space and biology. *Nature* **2004**, *432*, 824–828.
(12) Spirtes, P.; Glymour, C.; Scheines, R. *Causation, Prediction, and Search (Springer Lecture Notes in Statistics)*, 2nd ed., revised; MIT Press, Cambridge, MA, 2001.
(13) Whittaker, J. *Graphical Models in Applied Multivariate Statistics*; Wiley: New York, 1990.
(14) Newman, M. E. J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E.* **2004**, *69*, 066133.
(15) Fontaine, F.; Pastor, M.; Zamora, I.; Sanz, F. Anchor-GRIND: Filling the Gap between Standard 3D QSAR and the GRid-INdependent Descriptors. *J. Med. Chem.* **2005**, *48*, 2687–2694.
(16) Sokal, R. R.; Rohlf, F. J. *Biometry: The Principles and Practices of Statistics in Biological Research*, 3rd ed.; W. H. Freeman: 1994.
(17) *JChem JKlustor LibMCS*, version 5.3.8; ChemAxon: Budapest, Hungary, 2010.

CI100262S