生化学反応系の数理モデル
-パラメータフィッティング-

藤井 雅史
東京大学 黒田研

お知らせ

今日使うファイル類は
http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3/
に置いてあります。（テキストエンコーディングはSJIS）

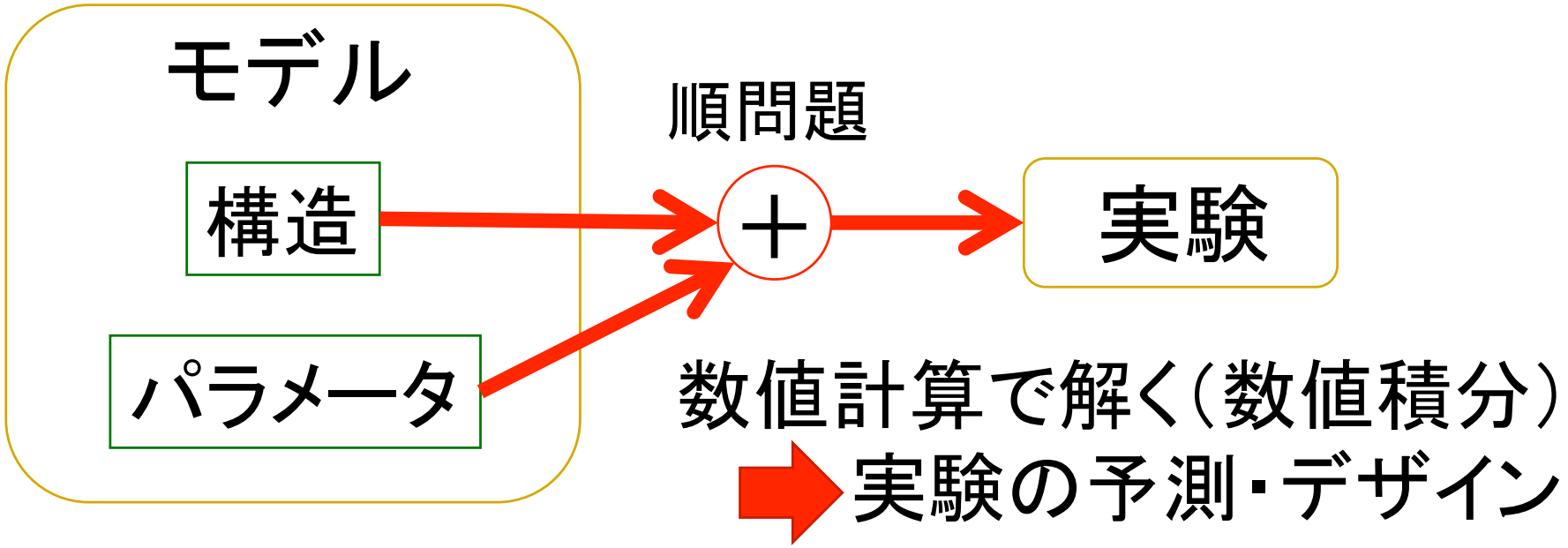
慣れてきたら自力で全部書く、あるいは、
これまで作ったプログラムを応用して作るようにして下さい。

課題が終わった人は、
積極的に発展課題に取り組んで下さい。

前回の復習

前提：どんな反応があるか（モデルの構造）は既知

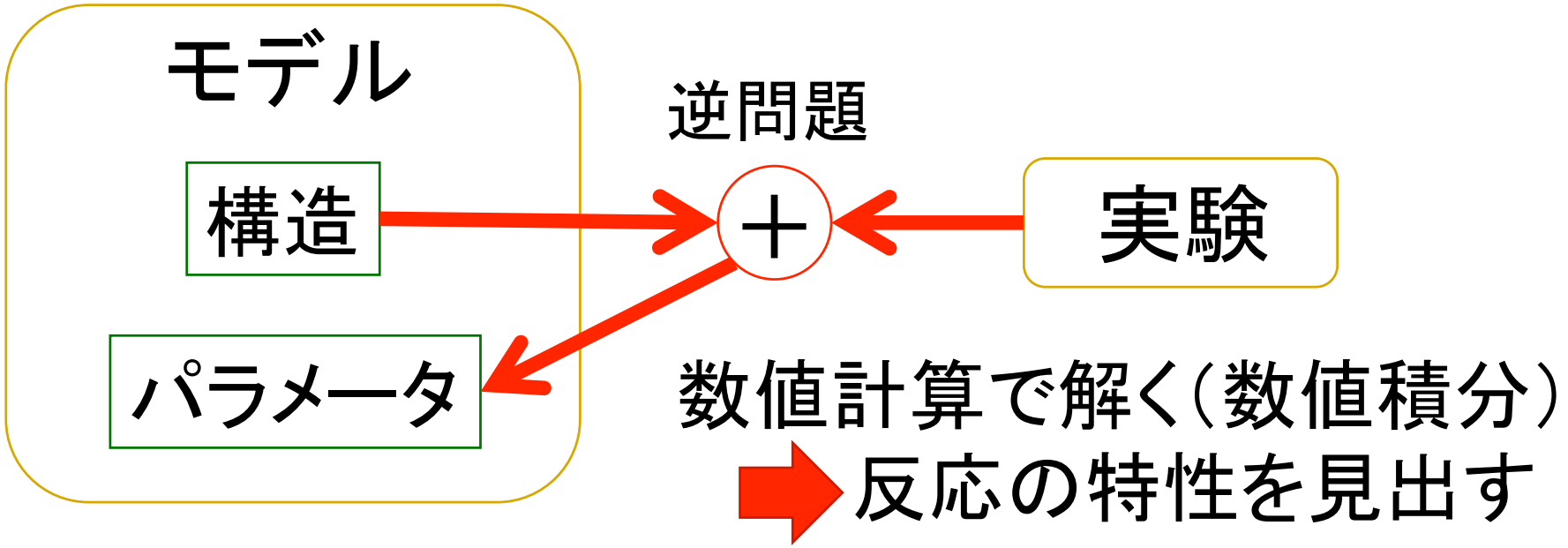
➡ 常微分方程式の形で記述



今日やること

前提：どんな反応があるか（モデルの構造）は既知

➡ 常微分方程式の形で記述



順問題（前回まで）

反応式

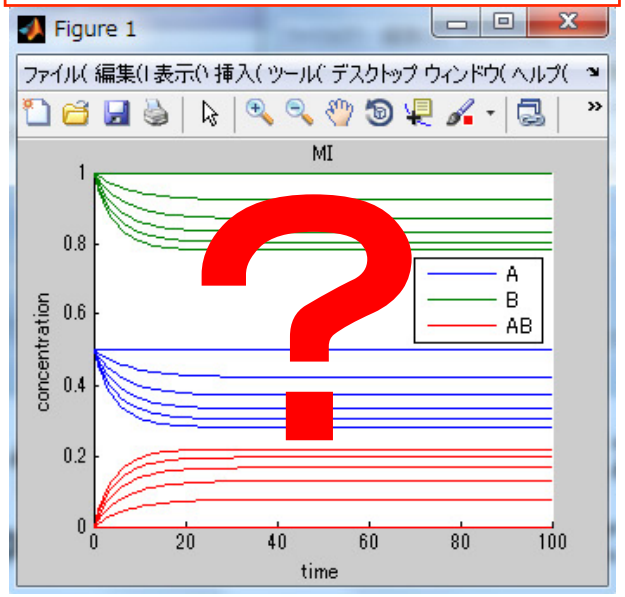
$$[A] + [B] \underset{k_b}{\overset{k_f}{\rightleftarrows}} [AB]$$

モデル

$$\frac{d[AB]}{dt} = k_f[A][B] - k_b[AB]$$

モデル＝反応式（構造）＋パラメータ
（定性的）（定量的）

数値シミュレーション



モデルさえ決まってしまえば
様々な条件下での振る舞いが
シミュレーション可能！

⇒ 解析、実験デザインに有用

逆問題（今回やること）

モデルの構造 ＋ 実験データ（時間波形）

↓

実験データに合うようなパラメータを探す

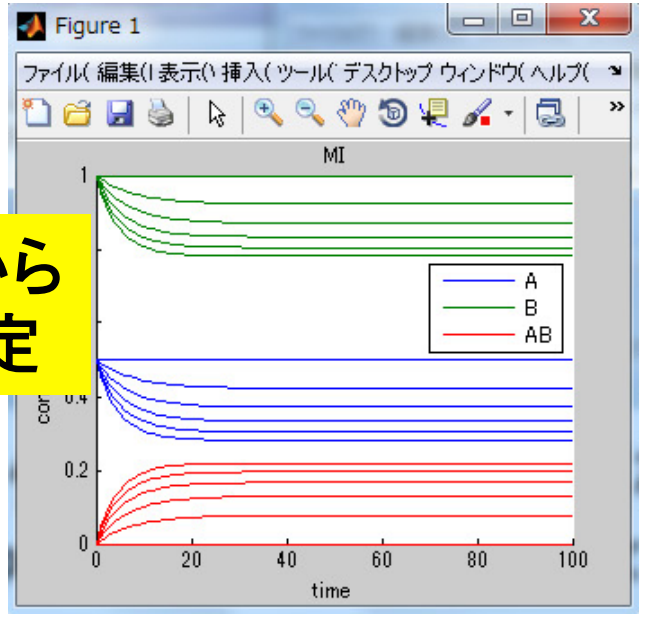
モデル

$$[A] + [B] \underset{k_b}{\overset{k_f}{\rightleftarrows}} [AB]$$

↓

$$\frac{d[AB]}{dt} = k_f[A][B] - k_b[AB]$$

時間波形から k_f、k_b を推定



逆問題（今回やること）

パラメータを探す…？

〇文献を参照　⇒　実験条件によってまちまち

〇反応の速度定数を実験的に測る！

⇒　1）パラメータが多いとしんどい…

2）誤差を含む（正確にわかるわけでない）

つまり大変！

ちょっと脱線して… モデルの簡略化

アンケート:「都道府県の大まかな位置を知るモデル」としては、
どちらがわかりやすいですか？

複雑モデル

簡単モデル

ちょっと脱線して… モデルの簡略化

アンケート:「都道府県の大まかな位置を知るモデル」としては、
どちらがわかりやすいですか？

複雑モデル

簡単モデル

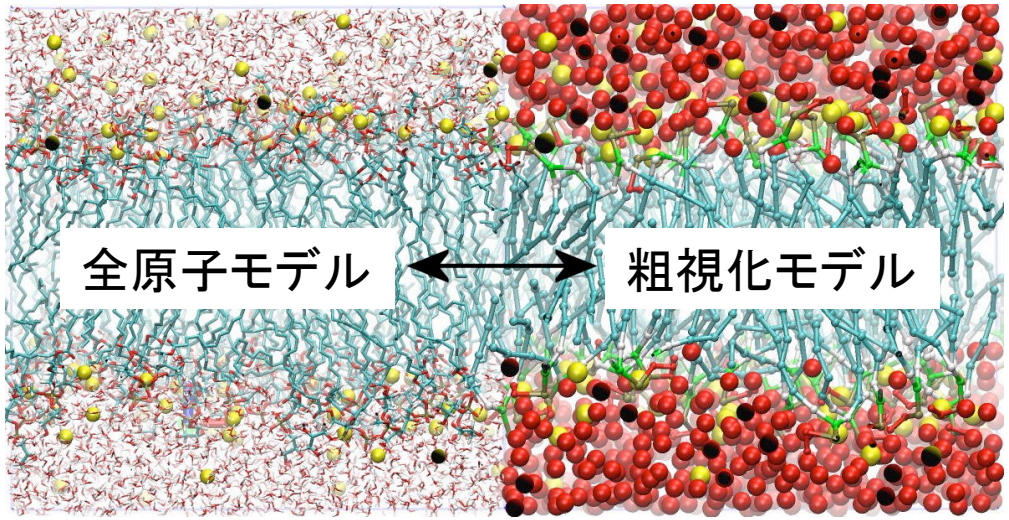
無駄な情報が無くて
わかりやすくて
いいじゃん！

＝要求を満足

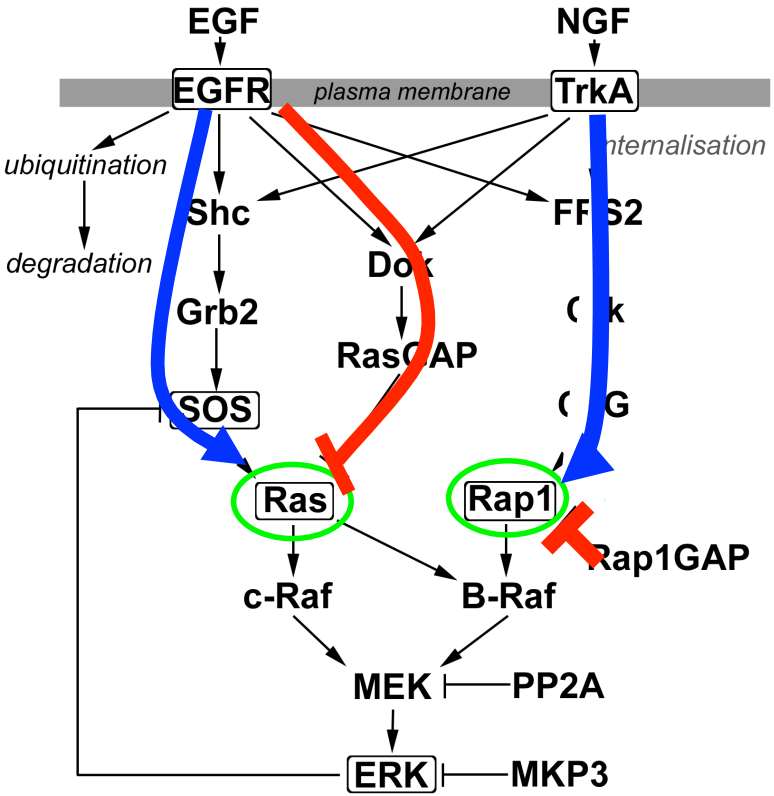
ちょっと脱線して… 生物系とモデルの簡略化

生物系とモデルの簡略化は親和性が高い

分子模型



反応ネットワーク



厳密だけど複雑なモデル
近似だけど簡単なモデル
どちらを選ぶか目的依存だが

粗視化・簡略化をすると本質を抜き出しやすい

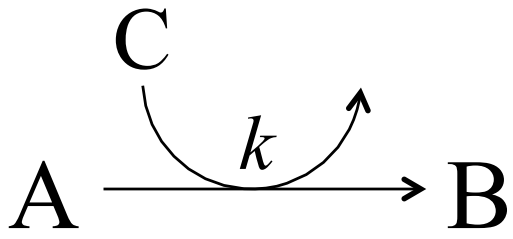
（詳細なモデルは計算時間がかかるという問題点も）

モデルの構造とパラメータ

例：酵素反応

定数と見せかけて
実はさらに詳細な反応が…
結合部位xが結合し
結合部位yが結合し
(以下略)

実際の反応　$A + C \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} AC \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} BC \xrightarrow{k_3} B + C$

簡略化　 

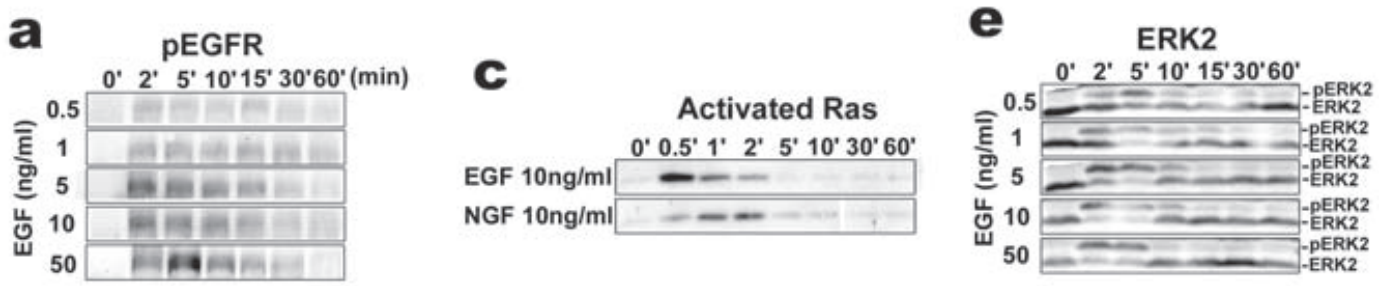
システム全体の振る舞いが知りたいとき
知りたい速度定数は？（$k_1, k_{-1}, k_2, k_{-2}, k_3, k_{-3}$）？$k$？

モデルの構築は目的依存（何を見たいのか）
➡ パラメータはモデルに依存して決まる！

モデルの決め方

例：細胞を刺激することで測定された
各分子の時間波形の実験データを再現するモデルが欲しい！

実験データ：ウエスタンブロッティング（WB）やイメージングなどにより取得



必要な情報：対象分子の（相対的な）濃度の時間波形…
不要な情報：詳細な反応過程、空間情報…

⇨ 反応過程を簡略化したモデルの構造を決定！

⇨ モデル・実験データに適したパラメータが欲しい！

…どうやって？

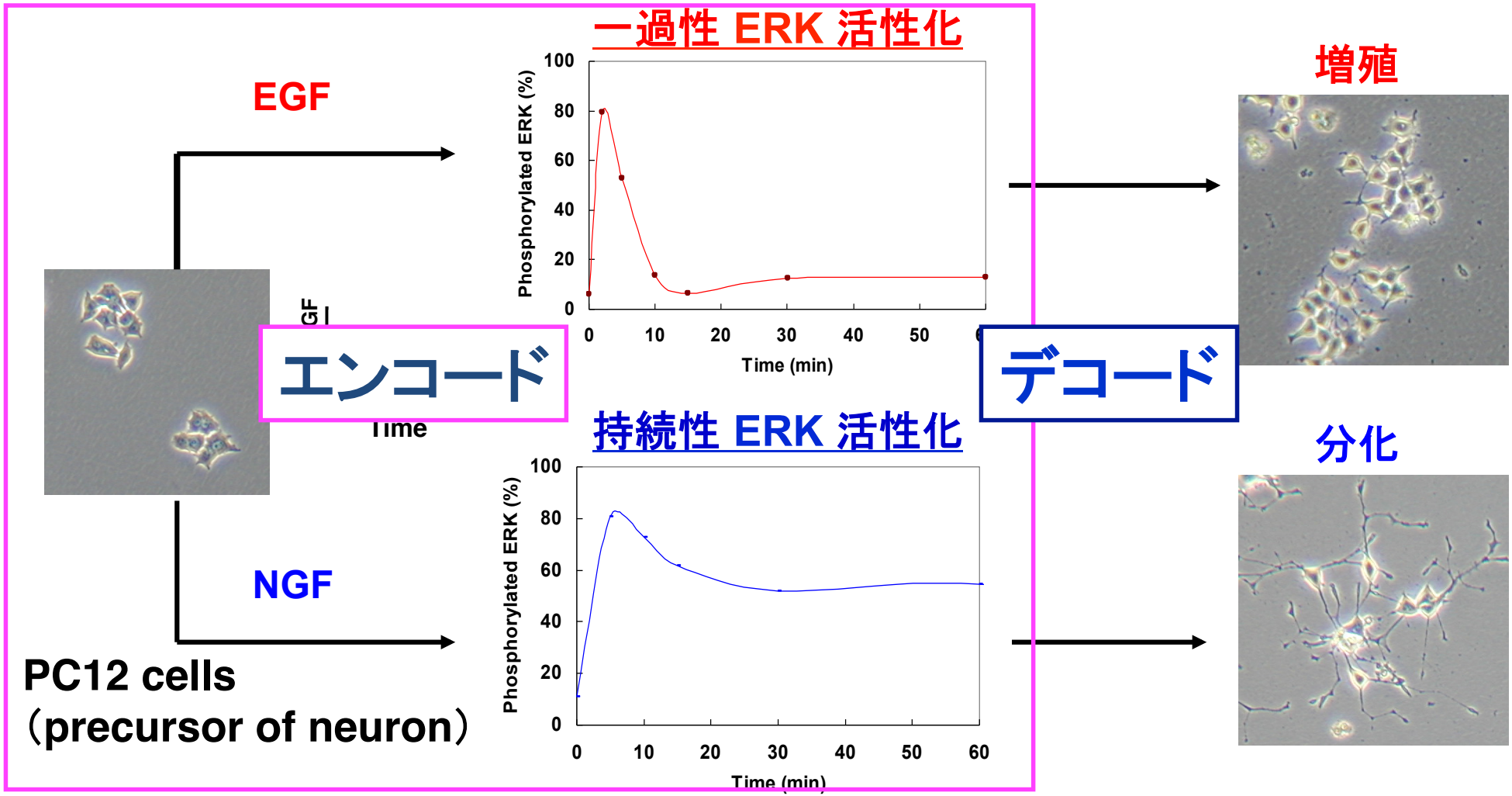
今日の流れ

1. Ras, Rap1のシンプルODEモデルを作成
2. パラメータ推定と最小2乗法について
3. ODEモデルの１つの未知パラメータを推定
4. ODEモデルの４つの未知パラメータを推定

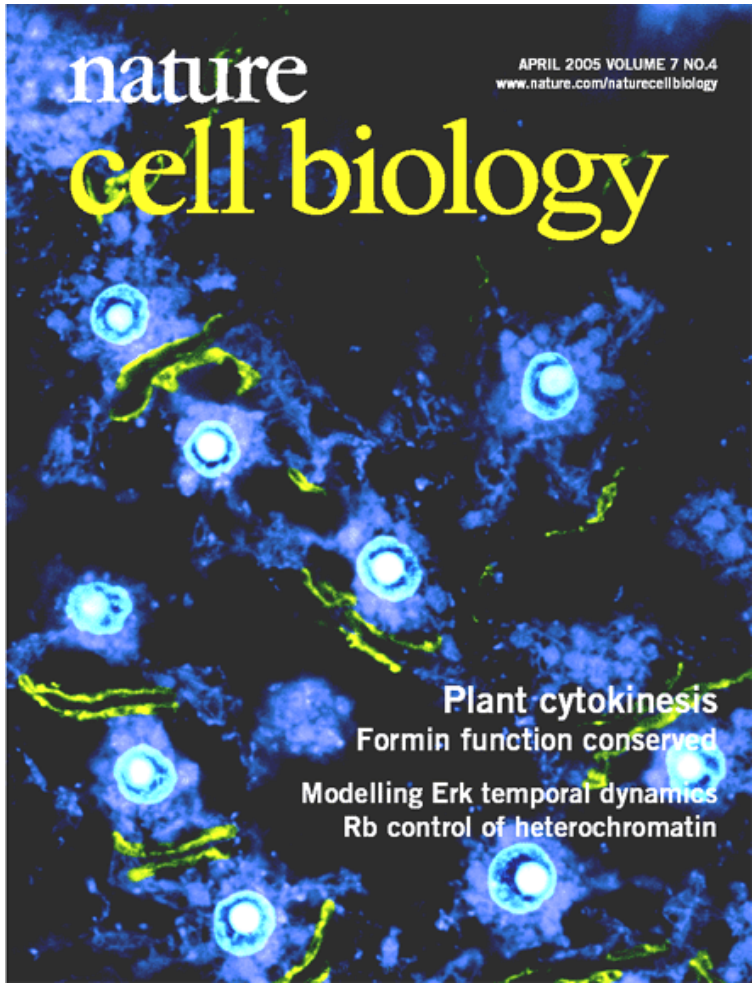
細胞運命決定機構：増殖と分化のスイッチ

現象を分子に帰着せさることができない！

ERKの時間波形が異なるだけで異なる現象を制御



細胞運命決定機構：増殖と分化のスイッチ
現象を分子に帰着せさることができない！



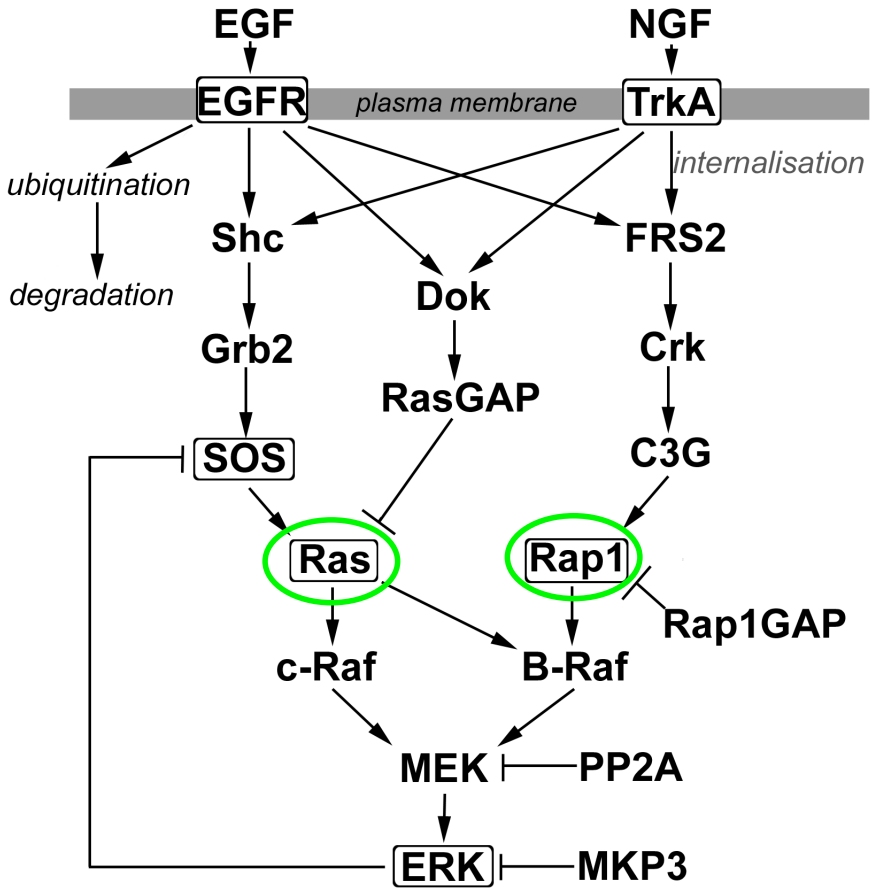
不活性化のメカニズムが異なるだけで
一過性応答：速さを捉える！
持続性応答：強さを捉える！

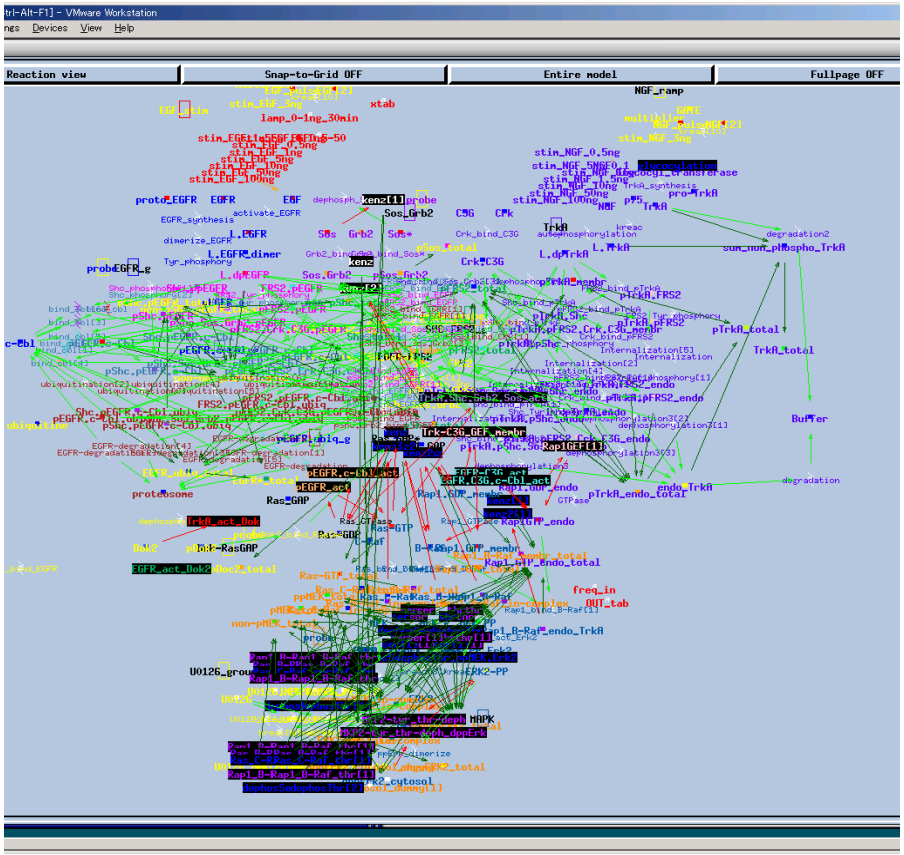
簡単な前向き制御
Feedforward systems
Activation and inhibition

Sasagawa, S. et al, Nat. Cell Biol. 2005, 7 (4), 365-373
www.kurodalab.org

Ras経路とRap1経路の生化学反応モデル

Erk経路のブロック線図

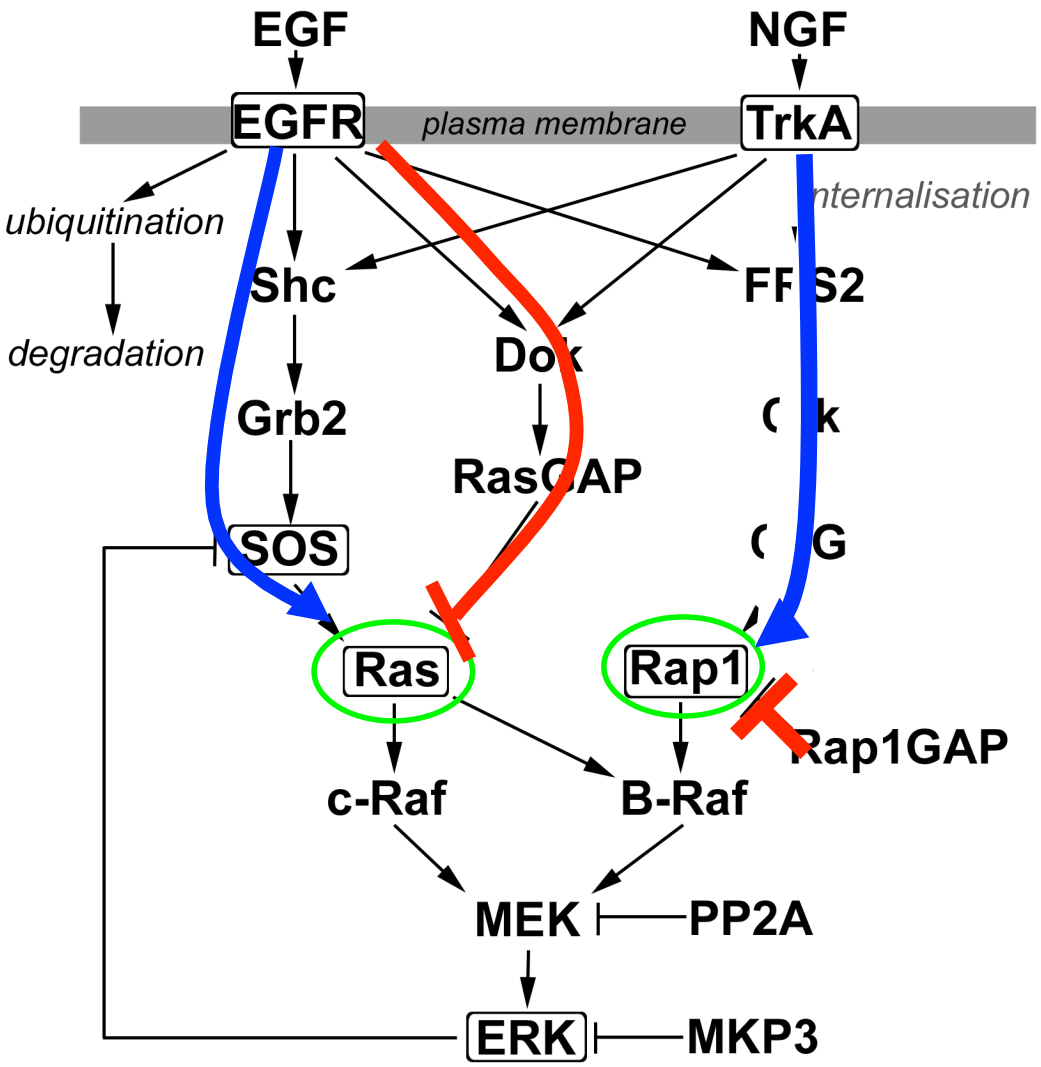




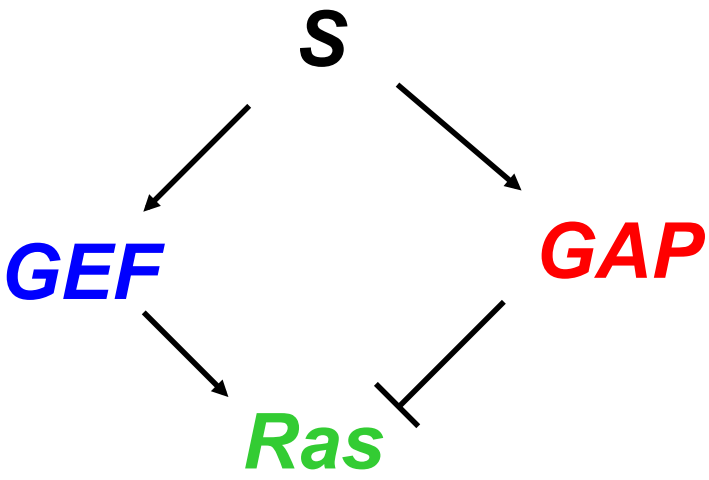
変数・パラメータが多すぎる！

ERK経路モデルの簡単化

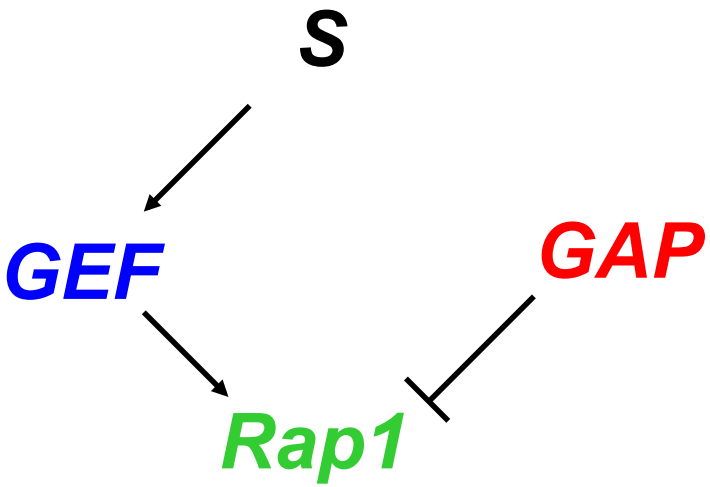
Erk経路のブロック線図



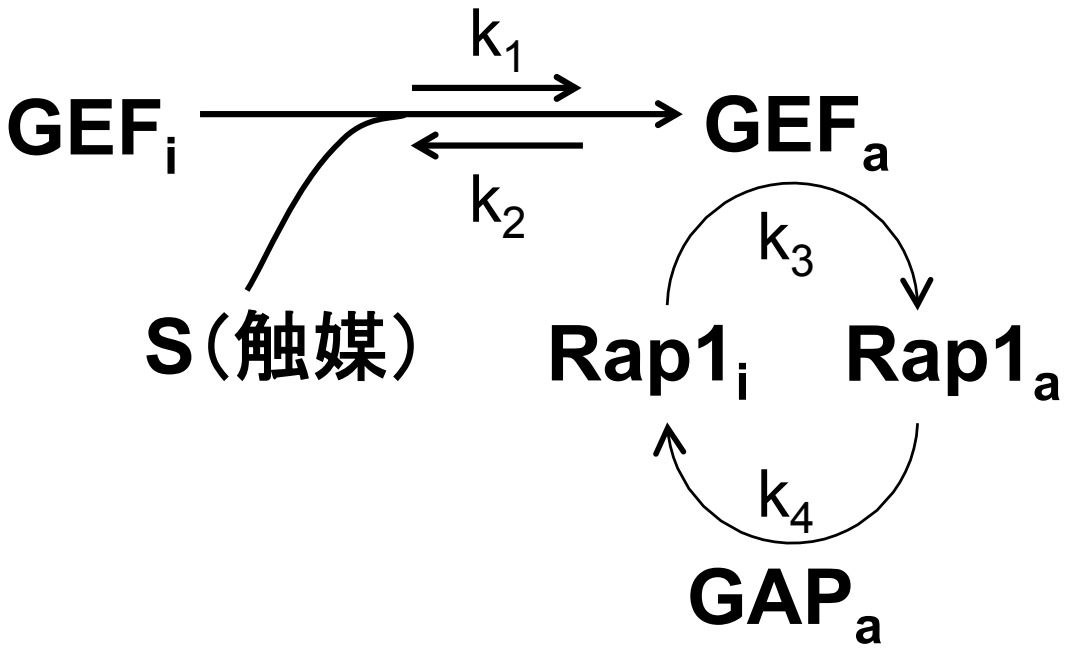
シンプルRasモデル



シンプルRap1モデル



シンプルRap1モデル



変数（分子）	初期濃度
S	1（定数）
GEF_a	0
GAP_a	0.005（定数）
$Rap1_a$	0

パラメータ	値
k_1	0.5
k_2	0.5
k_3	0.2
k_4	10

不活性型と活性型の総和は保存

$$GEF_i + GEF_a = 1$$
$$GAP_i + GAP_a = 1$$
$$Rap_i + Rap_a = 1$$

＊ひな形をダウンロードして，Rap1_model.mとして保存

http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3

“Rap1_model.m”をダウンロードして空欄を埋める

その1/2

```
function Rap1_model()
    % Rap1モデル
    time = 0:100;
    param = [                ]; % [k1, k2, k3, k4]の順に
    y0 = [                   ]; % [S, GEFa, GAPa, Rap1]の順に

    [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, param), time, y0);

    figure;
    plot(t_sim,timeCourse_sim(:,4));
    xlabel('time');
    ylabel('concentration');
    title('Rap1 model');

end
```

“Rap1_model.m”をダウンロードして空欄を埋める

その2/2

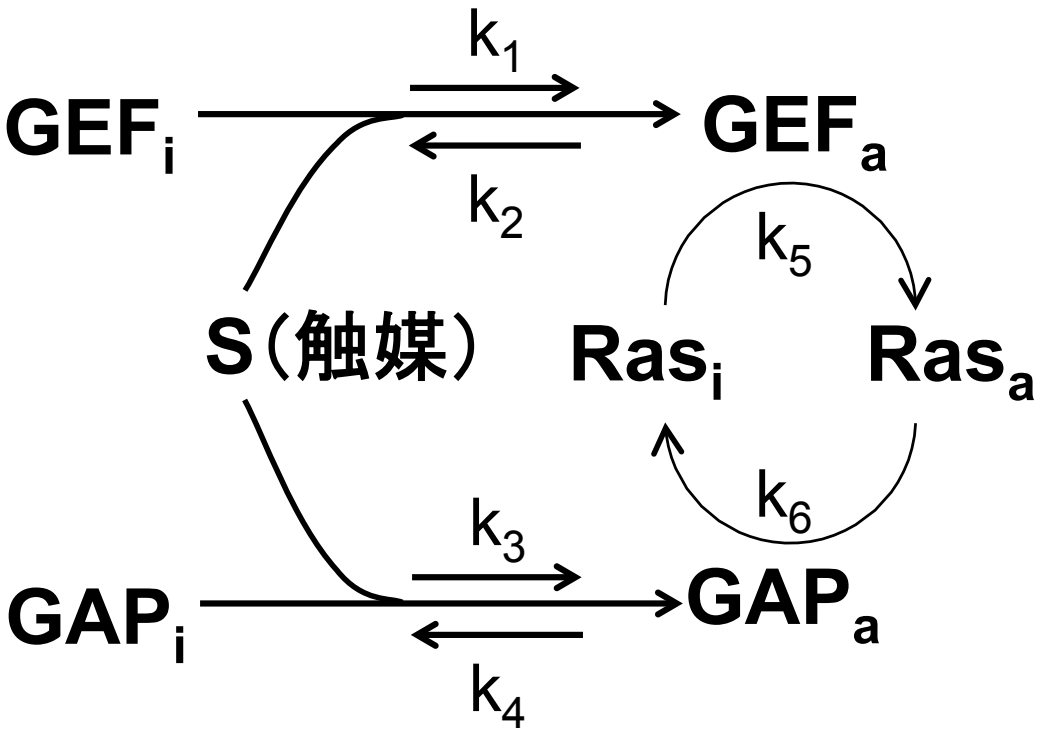
```
function dydt = ODE(t, y, param)
    S = y(1);
    GEFa = y(2); % GEFi = 1 - GEFa
    GAPa = y(3); % GAPi = 1 - GAPa
    Rap1a = y(4); % Rap1i = 1 - Rap1a

    k1 = param(1);
    k2 = param(2);
    k3 = param(3);
    k4 = param(4);

    dydt(1, :) = [                              ]; % S
    dydt(2, :) = [                              ]; % GEFa
    dydt(3, :) = [                              ]; % GAPa
    dydt(4, :) = [                              ]; % Rap1a
end
```

“Rap1_model.m”をダウンロードして空欄を埋める

解答例

```
function Rap1_model()
    % Rap1モデル
    time = 0:100;
    param = [0.5, 0.5, 0.2, 10.]; % [k1, k2, k3, k4]の順に
    y0 = [1, 0, 0.005, 0]; % [S, GEFa, GAPa, Rap1]の順に
    （略）
end
function dydt = ODE(t, y, param)
    （略）
    dydt(1, :) = 0; % S
    dydt(2, :) = k1 * S * (1 - GEFa) - k2 * GEFa; % GEFa
    dydt(3, :) = 0; % GAPa
    dydt(4, :) = k3 * GEFa * (1 - Rap1a) - k4 * Rap1a * GAPa; % Rap1a
end
```

シンプルRasモデル



変数（分子）	初期濃度
S	1（定数）
GEF_a	0
GAP_a	0
Ras_a	0

パラメータ	値
k_1	0.5
k_2	5
k_3	0.0005
k_4	0.005
k_5	0.05
k_6	100

不活性型と活性型の総和は保存

$$GEF_i + GEF_a = 1$$
$$GAP_i + GAP_a = 1$$
$$Ras_i + Ras_a = 1$$

＊ひな形をダウンロードして，Ras_model.mとして保存

http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3

“Ras_model.m”をダウンロードして空欄を埋める

その1/2

```
function Ras_model()
    % Rasモデル
    time = 0:100;
    param = [                ]; % [k1, k2, k3, k4, k5, k6]の順に
    y0 = [                ]; % [S, GEFa, GAPa, Rasa]の順に

    [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, param), time, y0);

    figure;
    plot(t_sim,timeCourse_sim(:,4));
    xlabel('time');
    ylabel('concentration');
    title('Rap1 model');

end
```

“Ras_model.m”をダウンロードして空欄を埋める

その2/2

```
function dydt = ODE(t, y, param)
    S = y(1);
    GEFa = y(2); % GEFi = 1 - GEFa
    GAPa = y(3); % GAPi = 1 - GAPa
    Rasa = y(4); % Rasi = 1 - Rasa

    k1 = param(1);
    k2 = param(2);
    k3 = param(3);
    k4 = param(4);
    k5 = param(5);
    k6 = param(6);


    dydt(1, :) = [                              ]; % S
    dydt(2, :) = [                              ]; % GEFa
    dydt(3, :) = [                              ]; % GAPa
    dydt(4, :) = [                              ]; % Rasa
end
```

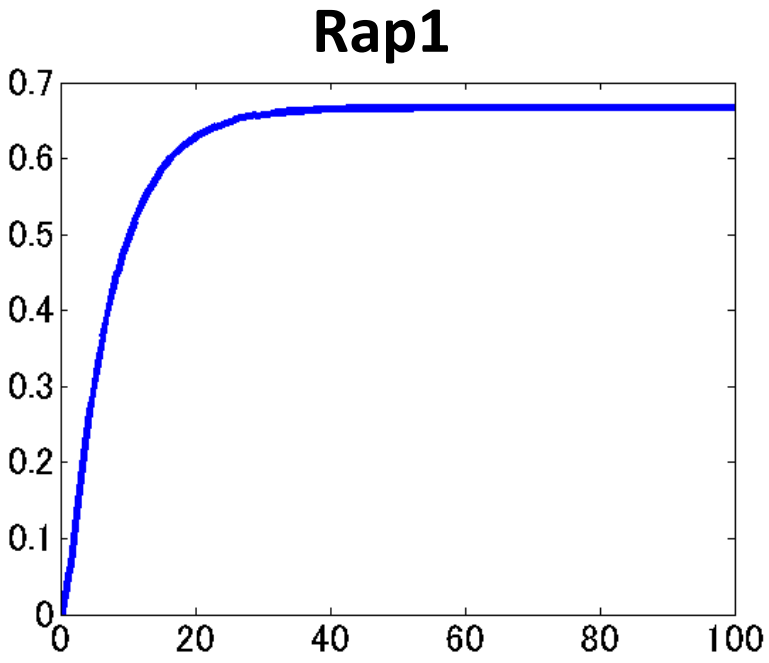
“Ras_model.m”をダウンロードして空欄を埋める

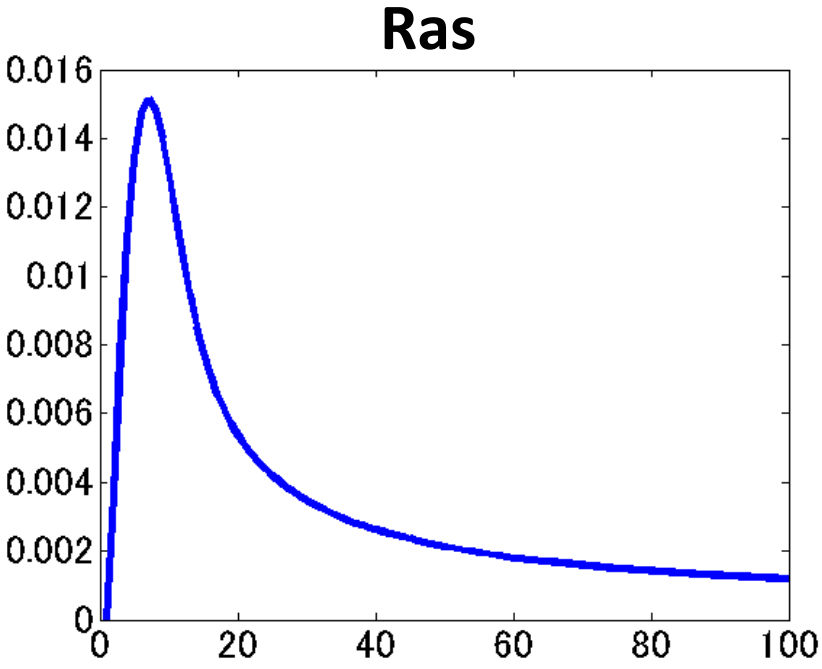
解答例

```
function Ras_model()
    % Rasモデル
    time = 0:100;
    param = [0.5, 5., 0.0005, 0.005, 0.05, 100.]; % [k1, k2, k3, k4, k5, k6]の順に
    y0 = [1, 0, 0, 0]; % [S, GEFa, GAPa, Ras]の順に
    （略）
end
function dydt = ODE(t, y, param)
    （略）
    dydt(1, :) = 0; % S
    dydt(2, :) = k1 * S * (1 - GEFa) - k2 * GEFa; % GEFa
    dydt(3, :) = k3 * S * (1 - GAPa) - k4 * GAPa; % GAPa
    dydt(4, :) = k5 * GEFa * (1 - Rasa) - k6 * Rasa * GAPa; % Rasa
end
```

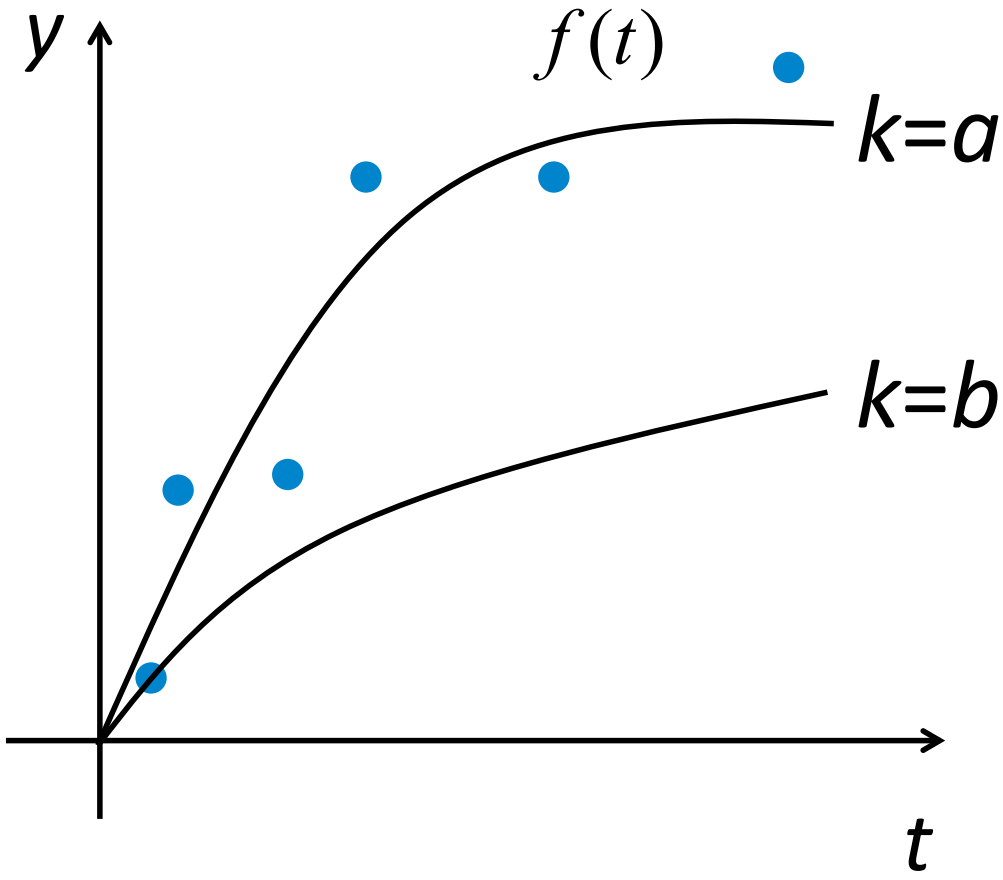
モデルの確認

波形をプロットして，モデルが正しく作れているか確認する.



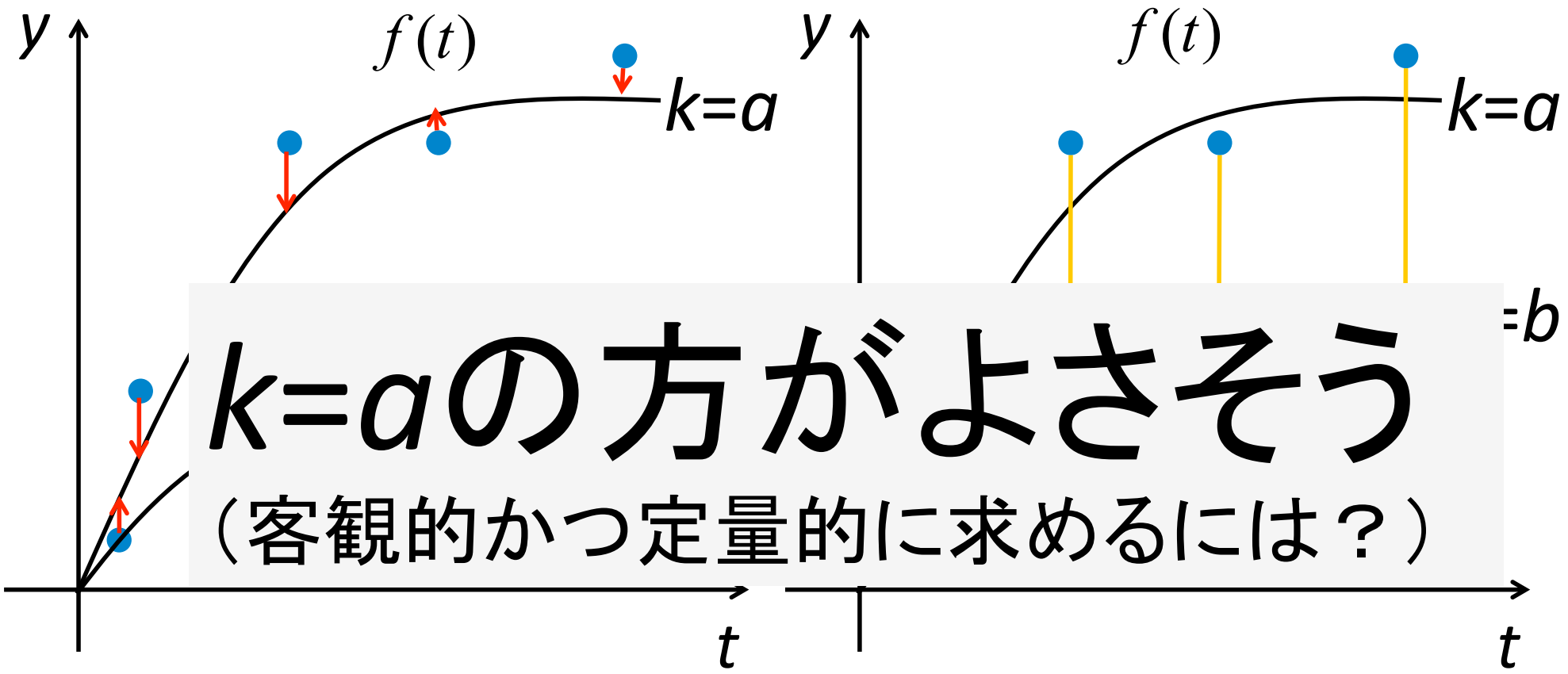


パラメータの決め方（1）



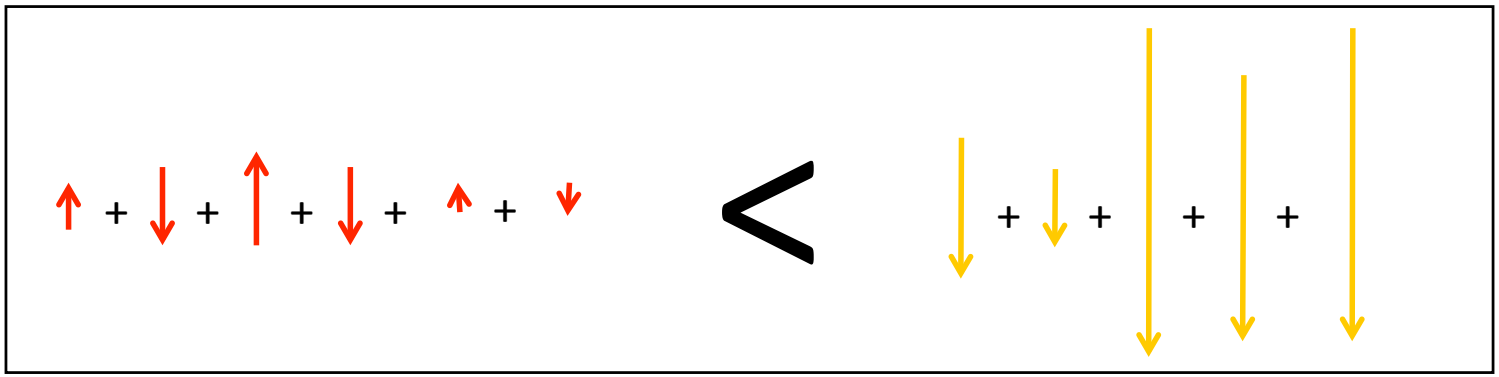
k=*a*と*k*=*b*，どっちがよいパラメータか？

パラメータの決め方（2）



k=aの方がよさそう

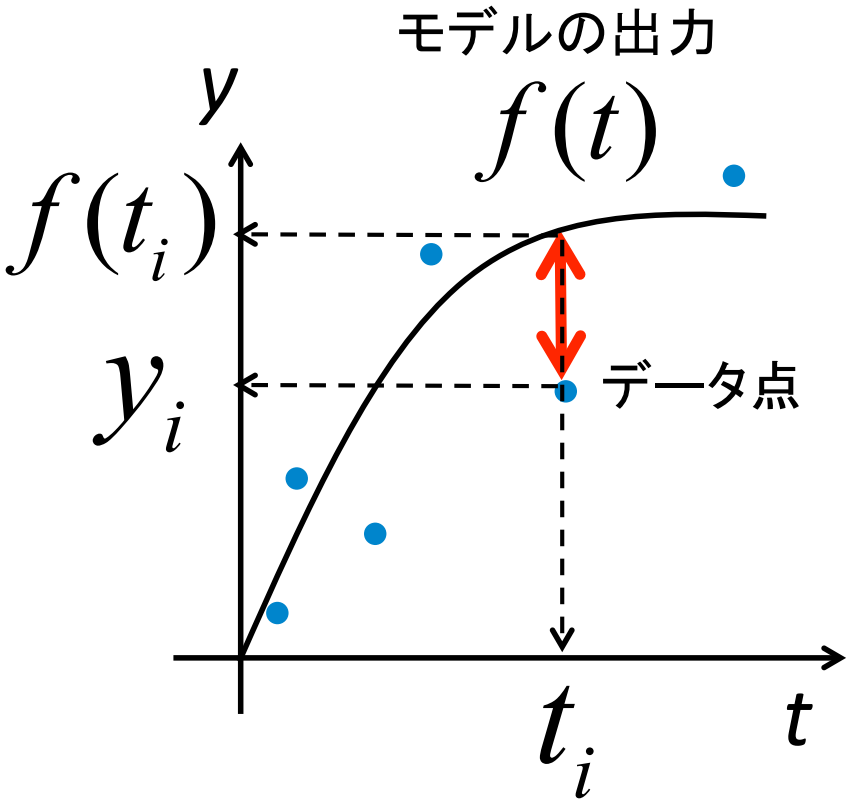
（客観的かつ定量的に求めるには？）



客観的かつ定量的な指標→評価関数

○残差　$y_i - f(t_i)$　モデルの出力（シミュレーションで得られる分子の時間波形）と，データ点の差

○残差2乗和　$\varepsilon^2 = \sum_{i=1}^{N} \left(y_i - f(t_i) \right)^2$　ユークリッド距離に相当



残差2乗和を評価関数として，最小にするようにパラメータを決める方法を**最小2乗法**という．

未知パラメータが1つの場合

手でRap1のパラメータk4を求めてみる

データ(ノイズなし)”data_Rap1_1para_noiseless.mat”をダウンロード
http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3

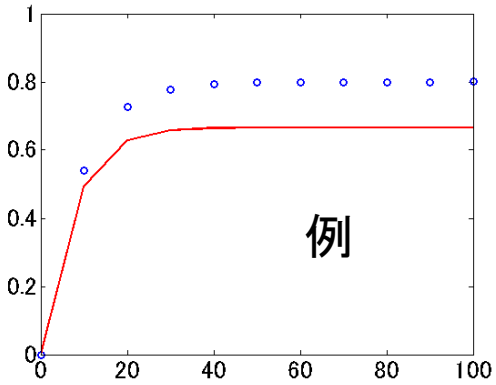
“Rap1_model.m”をコピーして，”hand_Rap1_1param.m”などとして改造

外から，引数でk4を与えるようにする．

k4を手で色々変えて，データにもっともよく合う値を探してみる．

```
function hand_Rap1_1param(k4)
    param = [0.5, 0.5, 0.2, □]; % [k1, k2, k3, k4]の順に
    y0 = [1, 0, 0.005, 0]; % [S, GEFa, GAPa, Rap1]の順に

    load('data_Rap1_1para_noiseless.mat');
    [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, param), time, y0);

    figure;
    plot(t, x, 'ro'); % データ点の表示
    hold on;
    plot(t_sim,timeCourse_sim(:, 4)); %シミュレーション結果の表示
    （略）
end
```

←表示部分を追加

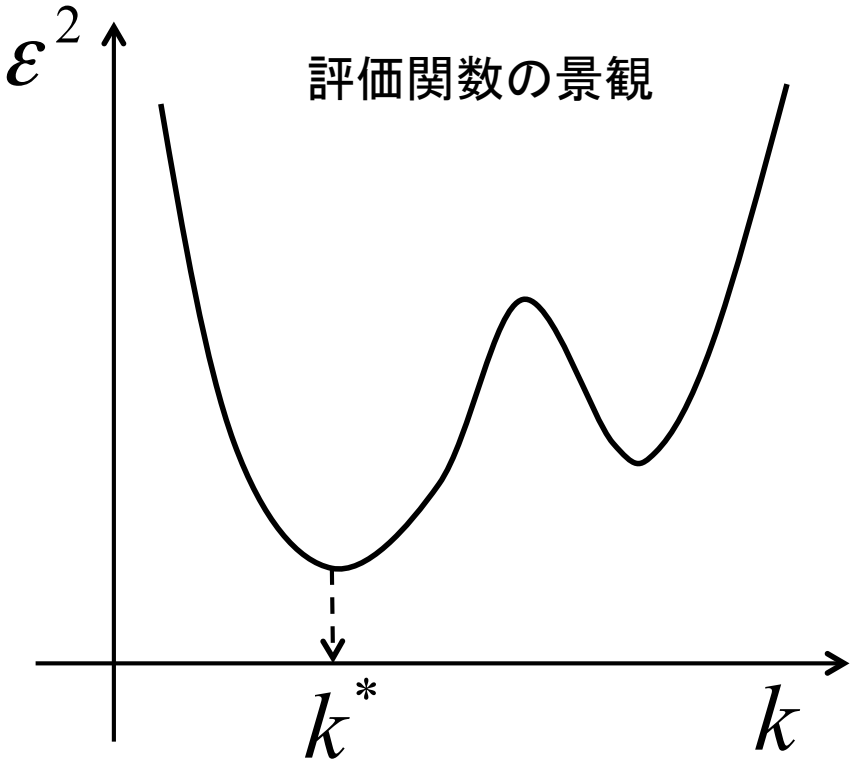


実行するときはコマンドウインドウ上から「hand_Rap1_1param(数値)」

Rap1モデルのパラメータk4を推定する（ノイズなし）

1．評価関数の景観を書く.

2．残差2乗和が最小となるときのパラメータkを求める.



パラメータkの値を少しずつ変えながら，残差2乗和を求める.
横軸にパラメータkの値，縦軸に残差2乗和ε^2をとってプロットする.

“fit_Rap1_1param.m”をダウンロードして空欄を埋める

その1/3

```
function [k4_est] = fit_Rap1_1param()

    param = [0.5, 0.5, 0.2, NaN]; % [k1, k2, k3, k4]の順に(k4は後で変えるので、仮にNaNを代入)
    y0 = [1, 0, 0.005, 0]; % [S, GEFa, GAPa, Rap1]

    load('data_Rap1_1para_noiseless.mat');

    %% k4を振って、残差2乗和(RSS)が最小となるときのk4を求める
     k4 = linspace(0.1, 10, 201);
    [x_Rap1_RSS] = landscape_Rap1(k4, t, x, param, y0); % t及びxはデータ点の時間とRap1aの波形

     [RSS_min, idx] = ____; % 最小値を探す関数minを使って、最小となるRSSとそのときの番号を求める
     k4_est = ____; % RSSが最小となるときのk4を代入。k4は配列であることに注意。

     %% 上で求めた残差2乗和(RSS)が最小となるときのk4を代入し、シミュレーション
     temp_param = param;
    temp_param(4) = ____; % k4は、RSS最小となるときの値を代入
     [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, temp_param), t, y0);
```

k4を0.1から10まで201分割刻みで動かし
残差2乗和（RSS）をサンプリング

fit_Rap1_1param()は次ページにも続く

“fit_Rap1_1param.m”をダウンロードして空欄を埋める

その2/3

```
%% 描画
figure(1);
%% 時間波形
subplot(1, 2, 1)
plot(t, x, 'o'); % データ点
hold on
plot(t_sim, timeCourse_sim(:, 4), ‘r-’); % シミュレーションの結果
%% RSSとk4の関係
subplot(1, 2, 2);
plot(k4, x_Rap1_RSS);
hold on
plot(k4_est, RSS_min, 'ro');

end
```

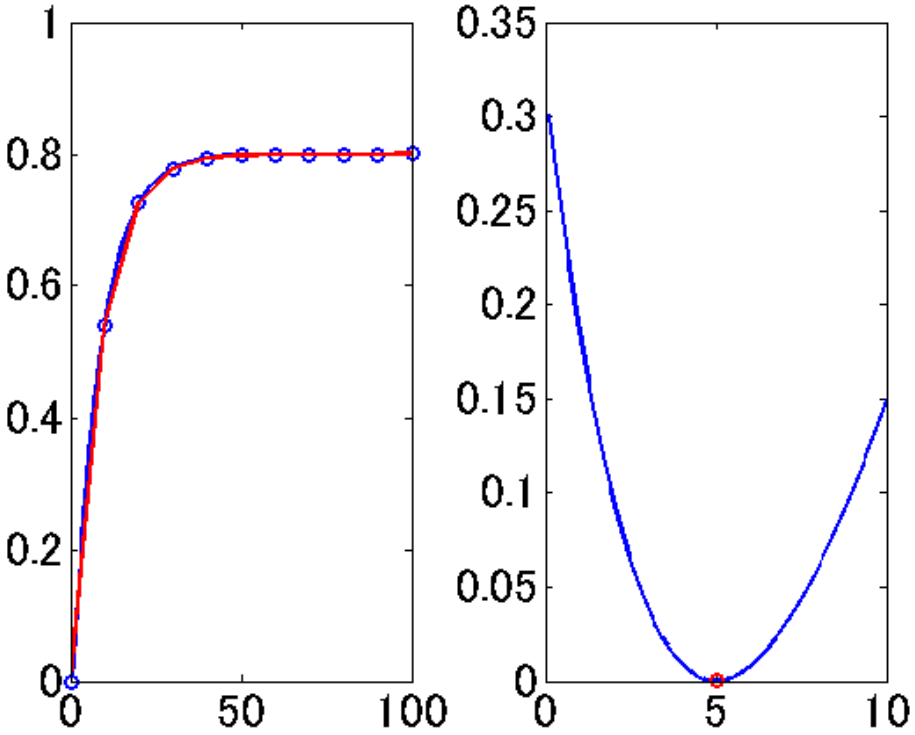
“fit_Rap1_1param.m”をダウンロードして空欄を埋める

その3/3

```
function [x_Rap1_RSS] = landscape_Rap1(k4, SamplingTime, x_Rap1, param, y0)

   vary_x_Rap1 = zeros(length(x_Rap1), length(k4)); % 各列毎に各k4でのRap1の時間波形が入るような配列を準備
   for i = 1:length(k4)
        param(4) = [      ]; % 変えているパラメータは？配列で与えられていることに注意
        [~, tc] = ode15s(@(t, y) ODE(t, y, param), SamplingTime, y0);
        vary_x_Rap1(:, i) = tc(:, 4);
   end
   x_Rap1_Residual = vary_x_Rap1 - x_Rap1 * ones(1, size(vary_x_Rap1, 2));
   x_Rap1_RSS = [                    ]; % 残差はx_Rap1_Residual。残差2乗和はどう表すか？
end

function dydt = ODE(t, y, param)
   % ここにモデルをコピー
end
```

正解

データとODEの出力　　評価関数の景観



K4 = 5.005

（真のモデルは，k4 = 5.0）

“fit_Rap1_1param.m”をダウンロードして空欄を埋める

解答例 その1/2

```
function [k4_est] = fit_Rap1_1param()

   （略）
   %% k4を振って、残差2乗和(RSS)が最小となるときのk4を求める
    k4 = linspace(0.1, 10, 201);
   [x_Rap1_RSS] = landscape_Rap1(k4, t, x, param, y0); % t及びxはデータ点の時間とRap1aの波形

    [RSS_min,idx] = min(x_Rap1_RSS); % 最小値を探す関数minを使って、最小となるRSSとそのときの番号を
求める
    k4_est = k4(idx); % RSSが最小となるときのk4を代入。k4は配列であることに注意。

    %% 上で求めた残差2乗和(RSS)が最小となるときのk4を代入し、シミュレーション
    temp_param = param;
    temp_param(4) = k4_est; % k4は、RSS最小となるときの値を代入
    [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, temp_param), t, y0);
   （略）
end
```

“fit_Rap1_1param.m”をダウンロードして空欄を埋める

解答例 その2/2

```
function [x_Rap1_RSS] = landscape_Rap1(k4, SamplingTime, x_Rap1, param, y0)

   vary_x_Rap1 = zeros(length(x_Rap1), length(k4)); % 各列毎に各k4でのRap1の時間波形が入るような配列を準備
   for i = 1:length(k4)
        param(4) = k4(i); % 変えているパラメータは？配列で与えられていることに注意
        [~, tc] = ode15s(@(t, y) ODE(t, y, param), SamplingTime, y0);
        vary_x_Rap1(:, i) = tc(:, 4);
   end
   x_Rap1_Residual = vary_x_Rap1 - x_Rap1 * ones(1, size(vary_x_Rap1, 2));
   x_Rap1_RSS = sum(x_Rap1_Residual.^2); % 残差はx_Rap1_Residual。残差2乗和はどう表すか？
end

function dydt = ODE(t, y,param)
   % ここにモデルをコピー
   （略）
end
```

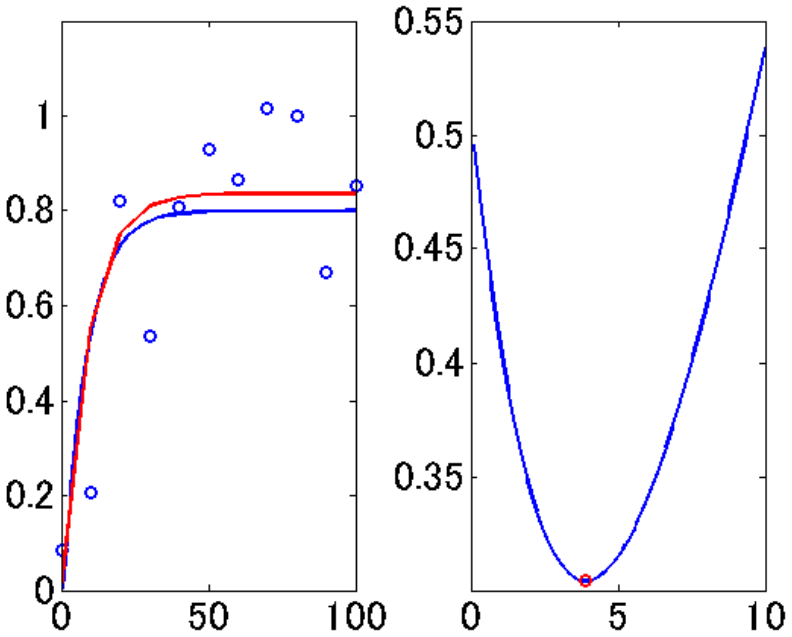
Rap1モデルのパラメータk4を推定する（ノイズあり）

1. データ(ノイズあり) ”data_Rap1_1para_noiseless.mat”をダウンロード
http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3

2. 評価関数の景観を書く.

3. 残差2乗和が最小となるときのパラメータ*k*を求める.

データとODEの出力　　評価関数の景観



K4 = 3.9115?

読み込むデータの変更

```
function [k4_est] = fit_Rap1_1param()

    param = [0.5, 0.5, 0.2, NaN]; % [k1, k2, k3, k4]の順に(k4は後で変えるので、仮にNaNを代入)
    y0 = [1, 0, 0.005, 0]; % [S, GEFa, GAPa, Rap1]

    % load('data_Rap1_1para_noiseless.mat');
    load('data_Rap1_1para_noise.mat');

    %% k4を振って、残差2乗和(RSS)が最小となるときのk4を求める
     k4 = linspace(0.1, 10, 201);
    [x_Rap1_RSS] = landscape_Rap1(k4, t, x, param, y0); % t及びxはデータ点の時間とRap1aの波
形
```

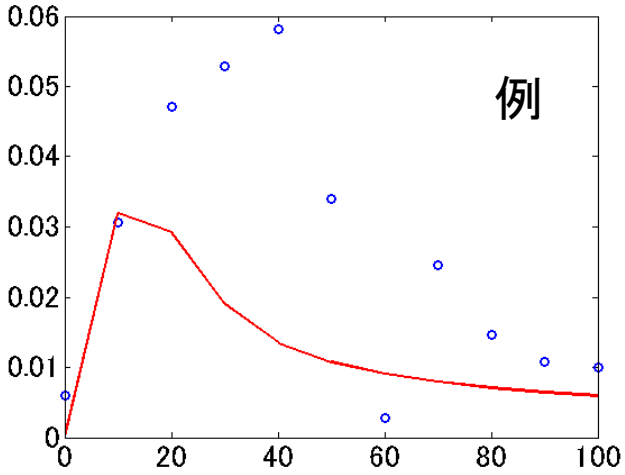
手でRasのパラメータk6を求めてみる

データ(ノイズなし)”data_Ras_1para_noise”をダウンロード
http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3

“Ras_model.m”をコピーして，”hand_Ras_1param.m”などとして改造

外から，引数でk6を与えるようにする．

Rap1モデルの時と同様に
k6を手で色々変えて，データにもっともよく合う値を探してみる．

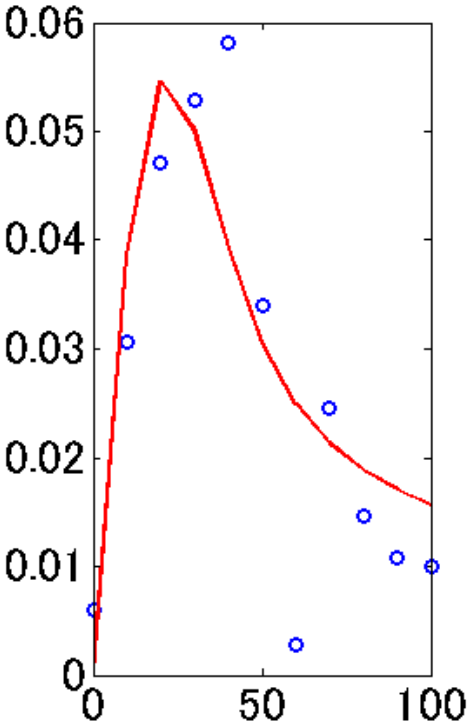


実行するときはコマンドウインドウ上から「hand_Ras_1param(‘数値’)」

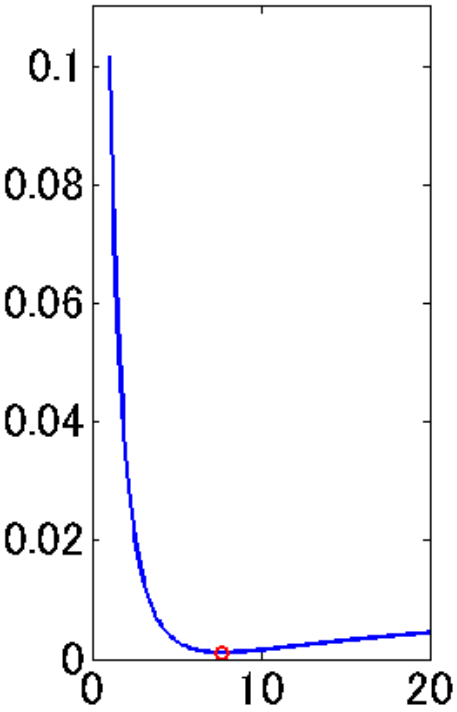
Rasモデルのパラメータk6を推定する（ノイズあり）

1. 評価関数の景観を書く.
2. 残差2乗和が最小となるときのパラメータkを求める.

データとODEの出力



評価関数の景観



手で求めた値と
比べてどうか？

“fit_Ras_1param.m”をダウンロードして空欄を埋める

その1/3

```
function [k6_est] = fit_Ras_1param()

   param = [0.5, 5, 0.0005, 0.005, 0.05, NaN]; % [k1, k2, k3, k4, k5, k6]の順に(k6は後で変えるので、仮にNaNを代入)
   y0 = [1, 0, 0, 0]; % [S, GEFa, GAPa, Ras]

   load('data_Ras_1para_noise.mat');

   %% k6を振って、残差2乗和(RSS)が最小となるときのk6を求める
    k6 = linspace(1, 20, 201);
   [x_Ras_RSS] = landscape_Ras(k6, t, x, param, y0); % t及びxはデータ点の時間とRasの波形

    [RSS_min, idx] = [          ]; % 最小値を探す関数minを使って、最小となるRSSとそのときの番号を求める
    k6_est = [      ]; % RSSが最小となるときのk6を代入。k6は配列であることに注意。

    %% 上で求めた残差2乗和(RSS)が最小となるときのk6を代入し、シミュレーション
    temp_param = param;
    temp_param( [ ] ) = [      ]; % k6は、RSS最小となるときの値を代入
    [t_sim,timeCourse_sim] = ode15s(@(t, y) ODE(t, y, temp_param), t, y0);
```

k6を1から10まで201分割刻みで動かし
残差2乗和（RSS）をサンプリング

fit_Ras_1param()は次ページにも続く

“fit_Ras_1param.m”をダウンロードして空欄を埋める

その2/3

```
%% 描画
figure(1);
%% 時間波形
subplot(1, 2, 1)
plot(t, x, 'o'); % データ点
hold on
plot(t_sim, timeCourse_sim(:, 4), ‘r-’); % シミュレーションの結果
%% RSSとk6の関係
subplot(1, 2, 2);
plot(k6, x_Ras_RSS);
hold on
plot(k6_est, RSS_min, 'ro');

end
```

“fit_Ras_1param.m”をダウンロードして空欄を埋める

その3/3

```
function [x_Ras_RSS] = landscape_Ras(k6, SamplingTime, x_Ras, param, y0)

   vary_x_Ras = zeros(length(x_Ras), length(k6)); % 各列毎に各k6でのRasの時間波形が入るような配列を準備
   for i = 1:length(k6)
        param(  ) =  ; % 変えているパラメータは？配列で与えられていることに注意
        [~, tc] = ode15s(@(t, y) ODE(t, y, param), SamplingTime, y0);
        vary_x_Ras(:, i) = tc(:, 4);
   end
   x_Ras_Residual = vary_x_Ras - x_Ras * ones(1, size(vary_x_Ras, 2));
   x_Ras_RSS =  ; % 残差はx_Rap1_Residual。残差2乗和はどう表すか？
end

function dydt = ODE(t, y, param)
   % ここにモデルをコピー
end
```

“fit_Ras_1param.m”をダウンロードして空欄を埋める

解答例 その1/2

```
function [k6_est] = fit_Ras_1param()

   （略）
   %% k6を振って、残差2乗和(RSS)が最小となるときのk6を求める
    k6 = linspace(0.1, 10, 201);
   [x_Ras_RSS] = landscape_Ras(k6, t, x, param, y0); % t及びxはデータ点の時間とRasaの波形

    [RSS_min,idx] = min(x_Ras_RSS); % 最小値を探す関数minを使って、最小となるRSSとそのときの番号を
求める
    k6_est = k6(idx); % RSSが最小となるときのk6を代入。k6は配列であることに注意。

    %% 上で求めた残差2乗和(RSS)が最小となるときのk6を代入し、シミュレーション
    temp_param = param;
    temp_param(6) = k6_est; % k6は、RSS最小となるときの値を代入
    [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, temp_param), t, y0);
   （略）
end
```

“fit_Ras_1param.m”をダウンロードして空欄を埋める

解答例 その2/2

```
function [x_Rap1_RSS] = landscape_Rap1(k6, SamplingTime, x_Ras, param, y0)

   vary_x_Rap1 = zeros(length(x_Rap1), length(k6)); % 各列毎に各k6でのRasの時間波形が入るような配列を準備
   for i = 1:length(k6)
        param(6) = k6(i); % 変えているパラメータは？配列で与えられていることに注意
        [~, tc] = ode15s(@(t, y) ODE(t, y, param), SamplingTime, y0);
        vary_x_Rap1(:, i) = tc(:, 4);
   end
   x_Ras_Residual = vary_x_Ras - x_Ras * ones(1, size(vary_x_Ras, 2));
   x_Ras_RSS = sum(x_Ras_Residual.^2); % 残差はx_Ras_Residual。残差2乗和はどう表すか？
end

function dydt = ODE(t, y,param)
   % ここにモデルをコピー
   （略）
end
```

議論

- 手と数値探索の結果は，どれぐらい一致しましたか？
- ノイズがデータにのると，パラメータの値はどうなりましたか？

未知パラメータが複数の場合

手でRap1の4パラメータを合わせる？

“hand_Rap1_1param.m”をコピーして，”hand_Rap1_4param.m”などとして改造する.

```
function hand_Rap1_1param(k)
    param = [ ]; % [k1, k2, k3, k4]の順に
    y0 = [1, 0, 0.005, 0]; % [S, GEFa, GAPa, Rap1]の順に

    load('data_Rap1_4para.mat');
    [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, param), time, y0);

    figure;
    for i = 1:2
        subplot(1, 2, i);
        plot(t, x(:, i*2), 'ro'); % データ点の表示
        hold on;
        plot(t_sim,timeCourse_sim(:, i * 2)); % シミュレーション結果の表示
    end
    （略）
end
```

※　手で合わせるのが大変なことを確認する程度でよい．真面目に合わせる必要はない．

GEFaとRap1a

↑表示部分を追加

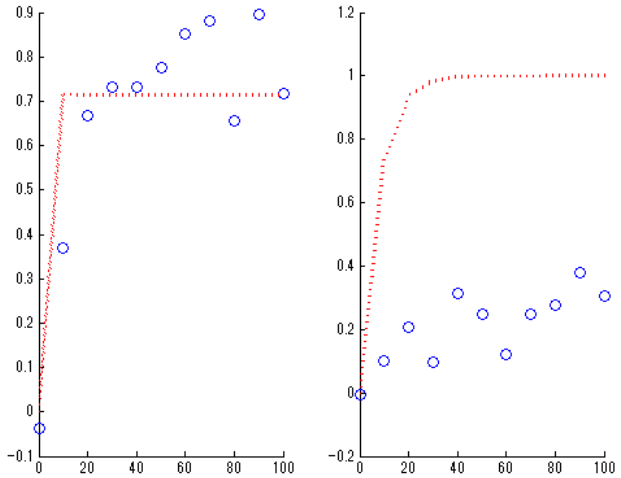
実行時：hand_Rap1_1param([数値, 数値, 数値, 数値])

[ ]を忘れないように

http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3

正解コード

```
function hand_Rap1_1param(k)
    param = k; % [k1, k2, k3, k4]の順に
    y0 = [1, 0, 0.005, 0]; % [S, GEFa, GAPa, Rap1]の順に

    load('data_Rap1_4para.mat');
    [t_sim, timeCourse_sim] = ode15s(@(t, y) ODE(t, y, param), time, y0);

    figure;
    for i = 1:2
        subplot(1, 2, i);
        plot(t, x(:, i*2), 'ro'); % データ点の表示
        hold on;
        plot(t_sim,timeCourse_sim(:, i * 2)); % シミュレーション結果の表示
    end
    （略）
end
```



Rap1のパラメータを進化計算を用いて探索する

パラメータの数が少ない → for loopでしらみつぶし
パラメータ数が多い → しらみつぶしに探すのは難しい
単純に，パラメータの数だけfor loopを回すと…？
→（パラメータ候補の数）$^{(パラメータ数)}$：膨大な回数

どうやってデータの再現性の高いパラメータの候補を探すか？

進化計算と呼ばれる分野の
最適化計算法のひとつである
“**Evolutionary Programming（EP）**”
を使って、効率的にパラメータを探索する

ひな形とcopasiep.mをダウンロード
http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3

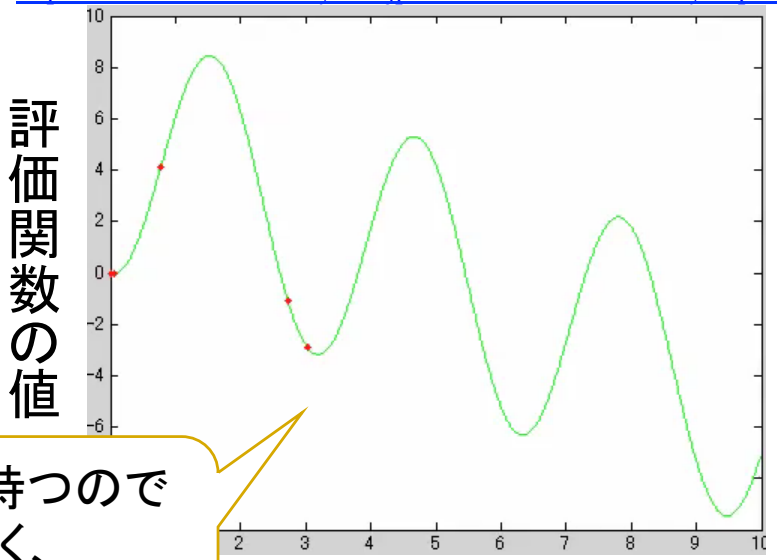
Evolutionary programming（EP）

Evolutionary programming(EP, 進化的計算)は
生物学における自然淘汰を模して作られた
メタヒューリスティックな
（（ある程度）汎用性のある経験的な）
最適化アルゴリズム

今回は評価関数（残差2乗和）が
最小となるようなODEのパラメータ
を得るためにEPを使う

EPのデモ

http://kurodalab.bi.s.u-tokyo.ac.jp/class/Summer/2014/Day3/ep.mp4



広い探索範囲を持つので
極小値に陥りにくく、
最小値に到達しやすい

〇大雑把な説明

1) 「パラメータの組」を1個体とみなして，最初に*N*個の個体が適当につくる
2) 各個体が子をつくる．各子には確率的な変異(mutation)が入る ⇒ 2*N*個体
3) 残差2乗和が小さいパラメータの組 → 優れている個体 → 生き残る
   残差2乗和が大きいパラメータの組 → 劣っている個体 → 淘汰
   → 個体数を*N*個にする．
4) 2)と3)を十分長く繰り返す
5) 最終的に最も優れた個体を，パラメータとして採用

もう少し詳細なEPのアルゴリズム

1. 乱数を初期値として，N 個の個体（解の候補）を生成する.
2. 個体のそれぞれのコピーをつくる.
3. 2.で作った各コピーに正規乱数を加える.（変異）
4. 3.の操作で作られた新しい N 個の個体と，元の個体を混ぜた $2N$ 個の各個体に対してスコアを求める. スコアは以下のように決める.
   a. スコアを決める対象の個体を除いて，$2N$個体の集団から無作為に q 個体を選ぶ.
   b. 選んだ q 個体のうち，スコアを決める対象の個体より評価関数の値（RSSなど）が悪い個体（最小2乗法の場合は，対象の個体よりRSSの値が大きい個体）の数をその個体のスコアとする.
5. スコアの下位 N 個の個体を削除する.（淘汰）
6. 終了条件を満たすまで2. ～7.（1世代に相当）の操作を繰り返す.
7. 残った中で最も評価関数の値が良い個体を“解”とする.

※EPには，バリエーションがいくつか存在する.

“fitEP_Rap1_4param.m”をダウンロードして空欄を埋める

その1/5

```
function [res_bestIndv, res_bestScore] = fitEP_Rap1_4para(n, numGeneration)
    % Rap1モデルのパラメータをEPで推定するモデル。
    % 引数のnは系列数。numGenerationは世代数。

    res_bestIndv = zeros(n, 4); % 結果(パラメータ)を格納する変数
    res_bestScore = zeros(n, 1); % 結果(RSS)を格納する変数

    for i = 1:n
        % 各系列で最小のRSSとなるパラメータを探し出し、変数に格納
        [res_bestIndv(i, :), res_bestScore(i, :)] = work_EP_Rap1(numGeneration);
    end
end
```

“fitEP_Rap1_4param.m”をダウンロードして空欄を埋める

その2/5

```
function [bestIndv, bestScore, report] = work_EP_Rap1(numGeneration)
    y0 = [1, 0, 0.005, 0]; % [S, GEFa, GAPa, Rap1]
    param = zeros(1, 4);
    load('data_Rap1_4para.mat');

    %%%%  EPパートここから  %%%%
    currentTime = clock; % 現在の時刻を取得
    second = currentTime(6); % 現在の時刻から’秒’を取得

    strSeed = '2014';
    seed = str2num(strSeed)*second*1000; % 乱数の種に‘秒’をかけることで、よりランダムに
    RandStream.setDefaultStream(RandStream(‘mt19937ar’, ‘Seed’, seed)); % 乱数の初期設定
    % RandStream.setGlobalStream(RandStream(‘mt19937ar’, ‘Seed’, seed)); % ver. 2013以降

    lb = 1e-3 * ones(size(param)); % パラメータ探索範囲の下限
    ub = 1e+2 * ones(size(param)); % パラメータ探索範囲の上限

    numParents = 20; % 子を生成する親の数
    [bestIndv, bestScore, report] = copasiep(@(param) cal_RSS(x, t, param, y0), ...
        numParents, numGeneration, lb, ub); % copasiのEPでパラメータサーチ
    %%%%  EPパートここまで  %%%%
```

“fitEP_Rap1_4param.m”をダウンロードして空欄を埋める

その3/5

```
%% 各系列で最小のRSSのときのパラメータを用いてシミュレーション
time = [0, 100];
[t_est, x_est] = ode15s(@(t, y) ODE(t, y, bestIndv), time, y0);

% データ点とシミュレーション結果の重ね描き
figure(1);
for i = 1:2
    subplot(1, 2, i);
    hold on
    plot(t, x(:, i*2), ‘o’); % データ点
    hold on
    plot(                , ‘r:’); % シミュレーション結果
    if i == 1
        title('GEF_a');
    else
        title('Rap1');
    end
end
```

“fitEP_Rap1_4param.m”をダウンロードして空欄を埋める

その4/5

```
    % EPでRSSが減少して行く様子を描画
    figure(2);
    subplot(1, 2, 1);
    hold on
    plot(report(:, 1), 'b-');
    title('最も良い個体');
    xlabel('世代数');
    ylabel('残差2乗和');

    subplot(1, 2, 2);
    hold on
    plot(report(:, 2), 'b:');
    title('全個体の平均');
    xlabel('世代数');
    ylabel('残差2乗和');
end
```

“fitEP_Rap1_4param.m”をダウンロードして空欄を埋める

その5/5

```
function [RSS] = cal_RSS(x_data, SamplingTime, param, y0)
    % 残差2乗和を計算
    [~, tc] = ode15s(@(t, y) ODE(t, y, param), SamplingTime, y0);
    residual = tc - x_data; % 残差
    RSS =                   ; % 残差2乗和
end

function dydt = ODE(t, y, param)
    % ここにモデルをコピー
    
end
```

EPの他にどんな方法があるか

・勾配を用いる方法

基本的には，ニュートン法の類，共役勾配法など．
また，拘束条件の有無によってアプローチが異なってくる．
凸最適化問題に向いている．

・確率を用いる方法

- シミュレーテッドアニーリング（SA）
  熱力学的な焼きなましの過程をまねた方法
- 遺伝的アルゴリズム（GA） 交叉 + 突然変異
- 進化的プログラミング（EP） 主に突然変異
- 分布推定アルゴリズム(EDA) （近年の主流？）
  「分布から解候補の生成」と「分布の推定」を繰り返す

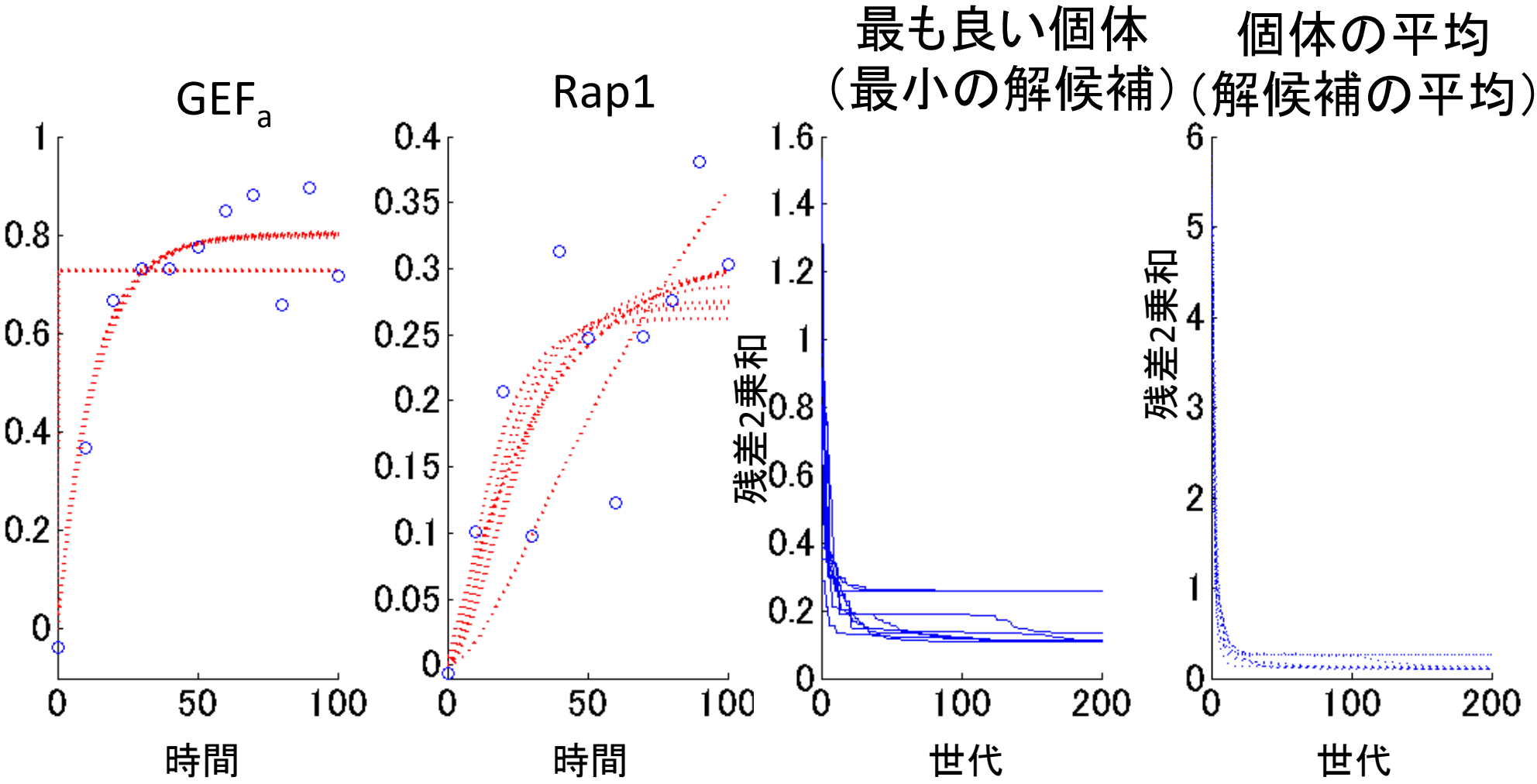
※ 勾配法と確率の両方を用いるハイブリッドもある．

注意点・補足事項

- 1次元空間の探索 は，しらみつぶしにやるより効率的な方法がある．⇒　黄金比探索法など．
- 手元のデータを最もよく再現できるパラメータが，最もよいパラメータとは限らない．（なぜか？）
- 今回は，どのデータにどのモデルを使うかは予めわかっていた．しかし，現実問題としては，そのようなことは，文献などからその系について事前に信頼性の高い情報がある場合に限られる．データからモデルを決めるには，統計的モデル選択規準がある．

実行結果の例

回数：n=10, 世代数：numGeneration=200などとして，実行.



ワークスペースの変数をクリックして，”res_bestIndv”, ”res_bestScore”をみて，
全体で最もよい解を確認する.（*※実行時に出力を格納するようにする.*）

“fitEP_Rap1_4param.m”をダウンロードして空欄を埋める

解答例 その3/5の部分

```
%% 各系列で最小のRSSのときのパラメータを用いてシミュレーション
time = [0, 100];
[t_est, x_est] = ode15s(@(t, y) ODE(t, y, bestIndv), time, y0);

% データ点とシミュレーション結果の重ね描き
figure(1);
for i = 1:2
    subplot(1, 2, i);
    hold on
    plot(t, x(:, i*2), ‘o’); % データ点
    hold on
    plot(t_est, x_est(:, i*2), ‘r:’); % シミュレーション結果
    if i == 1
        title('GEF_a');
    else
        title('Rap1');
    end
end
```

“fitEP_Rap1_4param.m”をダウンロードして空欄を埋める

解答例 その5/5の部分

```
function [RSS] = cal_RSS(x_data, SamplingTime, param, y0)
    % 残差2乗和を計算
    [~, tc] = ode15s(@(t, y) ODE(t, y, param), SamplingTime, y0);
    residual = tc - x_data; % 残差
    RSS = sum(residual(:).^2); % 残差2乗和
end

function dydt = ODE(t, y, param)
    % ここにモデルをコピー
    （略）
end
```

data_Rap1_4para.matを生成したモデルの真のパラメータ

```
param = [0.05, 0.01, 0.02, 7];
y0 = [1, 0, 0.005, 0]; % [S,GEFa,GAPa,Rap1a]
```

ノイズを加えたもの

議論

- パラメータはどれくらい合っていたか？
- 実行ごとに，結果が異なるのはなぜか？
- EPで，残差2乗和が変化しなくなったら，計算をやめてもよいか？
- パラメータの数の増加に対して，解を探索すべき対象となる領域は，どのように増えるか？
- そもそも，なぜEPを使うのか？

議論の返答例

- EPで，残差2乗和が変化しなくなったら，計算をやめてもよいか？ ⇒確率的な探索なので，まだよりよい解が生成される可能性がある．一般に，どれぐらいで止めればよいかは判断が難しく，現在は，経験に頼る部分が大きい．
- パラメータの数Nの増加に対して，解を探索すべき対象となる領域は，どのように増えるか？ ⇒1パラメータの探索領域をM分割すると，M^N点を探索する必要がある．例えば，N=10, M=200で24ケタ程度の大きさ．M=200でN=44以上では100ケタを越える．
- そもそも，なぜEPを使うのか？ ⇒探索空間が高次元で局所最適解が多い問題では，広域最適解を得るのは，一般には不可能．効率的に準最適解を見つける手段として，ここではEPを用いた．確率的な探索は，収束が判定しづらいなどのデメリットもあるが，よくない局所最適解に捕らわれにくいなどのメリットもある．効率的に解を探索できるなら，EPでなくてもよい．

発展課題

発展課題1

- RasモデルでRap1データをfitしてみる
  - Rap1モデルとRasモデルの関係は，どうなっているか？
  - Rap1モデルでRap1データをfitした結果と，Rasモデルでの Rap1データをfitした結果を比べて，どのような違いがあるか？
  - ノイズのある有限データから，RasモデルとRap1モデルを判別するにはどうすればよいか？

発展問題2

- Rap1モデル，Rasモデルの**ODE**を手で解いて，解析解を求める.

発展問題3：オイラー法

（特に時間が余った人向け，本編とは直接関係ない問題です．）

☆ODEを数値的に解く非常に簡単な方法：

$$\frac{dx}{dt} = f\left(x,t\right) \approx \frac{x_{n+1} - x_n}{h}$$

と近似して，

$$x_{n+1} \approx x_n + hf\left(x_n, t_n\right)$$

を計算する解法があり，オイラー法と呼ばれている．

オイラー法を実装して，1次反応モデルなど
の簡単なODEを解いてみる．

※数値解法には様々な手法があり，問題に応じて適切な数値解法を用いる必要がある．オイラー法はODEの数値解法の初歩を学ぶためには有用だが，実用性は高くない．

オイラー法（陽解法）の例

```
function [] = ode_euler(tau)

t0 = [0,100];
h = 0.1;

n = ceil((t0(2)-t0(1))/h);

x0 = 10;

x = zeros(n+1,1);
x(1) = x0;
t = linspace(t0(1),t0(2),length(x));

for a = 1:length(x)-1

    x(a+1) = x(a) + h * ode_fun(x(a),t(a),tau);
end

x_ana = x0 * exp(-t/tau);
plot(t,x,t,x_ana)

end

function [y] = ode_fun(x,t,tau)

y = -x / tau;

end
```

data_Rap1_4paraの正解パラメータ

```
param = [0.05,0.01,0.02,7];
y0 = [1,0,0.005,0]; % [S,GEFa,GAPa,Rap1a]
```

data_Ras_4paraの正解パラメータ

```
param = [0.01,0.1,0.008,0.004,2.5,5]; % [k1,k2,k3,k4,k5,k6]
y0 = [1,0,0,0]; % [S,GEFa,GAPa,Rasa]
```