

進化機構の進化とホメオカオス

池上高志

神戸大学自然科学研究科

1 はじめに

実際の生命現象に現われるデータをもとにモデルを構築しそのモデルの数理的側面も研究していくというアプローチをここでは従来の数理生物学と呼び、計算機のなかに構築される人工生命のアプローチと対比してその違いについて簡単に触れてみたい。

人工生命の研究と従来の数理生物の研究との決定的な違いは、「現実」との距離の取り方、フィードバックのかけかたにある。現実のデータとあわないとシミュレーションは間違っているとされる後者の評価基準に対し、人工生命の研究はむしろ現実との食い違いを重要視していく。

例えば人工生命の研究では、現実世界を構成する物質的制約から切り離された生命の可能なロジックを発見しようとする。また生物の進化や認識を一種の計算過程とみなして解釈し直なおそうと考えるのである。こういう視点に立った研究は計算機という人工世界のなかで初めて可能となるものである。

人工世界の研究はしたがって、現実とは関係のないものだという批判を受けるのだが、人工世界で発見される新しい現象や概念は、われわれの現実に対する見方をいやおうなく変更していくものなのである。

もちろんこういった研究は計算機の出現とともに始まったものがある。例えば物理においては、磁石のガラス状態であるスピングラスから生れた「凸凹な風景」という考え方や、天気予報やルーレットの予測が難しいのはなぜか、ということから生れた「カオス」は、生物や化学に進出して活躍している概念である。これらの概念は現実の現象のなかにヒントをえて見出されたものであるが、その後人工世界の中で切磋琢磨されて深みが増した概念だと思ふ。数学は数学的には厳密でなくても「新しい現象や概念」を提供してくれるものとして物理学に期待する。物理学や生物学は、計算機の中に表現された人工世界に新しい概念や現象を期待しはじめていると考えられないだろうか。

しかし人工生命も従来の数理生物の研究もおもしろいと思ふ対象は同じである。例えばそれは、複雑な社会構造の起源であったり進化のしかたの進化機構であったりする。これらの現象の説明をどういう形で理解しようとするかというのが第二の相違点となる。ひとつの例として宿主と寄生者の相互作用を考えてみよう。

寄生関係にある宿主(宿主)とパラサイト(寄生者)は、パラサイトが宿主から搾取する状態である。寄生状態に置かれた宿主はパラサイトから逃れようとし、パラサイトは宿主を追撃しようとする。そのひとつの戦略が宿主は突然変異によって遺伝子型を変化させることでパラサイトに攻撃されないものに変わるというものである。ここで考える突然変異は、遺伝子の複製に伴う校正手続きの精度と考える。そのため一定の変異率を保つことは生物にとってパラドクスに満ちたこととなってしまう。生物にとって自己複製こそ第一目的であるならば、複製のエラーはできるかぎり押さえたいものである。しかしこの複製のエラーが進化の種(たね)となり、ホス

トはパラサイトから逃れることができるのだから複製エラーがゼロになってしまっても困るのである。さらに、もし生物が積極的に複製エラーの大きさを自分の遺伝子上にコードしているとしたら、エラーの大きさを決めていた部分もまた自分自身の変異率によって突然変異にさらされるという逆説的な状況が出現してしまう。この逆説的な状況は、突然変異率をゼロにするという解決策しかないように見える。しかしこの小論で示されるようにカオスによる有意な突然変異率の維持は可能である。カオスによる突然変異率の維持を可能にした系はネットワークとしての共生関係を進化させるようになる。またこのカオスのふるまいをビット・スペースで見ることで遺伝的アルゴリズムとの関係が生まれ、遺伝子コードの進化に対するモデルがみえてくる。このような状況を計算機でシミュレートするとともに、すこに見出される並列分散処理システムとしての可能性を探り、ホランドによって提案された並列分散処理系としてのクラシファイヤー・システムとの対応付けも可能となる。

この議論の前半(突然変異の意味や、そのメタ変異機構)は従来の数理生物も持ってきた問題意識である。その解決の仕方をカオスを介したダイナミクスとしての理解にもっていき、さらに計算過程としての進化ダイナミクスに言及してみようとするのが人工生命としてのアプローチであろう。これに対し例えばシステム(ホストやパラサイト)の経済効率という視点から最適化問題(多くの場合平衡系)にもっていくことが従来の数理生物には多かったように思われる。

しかし最近ハミルトンらによる性の進化をカオスで説明しようという動きもあって違いはそんなに明白でもなくなりつつある。残念ながら研究会ではこれらの違いについてのつっこんだ議論にはならなかった。

アウトライン

この小論では人工生命の立場から、共生進化のモデルと突然変異のダイナミクスについて議論する。特にホランドによって提案された並列分散処理系としてのクラシファイヤー・システムとの対応付けにも言及し、自動化されたGAという側面を強調して議論してみたい。そこで§2では簡単にクラシファイヤー・システムの紹介をする。クラシファイヤーシステムで重要な考え方は、バケット・ブリゲード法とスキーマの進化にある。つぎに§3ではホストとパラサイト系の力学モデルについて紹介する[2, 3, 4]。ここでは遺伝子は1次元のビット配列として定式化される。このモデルでの重要な観点は突然変異率の突然変異という考え方である。§4ではシミュレーションの結果、高い突然変異率の進化と維持には大自由度の弱いカオスが働いていることが分かる。このカオスを系の恒常性を動的に保つという意味で、ホメオカオスと呼ぶことにする。ホメオカオスによる高い変異率の状態をビット空間の中でのダイナミクスとして解析しなおすとビット空間のクラスタリングという現象がみえてくる。このクラスタリングについて§5で詳しく解説し、ビットクラスターの時間的な分岐現象などについて報告する。§6ではビットのクラスタリングとスキーマの進化について議論し、並列分散処理の方向を考察する。

2 クラシファイヤー・システム

自律的な分散処理を行なうシステムとして、クラシファイヤー・システムという「ルール」群の生態系がホランドによって提唱されたのは20年近く昔になる[5]。システムが全体としてなすべき処理はルールという形で分散され、個々の「ルール」は自分の「条件」にあった時のみ、自分のルールを実行しようとする。また各ルールは「財」を所有する。ルール間には自己の財を増やそうとする経済原理が存在し、それにしたがってルールはその「財」を増やしたり減らしたりす

る。いくつかのルールが同時にルールを実行使用としたときには、財の多いルールほど通しやすくなる。また他のルールに投資して結果的に儲かると投資した人が順次収益を貰える。この仕組みがバケットブリゲード方式である。実際の経済市場では商品がヒットするとその小売店、卸売業者、生産者というネットワークが連鎖的に儲かる仕組みになっているが、同じことがクラシファイヤーにも出現する。新しいルールはクロスオーバーなどの遺伝的アルゴリズムによって生態系に参入し、儲からないルールは破産し生態系に関与しなくなっていく。こうしてルールの生態系は時間と共に更新されていく。ルール間の相互作用(どのルールとどのルールが影響しあうか)は一般に複雑で、生物群の生態系のように宿主と寄生者の関係(ルールAが儲かるとルールBも儲かる)が生成しうる。クラシファイヤー・システムの数理モデルとしての特徴のひとつは、ルールを1次元のビット列として表現し、複雑なルール間の相互作用をビット間マッチングとして表現したところにあると思う。このためクラシファイヤー・システムは、例えば免疫ネットワークのひとつの数理表現となりうるということが報告されている [6]。

このルール生態系においてどんなスキーマ(schemata)が進化してくるかが、その生態系を特徴づけることになる。スキーマとは、いくつかのルールを粗視化したメタ・ルールで、ルールの行動を生成する条件の特殊性を変化させることができる。このスキーマが進化適応系としてのクラシファイヤー・システムに柔軟性を持たせているのである。

例えば個々のルールが長さ5のビット列で表されるとすれば、各ルールは2の5乗超立方体の頂点として表現される。この時ルール群 [10001], [10010], [10000], [10011] をひとまとまりとして表現するルールは [100##] と表せる。ここで#は1でも0でもいいという意味の記号である。#の記号が多いほどそのルールは汎用性が高く、#が少ないほどそのルールの特殊性は強い。ここでルールを遺伝子にこじつければ、きちんと発現しなくてはいけない遺伝子コードはその特殊性を増し、どうでもいいコードは多様化するといえることができるかもしれない。本稿では遺伝子を1次元のビット列という点からクラシファイヤーのルールと関係づけ、宿主・寄生のダイナミクスのもとでのスキーマの自動生成・崩壊過程に言及する。

3 ホスト・パラサイトのモデリング

生物の個体変動におけるカオスは、カオス研究の初期に発見されたカオスのひとつであるが、このカオスは生物を絶滅の危機に導くありがたくないものと考えられていた [7]。しかしここでは逆にカオスこそが種の多様性の維持に必要なものである、という考察をしたい。

計算機上のモデルではホストとパラサイトの遺伝子型は有限長のビットパターンでコード化され、同じビットパターンを持ったホストとパラサイトのみが相互作用すると考える。定義より相互作用によってホストは損をし、パラサイトは得をする。

例えば遺伝子型 [0011101] を持つホストの時間ステップ n における個体数を $h_{29}(n)$ と表すとする。ここで29は遺伝子型 [0011101] の10進表示である。この個体数が時間とともにロジステック型の時間発展をするとすれば、その式は、

$$h_{29}(n+1) = a_{29}h_{29}(n)(1 - h_{29}(n)) \quad (1)$$

となる。ここで a_{29} はこのホストの増殖率である。この時このホストと同じ遺伝子型を持つパラサイトが $p_{29}(n)$ 個寄生したとすると、ホストの増殖率は例えば

$$a_{29} = a_{29} \exp(-\beta p_{29}(n)) \quad (2)$$

に滅じられてしまう [8]。パラサイトがいればいる程ホストは増殖できない。この関係は

$$h_{29}(n+1) = a_{29} \exp(-\beta p_{29}(n)) h_{29}(n) (1 - h_{29}(n)) \quad (3)$$

$$p_{29}(n+1) = a_{29} h_{29}(n) (1 - \exp(-\beta p_{29}(n))) \quad (4)$$

によって表せられる。増殖率 a_{29} は一般には遺伝子型ごとに異なる値を取る。パラサイトがいなければ増殖率ももっとも大きい種が一番栄える種ということになる。パラサイトがいる場合ホスト種は点突然変異 (1 ビット離れた遺伝子型へと変異する) によってパラサイトから逃れることができる。いっぱい寄生されている時には大きな突然変異の値を持つことが有利であるが、逆の時にはへたに変異しないほうが得になる。ホスト、パラサイトともに 1 ビット近傍へと突然変異する割合は次のように量子化して扱う。

$$\mu(i) = 2^{i-L} \quad (5)$$

ここで $(i=1,2,\dots,L)$ であり、 L は量子化された変異率の数である。ところで突然変異率 μ を決めている部分も自分自身の変異にさらされるため変異率 μ 値は、自分自身の μ の割合で他の μ 値に遷移していく、という自己言及的なルールを考える。簡単のためここでは変異率 $\mu(i)$ は遷移確率 $\mu(i)$ で、 $\mu(i+1)$ あるいは $\mu(i-1)$ に遷移するとしよう。すなわち一回の遷移で、校正率が倍精度になったり半分の精度になったりするわけである。この変異率の変異ルールの導入により、同じ遺伝子型であっても異なる突然変異率を持つことができる。それぞれの種は遺伝子型と変異レベルの両方で記述される。自己増殖した個体 $h_{i,j}$ のうち $\mu h_{i,j}$ が変異し、変異種のうち半分は変異レベル j のみを変化させ、半分は遺伝子パターン i のみを変化させる。これらは超立方体上の拡散過程とみなされ、以下の式で記述される。

$$h_{i,j}(n+1) = (1 - \mu_j) F_{i,j}^n(h, p) + \sum_{j'} \mu_{j'} F_{i,j'}^n / 4 + \mu_j \sum_{i'} F_{i',j}^n / 2k \quad (6)$$

$$p_{i,j}(n+1) = (1 - \mu_j) G_{i,j}^n(h, p) + \sum_{j'} \mu_{j'} G_{i,j'}^n / 4 + \mu_j \sum_{i'} G_{i',j}^n / 2k \quad (7)$$

ここで $F_{i,j}^n = a_i \exp(\beta \sum_j p_{i,j}(n)) h_{i,j}(n) (1 - \sum_j h_{i,j})$ で、 $G_{i,j}^n = h_{i,j}(n) (1 - \exp(\beta \sum_j p_{i,j}(n)))$ である。上の式で k はビットの長さである。シミュレーションでは 7 を考える。また変異の量子化数 L は 13 とする。

4 ホメオカオス

4.1 平均の突然変異率の落ち着き先

まず初めにパラサイトと相互作用がないホストの突然変異率の変化を考察しよう。この時もし各ホストの増殖率が遺伝子型によらないのであれば、突然変異はまったく有用ではないので、だんだんと下がってゼロに漸近していく。「変異率が自分の変異率で変異していく。」という自己言及的なダイナミクスの唯一の安定点が『変異率 = 0』という固定点だからである。次に増殖率が遺伝子型の関数だとしてみよう。このときには、いい遺伝子型と悪い遺伝子型の違いがあるので、変異率は単調にゼロに向かうわけではない。増殖率が遺伝子型の複雑な関数になっていけば、もっとも増殖率の大きな良い遺伝子型を探す問題は、りっぱな最適化問題である。最適化問題の難しさは、その評価関数が無数の準安定状態を複雑に持っていることに起因される。最適化問題を探索

する手法のひとつシミュレーテッド・アニーリング法(SA法)では、はじめに解を粗くサーチして次第に的を絞っていく、といったことをする。粗くサーチするとは高い温度に設定してサーチすることであり、次第に的を絞っていくとは、温度を段々に下げていくことに対応する。温度を突然変異率に置き換えれば、パラサイトのいないホストの進化過程は『与えられた評価関数の自動化されたSA法』として捉えることができる。より環境に適した種ほど子孫を多く残せるというのが、この進化ダイナミクスの主導原理である。局所最適解に入りこむとほとんどの変異種は損をし、あまり変異しない種が局所最適解にとどまって多くの子孫を残せる。このため集団の平均変異率は降下するのである。しかし運よくその局所解から脱出してよりよい解(高い増殖率)に変異種が到達した場合、脱出できた、すなわち突然変異率の(e.g. 温度の)高い種が今度はより多くの子孫を残して平均の変異率を上昇させる。このように変異率を上げ下げしながら最適解を探し、完全な局所最適解か、真の最適解に到達したところで突然変異率は(e.g. 温度は)ゼロに向かって下がり始めるのである。

4.2 パラサイトとの相互作用のある系

次に純粋にパラサイトによる効果をみるために増殖率は遺伝子型に依存せず一定とする。パラサイトとの相互作用を考慮すると、初期に低く設定した平均の突然変異率(以下すべて平均の値について論じる)は時間とともに高い突然変異率へと推移する。そして高い値に推移した突然変異率は、その時の相互作用の強さ(β)に応じた高さのまま安定に保たれることが確かめられる。最終的な突然変異率の値が、一定の値で保持される場合は、最終的にはゼロにいく場合と比べて力学系の状態として違いがあるようである。

系の安定性は相互作用 β の関数として変化する(図1)。ちょうど β の値がある点を越えて系の不安定性の指標である(最大)リアプノフ指数が0以上となる領域で最終的な突然変異率の値が高いまま保たれることが分かる。個体数数のカオティックな変動と、高い突然変異率の転移・維持とはどういう関係にあるのだろうか。定常解が安定化される場合には、増殖率が一定なので変異する優位さはなくなり、突然変異率はゼロになってしまう。系は突然変異を介して結合しているので変異が十分小さくなると本質的に1対のホストとパラサイトの2次元写像に還元されてしまう。一方定常解が不安定になる場合には、各種ごとの個体変動はほとんど同期しないので、より優位な種(すなわちパラサイトがあまり寄生していない種)というのが存在することとなる。したがって突然変異によって優位な種に移りえることになるので、突然変異率が残ってる方が進化的に優位となる。

高い突然変異率を維持する時に出現するカオスは、1対のホスト・パラサイトの発展を表わす2次元カオスの単なる直積ではない。変異率上昇後は本質的に部分系に切り分けられない大自由度のカオスとなっているのである。さらに特徴的なことは2次元カオスに比べ、同じパラメータ領域で不安定性がかえって減少していることである。この大自由度カオス状態では突然変異によりお互いに移り合える種がネットワークを構成し、ひとつの種の変化が全体に影響を及ぼすことを示唆している。つまりホストは互いに突然変異によって移りあうネットワークをつくることでパラサイトによって絶滅させられるのを防ぐように見える。ネットワークの結合はパラサイトの存在によって保たれているといえる。このネットワークをつくることでもとのカオスにひそむ不安定性をみんなで分け合って全体として弱めている。この特徴的な大自由度しかし弱められたカオスを、種の多様性を安定に保つという意味でホメオ・カオスと呼んでいる。

5 スキーマの進化

カオティックに時間発展する個体群の振る舞いを、ビット空間でながめるとどうなっているのだろうか。この様子は、個体群数の時間に関するスナップショットをとってみればいい。この時明らかにいくつかのクラスターに分れている様が観測される。図2に、横軸に[000000]から[111111]までを小さい順に並べ、縦軸にそのビット・パターンをもつ個体数をプロットしてある。これから分かるように、ホストは固定化されたいくつかのビットを共有してクラスター化しているように見える。この図2では、大きく4つに分れてクラスター化している。4つのクラスターは、それぞれ[#0##0##], [#0##1##], [#1##0##], [#1##1##]というスキーマとして表現される。ここで#はクラシファイヤー・システムにおける”don't care symbol”に相当し、1でも0でもよい。したがって各クラスターに属する異なる種の数 $は2の#の数のべき乗となる$ 。この場合には各クラスターに属する数は32である。このようなクラスターが生成されるのは、不安定性がそれほど強くない、しかしカオティックな領域にある場合で、不安定性が一定以上強くなると明確なクラスターをつくりにくくなる。またクラスターをつくっている領域でもそのクラスターは時間的にクラスターを作る種の組み合わせや数を変えていくことがわかる。例えばクラスターの時間的な分岐という現象を観測することができる。ある時間領域で4クラスターを形成していた系、[#0####0], [#0####1], [#1####0], [#1####1]が6番目のビットを揺るがせて[#0####0] → [#0####00] + [#0####10]という分岐が起きる。その結果全体が[#0####00], [#0####10], [#0####01], [#0####11], [#1####00], [#1####10], [#1####01], [#1####11]という8クラスターに分岐する。このようなクラスター化はホストだけでなくパラサイトでも等しく生じている。実はこのクラスター化と突然変異の上昇/下降には関係があることがわかるので、そのことを直感的に説明してみよう。

突然変異率がカオティックなダイナミクスのもとで高い値に維持されるのはパラサイトによりホストの適応度環境が、その凸凹度を時間的に変化させ、絶えず変異すると得なところを持つからであった。クラスター化はこれとは矛盾する効果である。最初個々のダイナミクスの持つ不安定性のためにバラバラに振動していた系は、いい環境と悪い環境の差があるために突然変異率を上昇させる。この結果クラスターが形成され、不安定性が押さえられる。特に完全にひとつのクラスターにそろってしまると、ホストにとってもパラサイトにとっても適応度は遺伝子型によらなくなり、高い突然変異率の優位さは失われて、変異率は下がり始める。こうなると異なる遺伝子型間の結合が切れて、もとのダイナミクスのもつ不安定性が顔を出してくる。再びクラスターは壊れて、突然変異率を再び上昇させるということになる。このシナリオにしたがって突然変異率は時間的に変動するのである。

6 ディスカッション

ここであらためてクラシファイヤー・システムとホスト・パラサイト系を比較してみよう。バケット・ブリゲード法は投資したルールがいい結果をあげると、投資したルールに収益がまわってくるという仕組みであった。ホスト・パラサイト系では高い変異率をもってパターンAからパターンBに変異した場合、BがAより有利であればその変異した個体は数を増すので、結果として平均の変異率を上昇させる。クラシファイヤーではあらかじめ#記号を持っていて、交叉によって1遺伝子上の数をコントロールし、スキーマを構成する。ホスト・パラサイト系では同じ個体数をもつ遺伝子群をひとまとりとみること、スキーマが自動的に定義される。

ホスト種を外部のデータと考え、パラサイト種を内部処理系として考えると外部のデータ構造

の持つダイナミクスに応じてスキーマが自動生成されてくると考えることができる。データが安定なダイナミクスを持っていれば生じるクラスターは時間的に安定なものとなる。一方データが不安定なダイナミクスをもつようなものであれば、時間的に不安定なクラスター化が生ずるだろう。ホスト・パラサイト系はデータの固有のダイナミクスに応じてスキーマという系の内部表現を作り上げるのである。変異率の大きさは生成されたスキーマ間の相関の強さを示すものとなる。

データ群のダイナミクスを見ているという点で、スタティックに与えられたデータ群のカテゴリライズの問題とは対照的である。データ群のダイナミクスがある種の癖(固有のダイナミクス)を持っているときには、それがスキーマに反映されると期待される。

さらにわれわれのホスト・パラサイト系はより直接的な免疫系のメタファーとなっていることがわかる。生体防御機構を担う免疫系ではまだ接触していない未知の抗原に対応するためにも多様な抗体を常に保持している必要がある。事実人間の免疫系における突然変異率は体細胞レベルの1000倍以上に保たれていることが知られている。免疫の抗体産出をコントロールしているのは抗体同志がホストとパラサイトのように相互作用しあうイデオタイプ・ネットワークによるといわれている[9]。この抗体の多様性を動的に保つ機構としてホメオカオスが働いているかもしれないと我々は考えている。ネットワークに生ずるイデオタイプのクラスターがあれば、それは抗原基パターンによって語られる外部のデータ構造の内部表現とみなすことができるだろう。免疫ネットワークによる外部抗原の表現は、従来のようにひとつのイデオタイプに求めるのではなく、動的なクラスターに求められるかもしれない。

参考文献

- [1] J.Arthur and F.Quinn, Bull. Ame.Math.Soc. 29(1993) 1.
- [2] T.Ikegami and K.Kaneko, Physica D 40 (1990) 235.
- [3] K.Kaneko and T.Ikegami, Physica D 56 (1992) 406.
- [4] T.Ikegami and K.Kaneko, CHAOS 2 (1992) 397.
- [5] J.H.Holland, "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems." In *Machine Learning, An Artificial Intelligence Approach* edited by R.S.Michalskii, J.G.Carbonell and T.M.Mitchell, Vol.II , pp .593-623. Morgan Kaufmann, 1986.
- [6] J.D.Farmer, N.H.Packard and A.S.Perelson, Physica 22D (1986) 187-204.
- [7] A.Berryman and J.A.Milstein, Trends Ecol.Evol. 4, 26(1989).
- [8] M.P.Hassell, H.N.Comins and R.M.May, Nature 353(1991)255.
- [9] N.K.Jerne, Sci.Ame. 229(1973) 52.

図の説明}

図1 増殖率 $a=3.89$ 時のホストとパラサイトの相互作用の強さ β とする。ちょうど $\beta=5$ のあたりで系の状態が大きく変化している。1) β とリアブノフ指数の関係。白点は1対のホストとパラサイト(本文中式3と4)の指数。黒点7ビット変異レベル13の結合系の指数(本文中式6と7)。2) β と平均の突然変異率。白点はホスト、黒点はパラサイト。

図2 ホスト個体数の時間スナップ・ショット。横軸はビット・パターンで左側から [0000000],[0000001],[0000010],…………,[1111111]。縦軸は各ビット・パターンの個体数。 $a=3.4$, $\beta=6.9$ 。この例では4クラスター [#0##0##], [#0##1##], [#1##0##], [#1##1##] に分割している。

図3 図2と同じ表示法。 $t=2120$ で4クラスター [#0####0], [#0####1], [#1####0], [#1####1] に分れていた状態が $t=2470$ で6ビット目を揺るがせて8クラスターに分割する。8クラスターは [#0####00], [#0####10], [#0####01], [#0####11], [#1####00], [#1####10], [#1####01], [#1####11] この時パラメーターは $a=3.4$, $\beta=7.0$ 。

図4 クラスターと突然変異率と積算された最大リアブノフ指数の関係。パラメーターは $a=3.4$, $\beta=7.0$ で、4クラスターが1クラスターになり、8クラスター化する。1) クラスター [#0####10] と [#0####00] の個体数の差を時間の関数としてプロット。 $t=2400$ でクラスターが分岐する。2) パラサイト(濃線)とホスト(淡線)の平均の変異率。 $t=2300$ あたりで、4クラスターが1クラスターになり、変異率が下がり始める。3) 積算された最大リアブノフ指数をみると、1クラスターになって変異率が下がり始め、一定以上下がると不安定性が一挙に高まることが分かる。 $t=2900$ を過ぎて8クラスターはもとの4クラスターに戻っていく。

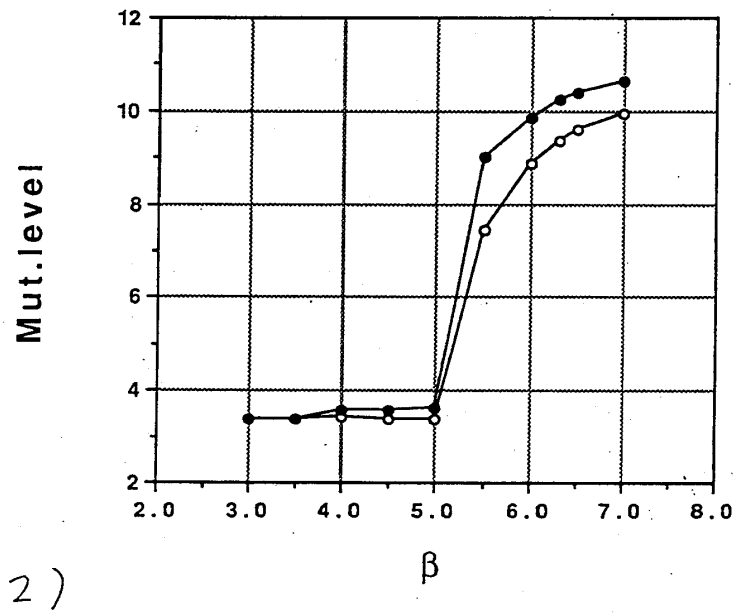
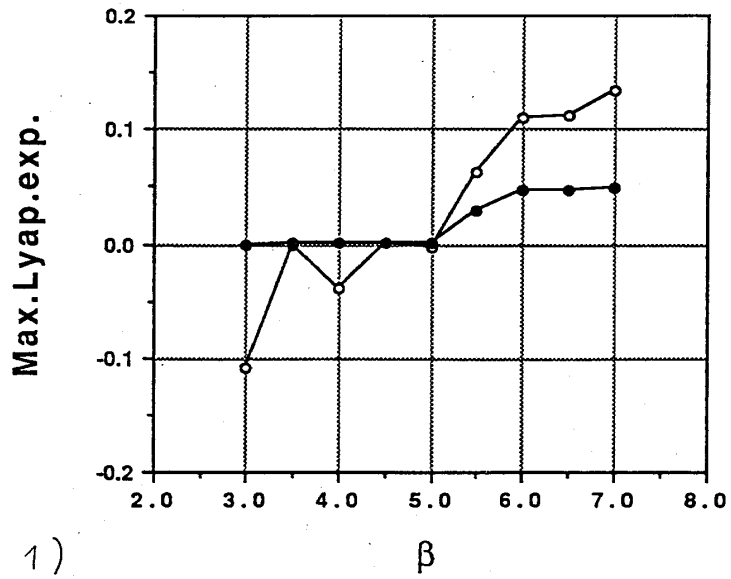


図 1

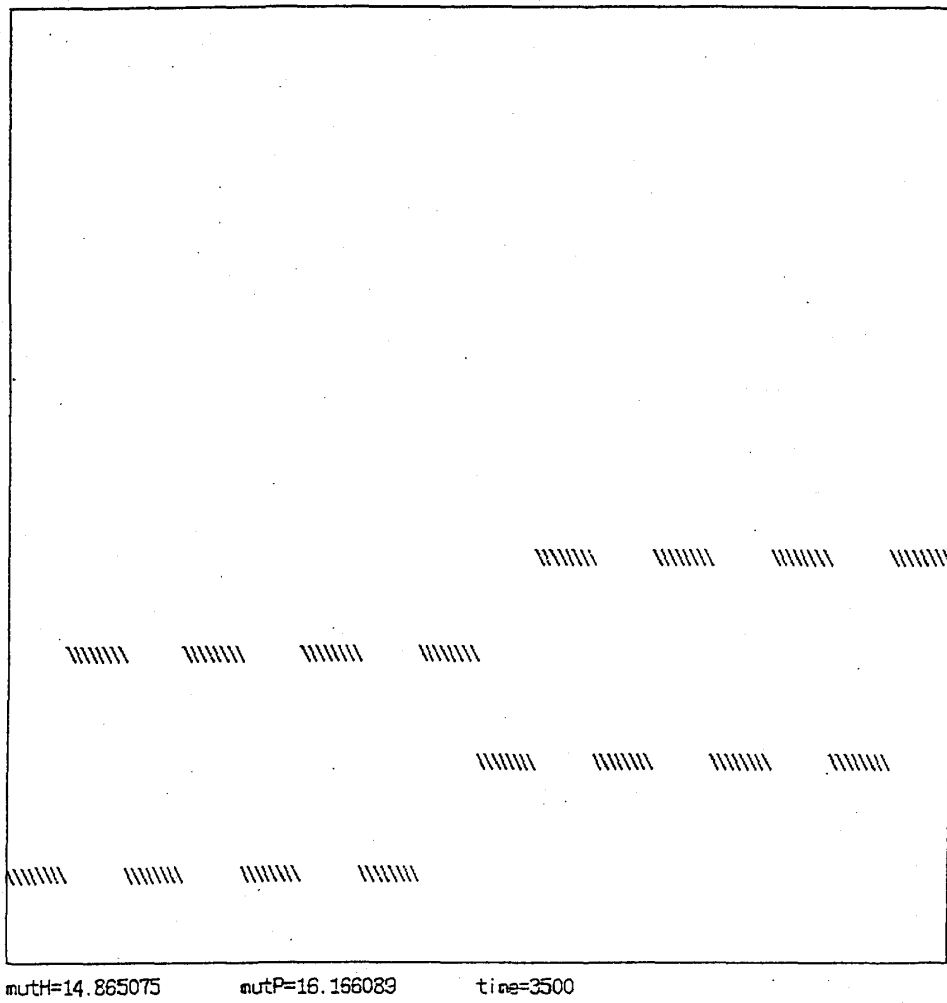
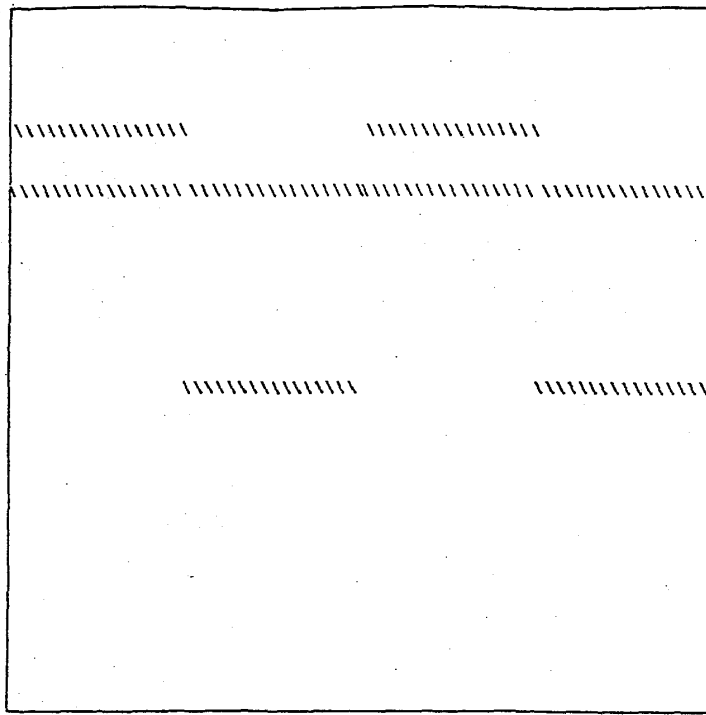
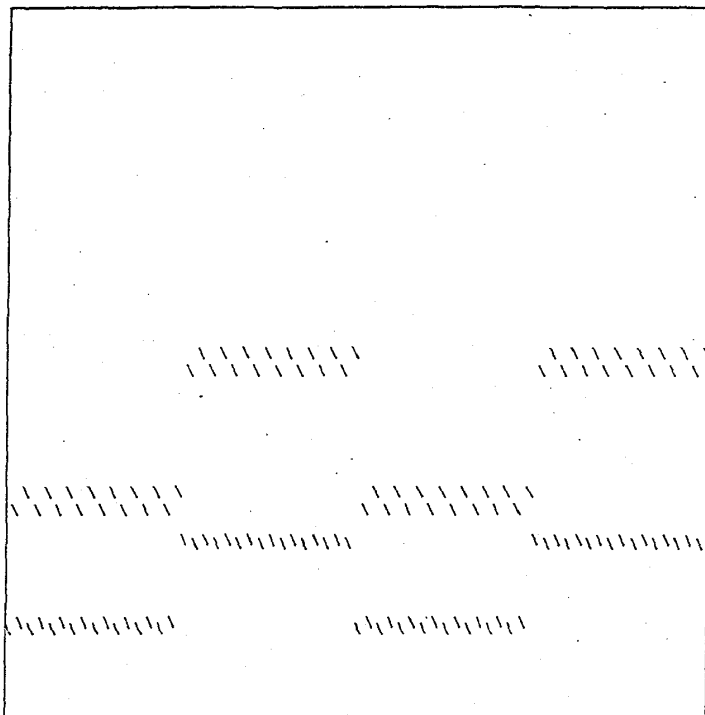


図 2

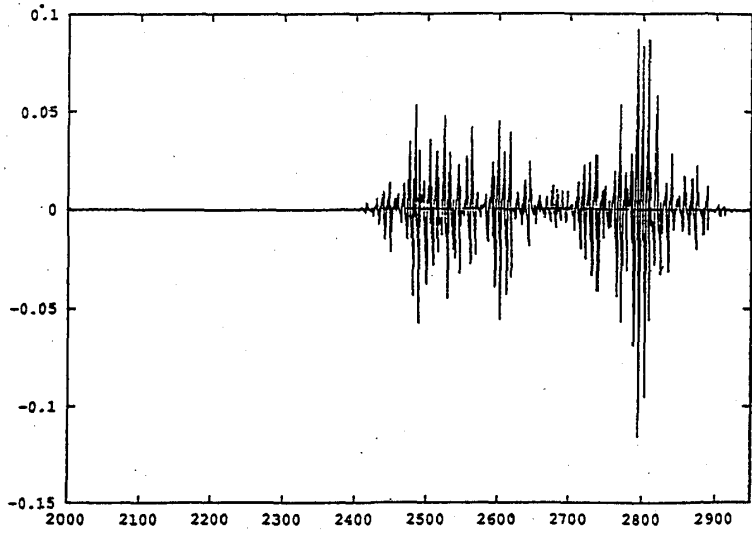


mutH=14.870233 mutP=16.084855 time=2120

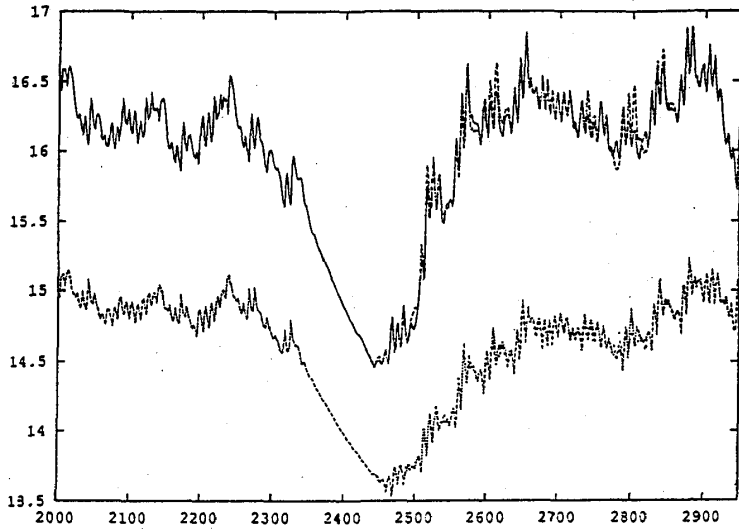


mutH=13.675313 mutP=14.618050 time=2470

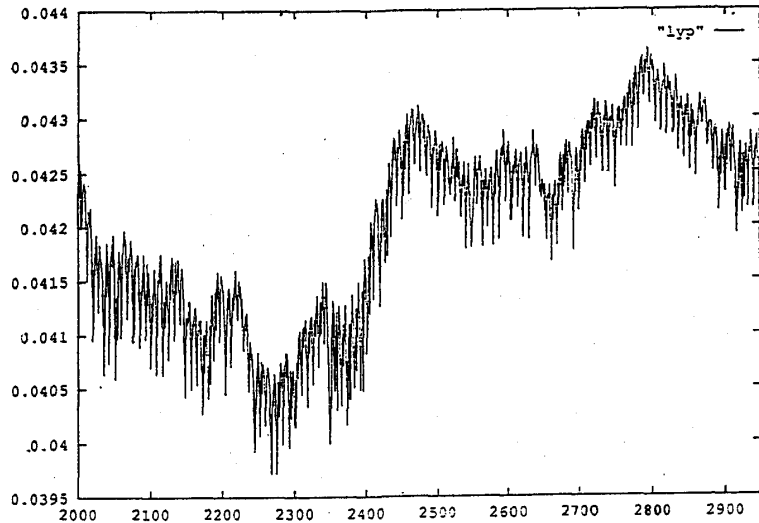
図 3



1)



2)



3)

図4