



研究ノート

酵素はなぜ良くできているのか

ト 部 格*

Why are enzymes well-designed?

Key Words: enzyme, evolution, sequence space, structure-function relationship

1. はじめに

生物を相手に研究していると、つくづく生物は良くできていると思う。生物は生命のない分子から構成されているが、個々の構成分子もまた良くできている。中でも酵素には、いつも感心させられる。このような酵素や生物を理解することは、結局、「なぜ良くできているのか」の質問に答えることだと思って研究を進めている。

2. 酵素のどこが良くできているか

酵素は、生体内で進行しているさまざまな化学反応の触媒である。その触媒機能に関しては、多種多様の酵素で古くから研究されており、その優秀さ、とくに、常温で高い活性を示すことと、特異性とよばれる反応選択性に優れていることはよく知られている。

酵素の本体は、20種類のアミノ酸が数百個、一本の紐のようにつながったタンパク質である。この紐をバラバラにしたアミノ酸混合物には、酵素の高度な機能は存在しない。つまり、アミノ酸という大した機能もない簡単な材料を単に一列に並べただけで、生物に必要な化学反応全てにわたる触媒ができているというのは驚くべ

きことである。

酵素の種類が違えばアミノ酸の並び方が異なっている。そして、そのアミノ酸の並び方は、遺伝子であるDNAの塩基の並び方で記録されている。生体内では、その遺伝子に書かれた設計図をもとに、自動的に酵素が作られてくる。このオートメーション工場のユニークなところは、設計図に従ってできた酵素が自分達の設計図をコピーし、そのコピーから自分達を作り出すという自己言及的なシステムになっている点である。このシステムは、一度動き出すとエネルギーのある限り自己再生し続けることができる。しかし、ニワトリとタマゴの問題と同様、酵素と設計図のどちらが先にできたかは永遠の謎である。そして、この良くできたシステム、良くできている酵素の一番の良さは、こんな素晴らしいものが誰がどうしようと思ったわけでもなく、勝手にできてしまっていることであろう。

3. なぜアミノ酸をつないだだけで このような機能が生まれるのか

これは古くからの酵素に対する質問である。そして、膨大な研究の蓄積があるにもかかわらず、今でも酵素に関する研究の中心テーマとなっている。このような難しい問題に研究者が答えられるかもしれないと思うようになったのは、酵素の立体構造をX線結晶解析により見ることができるようにになってからで、形がわかれればその機能が生まれるからくりが発見できそうに思えるからである。そしてそのからくりがわかれれば、酵素の設計原理を手に入れたことになり、望みの機能をもつ新しい酵素をわれわれの手で

*Itaru URABE
1945年2月28日生
1969年大阪大学大学院工学研究科
醸酵工学専攻修士課程終了
現在、大阪大学大学院工学研究科
応用生物工学専攻、教授、工学博士、醸酵工学
TEL 06-879-7427
FAX 06-879-7428



作ることができそうである。

ここで質問が2つに分かれる。1つは、アミノ酸がある順序でつながった一次配列をもつ紐がどのような立体的な形をとるのかという、高次構造構築原理に関するもので、物理の問題である。もう1つは、ある触媒機能をもつにはどのような高次構造をとらねばならないかという、機能発現原理に関するもので、これは化学の問題である。もちろん、各質問には逆方向の問い合わせもある。これを図示すると、図1の逆三角形の下2辺における相関関係を明らかにしようとするものである。

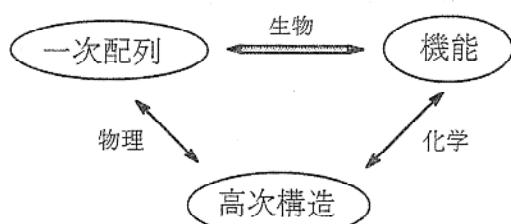


図1 酶素の設計原理

1本の紐がどのような形をとるかは、原理的には力学の基本法則に従って答えが出てくるものである。つまり、酵素分子を構成する各原子について力学的方程式を立て、最終的に各原子がどの位置に収束するかを解けばよい。方程式を立てるのは難しくはないが、やっかいなのは数が多いことである。酵素分子は数百のアミノ酸からできており、水溶液中では水分子も考慮しなければならない。コンピューターの進歩により計算速度は飛躍的に速くなっているが、まともに計算するのは不可能である。そこで種々の近似をして答えらしきものを見せていているわけであるが、複雑系の常として、そのような近似が思いもよらぬ大きな変化を引き起こすことがあるので、あまりあてにできない。この事情は、気象予測の問題と同様である。

問題をもっと抽象化し、1本の紐がなぜ1つの形に収束しうるのかといった研究も盛んである。この場合はモデル化できるので、構築原理の理解は進んでいるが、それだけでは、ある1本の紐が実際どのような形になるのかという、個々の問題に答えることはできないのである。

酵素の触媒機能をその高次構造から定量的に

予測することは、さらに難しい問題である。これを解くためには分子軌道理論を用いるのであろうが、酵素反応について計算することは不可能である。このことは、人の顔や形を見ただけではその人の食物の好みや走る速さが予測できないのと同様である。もちろん、酵素の触媒機能をその高次構造を用いて説明することはよく行われているが、その説明は定性的かつ後付け的なものにすぎない。

アミノ酸がある順序でつないだら、このような形をとりこのような機能をもつということを、物理と化学の言葉で説明できれば、なるほどそういう訳で酵素は良くできているのかと納得できるのだが、どうもそれは無理らしい。この問題は、研究者にとっての永遠の良きテーマとなってしまっている。

4. なぜ酵素は自然にできたのか

生物は設計図に従って酵素を作り、その酵素の働きで生きている。生物は酵素の設計図をどうやって手に入れたのだろう。生物が、高次構造構築原理と機能発現原理をもとに自分で設計したとは考えられないし、神様が作ったとも思いたくない。生物は生物特有の法則を使って酵素を作り上げたのだろう。

生物が使うのは進化の法則である。生物は自己再生システムであるが、設計図をコピーするときにときどき間違いを起こす。少し違った設計図からは少し違った酵素や生物ができるため、生物は多様化する。多様な生物が自己再生を続けてゆくと、集団内で競争や協調が生じ、うまく生き残れるものと脱落して消滅するものに分れる。どんなものが生き残るかは複雑な相互作用の結果なので、簡単な方程式では表現できないが、偶然起こる間違いとあるふるい分けの法則により、生物は多様性を保ちつつ進化してきたのである。

この生物進化のプロセスは酵素の進化にもあてはまる。ここでふるい分けの法則を単純化し、適応度の高いものが生き残るとしよう。適応度は酵素の機能によって決まる。一方、設計図はアミノ酸の並び方、即ち一次配列に相当する。すると、酵素は一次配列をランダムに変化しな

がらより適応度の高い方へ進化してきたことになる。つまり生物は、図1の上辺に示すように、タンパク質の一次配列と機能との相関関係をもとに、進化の法則を使って酵素を作り上げてきたのだ。

それではその相関関係はどうなっているのだろう。20種類のアミノ酸がn個並んだタンパク質の種類は 20^n あり、それぞれの配列はn次元配列空間の1点として表わされる。各配列がもつ機能、即ち適応度をn+1次元目の軸方向にプロットすると、一次配列と適応度との相関関係を示す超曲面が現われる。これが適応度の地形である。図2は、配列空間を2次元に圧縮して示した適応度地形の概念図である。酵素進化のプロセスは、この図中の太線のように示すことができる。もちろん生物がこの地形を知っているわけではないが、適応度の高い方へと登ってゆくことにより、自然に今のような酵素ができたのである。

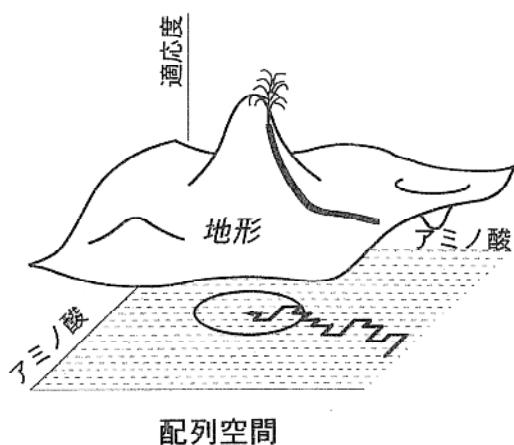


図2 タンパク質配列空間における適応度の地形

さて、自然に高い方へ登ってゆけるためには、地形にいくつかの制約が必要である。まず、山にはある広さの領域がなければならない。平原の中の一本杉のような山であれば、平原を歩き回ってそこに辿り着く確率はほとんどない。また、山の領域が空間全体のほんの一部だとしても、その裾野は広がって全体に勾配を与えていなければならない。進化の初期にランダムにできたタンパク質でもわずかながら機能をもっており、その機能が一次配列の変化に応じて変わるものでなければ、進化の法則も使いようがない。そこで、実際の地形がどうなっているかを調べてみた^{1,2)}。この研究は現在進行中であるが、上記の条件は満足されているようである。結局、酵素が良くできているのは、タンパク質の配列空間の地形が、生物が進化の法則を使うのに都合良くできているからであるということになる。配列空間の地形を理解することは、酵素の新しい面を理解することであり、また、進化の法則を使って新しい酵素を創出するという進化工学の可能性を確かなものにすることもある。

文 献

- 1) Trakulnaleamsai, S., Yomo, T., Yoshikawa, M., Aihara, S., & Urake, I. (1995) *J. Ferment. Bioeng.*, 79, 107-118.
- 2) Prijambada, I. D., Yomo, T., Tanaka, F., Kawama, T., Yamamoto, K., Hasegawa, A., Shima, Y., Negoro, S., & Urake, I. (1996) *FEBS Lett.* 382, 21-25.

