NAIST-IS-DD1461204

博士論文

人獣共通感染症 RNA ウイルスゲノム配列の方向性のある変化および再発性の研究

和田　佳子

２０１８年３月２３日

奈良先端科学技術大学院大学

情報科学研究科

本論文は奈良先端科学技術大学院大学に
博士授与の要件として提出した博士（理学）論文である

和田　佳子

審査委員：

金谷　重彦　教授	（主指導教員）
佐藤　嘉伸　教授	（副指導教員）
Md.Altaf-Ul-Amin 准教授	（副指導教員）
小野　直亮　准教授	（副指導教員）

人獣共通感染症 RNA ウイルスゲノム配列の方向性のある変化および再発性の研究

和田 佳子

内容梗概

エボラウイルス、MERS コロナウイルス、インフルエンザウイルスは、急速に突然変異する人獣共通感染症 RNA ウイルスである。ウイルスは増殖の際、多くの宿主因子に依存するが、ヒト細胞は非ヒト宿主から侵入するウイルスにとって理想的な増殖環境であるとは限らない。ウイルスがヒト以外の宿主からヒト細胞内に侵入すると、ヒトの免疫系により全滅させられることもあるが、ヒトの免疫系から逃れつつ、ヒトの細胞内のリソースを効率よく利用することで大増殖することがある。このようにウイルスは突然変異を繰り返しながら、ヒトの細胞内の環境に急速に適応していく。ウイルスの感染が沈静化してから、十分な時間が経過した後に、再度このようなヒト以外の宿主からの感染が再発した場合に、もし、ウイルスがヒトの細胞に適応するためにゲノムを改編していったプロセスに一定のパターンが見いだせるならば、この変化パターンをうまく活用することで、効果の高いワクチンの開発など、有効な感染予防策を打ち立てられる可能性が見えてくる。

そこでウイルスのゲノム中の、様々な長さのオリゴヌクレオチド組成の時系列解析を行うことで、ヒト以外の宿主からヒト細胞内にウイルスが侵入した後のウイルスのゲノム配列のオリゴヌクレオチド組成におけるパターンの時間変化を調べた。

この情報論的な感染予防戦略が正しいことを裏付けるように、最近の西アフリカのエボラウイルスの大流行の際に、極めてはっきりとした方向性を持ったオリゴヌクレオチド組成の時間変化が、ギニア、リベリア、シエラレオネの 3 つの別々の地域で共通して観察された。さらに、中東から始まった最近の MERS コロナウイルスの流行においても、オリゴヌクレオチド組成における方向性のある時系列変化が観察され、明確な組成変化が特定のウイルスだけに見られる特殊な傾向ではないことが明らかとなった。ヒト A 型インフルエンザウイルスについても、数十年の間隔でヒト以外の宿主からヒト集団へと侵入した 3 つの異なる亜型に関して、いずれもオリゴヌクレオチド組成に方向性と再現性のある時系列変化が観察された。この 3 つの亜型に共通して認められた明らかな方向性のある変化を示す 20 ヌクレオチド程度の長さのオリゴヌクレオチド類は、ヒト A 型インフルエンザウイルスの siRNA ターゲット配列のいくつかに対応しており、その siRNA の活性は実験的にも証明されている。これらのことから、本研究で示したオリゴヌクレオチド組成の方向性のある時系列変化の予測技術は、診断 RT-PCR プライマーの開発や、長い期間に渡って有効性を保持する治療用オリゴヌクレオチド（核酸医薬）のドラッグ・デザインにとっての必須の基盤技術となる。

キーワード

バイオインフォマティクス、ゲノム解析、データサイエンス、ＲＮＡウイルス

*奈良先端科学技術大学院大学情報科学研究科　情報科学専攻　博士学位論文,

NAIST-IS-DD1461204　2018 年 03 月 23 日.

Directional and reoccurring sequence change

in zoonotic RNA virus genomes visualized by timeseries word count

Yoshiko Wada

Abstract

Ebola virus, *MERS coronavirus* and *influenza virus* are zoonotic RNA viruses, which mutate very rapidly. While viruses depend on many host factors during proliferation, human cells are not always the ideal growth environment for viruses invading from nonhuman hosts. When a virus invades into a human cell from a nonhuman host, viruses tries to make full use of the resources of human cells for efficient proliferation. For this reason the virus adapts to the environment while mutating, but in some cases it may be annihilated by the human immune system. In other cases viruses sometimes find ways to coexist in humans by weakening the toxicity to humans. If patterns can be found in the process that the virus modified the genome to adapt to human cells when such virus infection occurred again over time, by utilizing this pattern change, it will be possible for us to establish effective measures against virus infection and develope some highly effective vaccines. So we performed a time series analysis of short oligonucleotide composition and long oligonucleotide composition in the viral genome, and found specific patterns in the oligonucleotide composition of the genomic sequence of the virus after invading into the human cell from a nonhuman host. Directional time series changes in oligonucleotide composition were commonly observed in the three regions of Guinea, Liberia, and Sierra Leone at the time of the recent outbreak of *Ebola virus* in West Africa. Directional time series changes in oligonucleotide composition were also observed in the recent *MERS coronavirus* epidemic that began in the Middle East. Regarding *human influenza A virus*, a common directional time series change was observed in oligonucleotide composition with respect to all three subtypes. Long oligonucleotides showing obvious directional changes observed commonly in these three subtypes correspond to some of the siRNA target sequences of *human influenza A virus*, and experiments on the activity of these sequences have already been reported. By predicting directional time-series changes and recurrence of oligonucleotide composition, it leads to the development of diagnostic RT-PCR primers, and our analysis is an essential technology element for nucleic acid pharmaceutical design to develop therapeutic oligonucleotides with long effectiveness.

Keyword

Bioinformatics, genome analysis, data science, RNA virus

Doctoral Dissertation, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1461204　March, 23, 2018.

目次

図目録

表目録

第１章　研究の背景

最近の西アフリカでのエボラウイルスの大流行[1-4]や新興および再興するインフルエンザウイルスの流行[5]など、ウイルスは公衆衛生に重大な脅威をもたらしている。このような人獣共通 RNA ウイルスは急速に突然変異するため、突然変異に対応して、長い有効性を有する診断用オリゴヌクレオチドや治療用オリゴヌクレオチドを設計するためには、突然変異によるウイルスの配列の時系列変化を、明らかにする必要がある。

そのためには、データ駆動型のビッグデータ解析が必須であり、大規模なワードカウントの頻度解析など、ビッグデータ解析で使用される様々な技術を取り入れることにより、ウイルスのゲノム配列の分子進化的変化を解き明かすことが重要である。

ウイルスが増殖するためには、宿主の生体防御機構を維持するための免疫システムから逃れなければならない[7-9]。増殖の際には、多くの宿主因子に依存することになるが、ヒト細胞は非ヒト宿主から侵入するウイルスにとって理想的な増殖環境であるとは考えづらい。ウイルスがヒト以外の宿主からヒト細胞内に侵入すると、ヒト細胞の中で効率よく増殖するために、ヒト細胞内のリソースを最大限利用しようとする。このためにウイルスは突然変異しながら環境に適応していくが、場合によってはヒトの免疫系により全滅させられることもある。しかし、ヒトの免疫系から逃れるようにゲノム配列を変化させて、ヒトの中で増殖する道を探り当てる場合もある。

同じゲノム G + C%をもつ種間でも、オリゴヌクレオチド組成は著しく変化する[10-11]。また、インフルエンザウイルスの場合、ヒトから分離された A 型インフルエンザ株の G＋C%は、自然宿主（トリ）から分離された株よりも低く、

ヒト・ヒト感染の間に A/U モチーフ内の CG ジヌクレオチドの割合が減少することが報告されている[11]。

短いオリゴヌクレオチドの組成は、微生物ゲノムを分類するための系統進化学の分野において、“ゲノムシグネチャ（genome signature）[6] ”と呼ばれる指標として使われてきており、ゲノム内に潜むパターンを探索する際に、非常に有効な切り口となる。

以前の我々の研究で、一括学習型自己組織化マップ BLSOM（Batch Learning Self-Organizing Map[12-13]）を使用することにより、2009 年のインフルエンザ H1N1 / 09 のパンデミックの発症[16-18]後に、多種類のオリゴヌクレオチドにおいて、はっきりとした方向性のある組成変化が生じていることを発見した[14-15]。

BLSOM は教師なし学習のタイプの機械学習のアルゴリズムであり、膨大なデータを自動クラスタリングする機能を有している。この機能を使って、インフルエンザのゲノムのオリゴヌクレオチドの頻度パターンを BLSOM で解析したところ、ウイルスが宿主を変える際に、宿主の細胞内リソースを最大限活用するために、自己のゲノムを急速に改編していくプロセスが明らかになった。しかし、BLSOM はアルゴリズムの構成上、オリゴヌクレオチドの長さが 10 を超えるころから、数テラバイトを超えるメモリを必要とし、最先端の超並列スパコンを活用しても膨大な計算時間を必要とする。核酸医薬のデザインなどに役立てるためには、現時点の BLSOM が扱えるオリゴヌクレオチドの長さの限界を超えた領域でのゲノム解析が必要となる。また、近い将来パンデミックを引き起こす可能性のある人獣共通感染症のウイルスの進化戦略を事前に予測可能なシステムを構築するためには、データ相関に基づいた 2 次元マップから間接的に時系列パターンを類推する BLSOM のようなシステムではなく、時間軸を直接取り込んだ時系列解析が必須となる。そこで、本研究では、1 から 100 までの長さのオリゴヌクレオチドの頻度を網羅的に解析することを可能にした新たなプログラムを開発し、ウイルスのオリゴヌクレオチド頻度の時系列パターンを詳細に

解析することで、診断用 PCR プライマーや核酸治療薬の開発を支援する知識発見型のシステムを構築することを目的として本研究を実施した。

1.1 RNA ウイルスの概要

RNA ウイルスは、RNA をゲノムとするウイルスで、2 本鎖 RNA ウイルス、1 本鎖 RNA ウイルスがあり、1 本鎖 RNA ウイルスは、プラス鎖の RNA ウイルス（mRNA と同じ方向に読み取られる）とマイナス鎖の RNA ウイルス（mRNA と相補的な塩基配列なので、このままでは読み取れない）がある。プラス鎖の RNA ウイルスでは、ウイルスの RNA が直接翻訳されてタンパク質の合成がおこなわれるが，マイナス鎖の RNA ウイルスでは、ウイルス粒子に入っている RNA 依存性 RNA ポリメラーゼにより、マイナス鎖 RNA を鋳型としてプラス鎖の RNA が合成され、それが翻訳されてタンパク質の合成がおこなわれる。いずれも、ウイルスの遺伝子の複製は、宿主に感染後、ウイルスの RNA と相補的な配列の RNA が作られ、それを鋳型として複製が行われる。ただし、RNA ウイルスの中には逆転写酵素を持つウイルス（レトロウイルス）が存在し、それらは逆転写酵素によりウイルスの RNA を鋳型として 2 本鎖 DNA を合成し宿主細胞に組み込まれてウイルスの RNA やウイルスタンパク質が合成される。

RNA ウイルスは，突然変異により次々とゲノムを変化させるが、1 本鎖 RNA ウイルスの遺伝子の変異が早いのは、２本鎖のような修復機能が働かないためと考えられる。

このため、人獣共通 RNA ウイルスは急速に突然変異し、大流行をひき起こす。人獣共通感染症の RNA ウイルスには、インフルエンザウイルス、エボラウイルス、MERS コロナウイルス、ジカウイルスなどがあり、いずれもパンデミックを一度引き起こしてしまえば、人類に計り知れないダメージを与えてしまう潜在的な脅威を秘めている。

1.2　インフルエンザウイルスの概要

インフルエンザウイルスは、マイナス鎖の 1 本鎖 RNA ウイルスであり、細胞膜を由来とするエンベロープを持つ。ウイルスの構造タンパク質（M1）と核タンパク質（NP 蛋白）の抗原性の違いにより、A 型インフルエンザウイルス、B 型インフルエンザウイルス､C 型インフルエンザウイルスの 3 属に分類される。A 型インフルエンザウイルスと B 型インフルエンザウイルスのゲノムは 8 分節の RNA を持っており、C 型インフルエンザウイルスは 7 分節の RNA を持っている。各分節はウイルスのタンパク質の情報をコードしている。

A 型インフルエンザウイルスでは、分節に応じて、HA（ヘマグルチニン）、NA（ノイラミニダーゼ）、PA（RNA ポリメラーゼ α サブユニット）、PB1（RNA ポリメラーゼ β1 サブユニット）、PB2（RNA ポリメラーゼ β2 サブユニット）、M（マトリクス蛋白、M1、M2）、NP（核蛋白）、NS（非構造蛋白、NS1、NS2）の 10 種類のタンパク質をコードしている。

また、エンベロープ表面上のヘマグルチニン（赤血球凝集素 HA）とノイラミニダーゼ（NA）糖蛋白 の（C 型：ヘマグルチニン－エステラーゼ（HE））抗原性の違いにより、亜型や株が分類される。A 型インフルエンザウイルスでは、ヘマグルチニンは 16 種類あり、ノイラミニダーゼは 9 種類あるので、144 種類の亜型に分類される。

北海道大学大学院獣医学研究科微生物学教室の発表によると、宿主により亜型が次のように分布している。

ニワトリ：

Ｈ1－7、Ｈ9－１１、Ｎ1－９

カモ：

Ｈ1－１６、Ｎ1－９

ヒト：

H1N1、H2N2、H3N2、H5N1、N5N6、H6N1、H7N7、H9N2、H1 0N7、H1 0N8

ブタ：

H1N1、H1N2、H1N7、H2N3、H3N1、H3N2、H3N3、H3N8、N4N6、H4N8、H5N1、H5N2、N6N6、H7N2、H9N2

ウマ：

H3N8、H7N7

同じ亜型の中でも、インフルエンザウイルスは RNA の複製に伴う突然変異の蓄積によって、HA と NA の抗原性は少しずつ変化する。これを**連続抗原変異**（**antigenic drift**）という。インフルエンザウイルス では連続抗原変異が頻繁に起こるので、毎年のように流行を繰り返す。

宿主の細胞表面に存在するウイルスレセプター（ここに HA が結合する）のシアル酸の構造が、トリとヒトでは異なるため、亜型ごとにヒトに感染しやすいかどうかが変わってくる。また、通常は一種類のウイルスしか感染しないので、大規模な抗原の変異は起こらない。ところが、ブタはトリ型とヒト型の両方に感染することがあるため、ブタの中でインフルエンザウイルスの抗原性が大きく変化して（**抗原の不連続変異**）、ヒトに感染できるようになり、新型インフルエンザウイルスとして流行が起こる。

A 型インフルエンザは、数年から数十年ごとに世界的な大流行が見られるが、これは突然別の亜型のウイルスが出現して、従来の亜型ウイルスにとって代わることによって起こっている。

代表的なパンデミックの例をいくつか列挙すると、

1918 年、スペインかぜ（H1N1）、39 年間流行

1957 年、アジアかぜ（H2N2）、11 年間流行

1968 年、香港型（H3N2）、 2 年間流行

1977 年、ソ連型（H1N1）、 2 年間流行

などがある。

2018 年の現時点では、A 型インフルエンザの亜型 H3N2 と H1N1、および B 型インフルエンザが世界中で流行している。

1.3　インフルエンザウイルス感染症

インフルエンザウイルス感染症は、主に飛沫感染で感染が広まる。つまり、インフルエンザに感染したヒトの咳などのしぶきに含まれるインフルエンザウイルスを吸い込むことにより感染がおこる。インフルエンザウイルスは、口や鼻から入り、ウイルスのヘマグルチニンが、気道上皮細胞のシアル酸残基を持つ糖タンパク質に結合することで細胞表面に吸着する。その後、細胞のエンドサイトーシスにより、細胞内に取り込まれる。細胞内の酸性環境下でウイルスが脱殻してウイルスゲノムは宿主細胞核に移動し、mRNA の合成、ウイルスゲノム RNA の複製がおこなわれ、細胞質に移動したmRNA によりウイルスのタンパク質の合成が行われる。ウイルスのゲノム RNA とウイルスタンパク質が宿主の細胞表面に移動してウイルスの粒子が形成され、ノイラミニダーゼにより細胞表面のシアル酸が分解されてウイルス粒子が放出される。これを繰り返してウイルスが増殖する。

臨床症状は A 型または B 型インフルエンザウイルスの感染後 1〜3 日間ほどの潜伏期間があり、発熱（通常 38℃以上の高熱）、頭痛、全身倦怠感、筋肉痛・関節痛などが現われる。その後、咳、鼻汁などの上気道炎症状がみられ、約 1 週間の経過で軽快する。

インフルエンザの検査には咽頭ぬぐい液や鼻腔吸引液などからのウイルスを検出する検査（ウイルス分離、PCR によるウイルスゲノムの検出、抗原検査迅速診断キット）と血清中の抗体価を測定する（ペア血清）検査がある。治療のためには短時間で診断できる迅速診断キットが有効である。しかしワクチンを製造するためには、ウイルスの分離が必要である。

ウイルスの変異が進むにつれて、既存のワクチンの効果は徐々に弱まっていき、場合によっては、全く効果が無くなるケースもある。

1.4　インフルエンザウイルス感染症の治療薬

インフルエンザの抗ウイルス薬には、アマンタジン、ザナミビル、オセルタミビルなどがある。塩酸アマンタジンは、A 型インフルエンザウイルスの表面にある M2 蛋白質に作用してインフルエンザウイルスが細胞に侵入するのを阻害する。しかし、B 型インフルエンザには M2 蛋白質がないので、アマンタジンは効かない。

リン酸オセルタミビルとザナミビルは、インフルエンザウイルスが細胞外へ出て行く際に働くノイラミニダーゼの作用を阻害することにより、増殖したインフルエンザウイルスが細胞外へ出て行くことを阻害する。A 型インフルエンザも B 型インフルエンザもノイラミニダーゼを持っているため、リン酸オセルタミビルとザナミビルは、A 型インフルエンザ、B 型インフルエンザの両方に有効である。

しかし、アマンタジン耐性インフルエンザウイルスや、ザナミビル耐性インフルエンザウイルスの出現が既に報告されている。アマンタジン耐性は、M2 タンパク質の構造の連続変異によって獲得される。ザナミビルとオセルタミビルの両薬剤に耐性を持つウイルスの出現も報告されている。

1.5 BLSOM を用いたインフルエンザウイルスのゲノム配列の方向性のある変化の予測

一括学習型自己組織化マップ BLSOM（Batch Learning Self-Organizing Map[12-13]）は教師なし学習のタイプの機械学習のアルゴリズムであり、主成分分析で求めた 2 次元格子の初期状態から出発して、学習の経過とともに膨大なデータを自動クラスタリングする機能を有している。一括学習するため、学習結果はデータの入力の順番には依存しない。

以前、我々のグループは、NCBI インフルエンザウイルスリソース[27]（http://www. ncbi.nlm.nih.gov/genomes/FLU/ ）に登録されていた全部の A 型インフルエンザウイルス、59512 セグメント（7439 株）に対して、2 連続塩基組成、3 連続塩基組成、4 連続塩基組成およびコドン組成の BLSOM 解析を行った[14]。図 1 は 4 連続塩基組成に基づいた BLSOM 解析の結果である。

格子点に単一の宿主生物に由来する配列のみが登録されている場合は、宿主のカテゴリーの色で着色し、複数の宿主由来の配列が登録されている場合は黒で着色し、どの配列も登録されていない格子点は白で着色している。

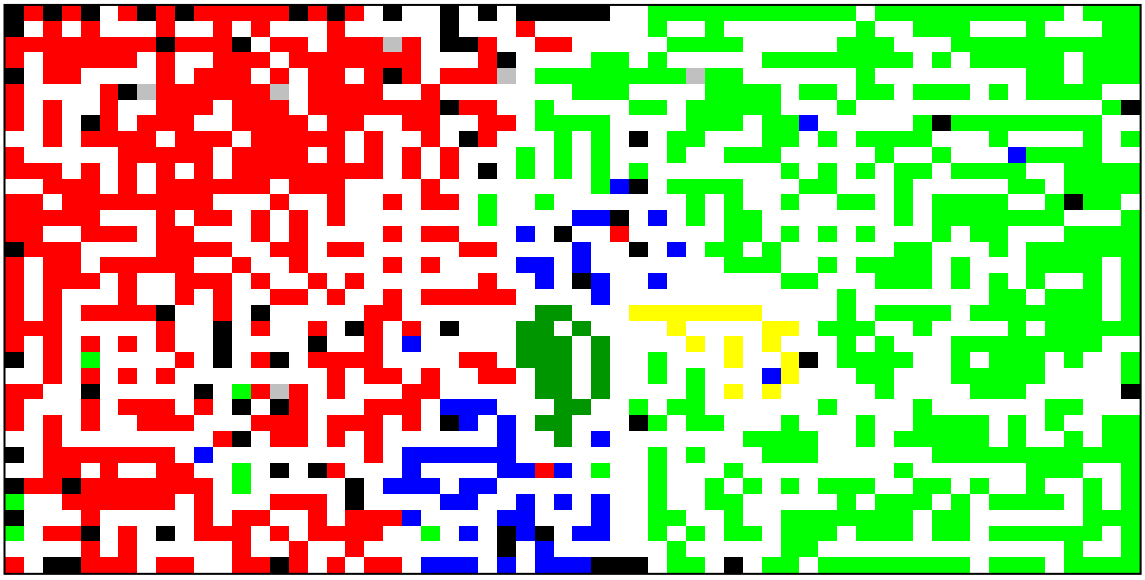
この図からわかるように、感染した宿主ごとに、極めて分離特性が強いクラスターが形成される。2 連続塩基組成、3 連続塩基組成、コドン組成の BLSOM 解析においても、はっきりとしたクラスター形成が見られた。

また、トリ、ブタ、ヒトのインフルエンザウイルスの 8 つのセグメントごとに、4 連続塩基についての BLSOM 解析を行ったところ、宿主ごとにクラスター形成が見られた。ここで、初期段階で単離された H1N1 / 09 系統のセグメント 1 とセグメント 3 の配列は、BLSOM のトリ領域内に位置した。セグメント 2 はヒトの領域に近接していた。このように、個々のセグメントの BLSOM 分析により、ウイルスのセグメントレベルでの進化の履歴を明確にすることが可能となる。

図 2 は 2 連続塩基組成、3 連続塩基組成、4 連続塩基組成に基づいた BLSOM 解析の結果である。この図からも宿主ごとにクラスター形成が見られた。一番左の列の図の濃い緑の領域は、新型インフルエンザ（H1N1/09）の領域で、右側の 3 列の図はそれぞれ、左から、2009 年 4 月、8 月、12 月に分離された H1N1/09 株に着目して表示したもので、その時に分離された H1N1/09 が配置されている格子点はピンク色に着色している。他の格子点は図 1 と同様に着色している。これにより、パンデミックの非常に早い段階（4 月）では、H1N1/09 は大部分がトリおよびブタの領域付近に存在したが、後期（12 月）になると、トリの領域近くの H1N1/09 が少なくなり、ヒトの領域に近い H1N1/09 が増えており、配列に指向性のある変化が観察できた。

図 3 は 4 連続塩基組成に基づいた BLSOM 解析の結果で、左上の図は図 1 と同様に着色している。新型の H1N1/09 の領域は、オレンジ色の円で囲んだ。また、H5N1 の領域を水色の円で囲んだ。H5N1 はトリの領域にあり、新型の H1N1/09 の領域はトリとヒトとブタに接している。真ん中の列と右の列の図は、図の上に表示してある 4 連続塩基配列の頻度が多いときはピンク色、頻度が少ないときは濃い緑で表示している。ここに表示した 4 つの 4 連続塩基組成（AGCG、CCAC、CGGC、UUUU）に関して、新型の H1N1/09 は、トリの場合と同様な傾向を示している。

表 1 は「H1N1/09 で好まれるが、ヒト株で好まれない配列」と「H1N1/09 で好まれないが、ヒト株で好まれる配列」を示した。ヒト株で好まれない配列は今後減少していき、ヒト株で好まれる配列は今後、増加していくことが予想される。



■：トリ, 1948 株　■：ヒト, 2788 株　■：新型 , 167 株　■：ウマ, 68 株
■：ブタ, 249 株　■：その他（アザラシ，トラ etc）, 130 株

単一の宿主生物に由来する配列のみが分離していた格子点は宿主カテゴリー別の色で着色し、複数の宿主由来配列が混在している場合には黒で示している。どの配列も分類されていない格子点は白色で示している。

図 1　全 A 型インフルエンザウイルス 5350 株を対象とした
4 連続塩基頻度に基づいた BLSOM 解析

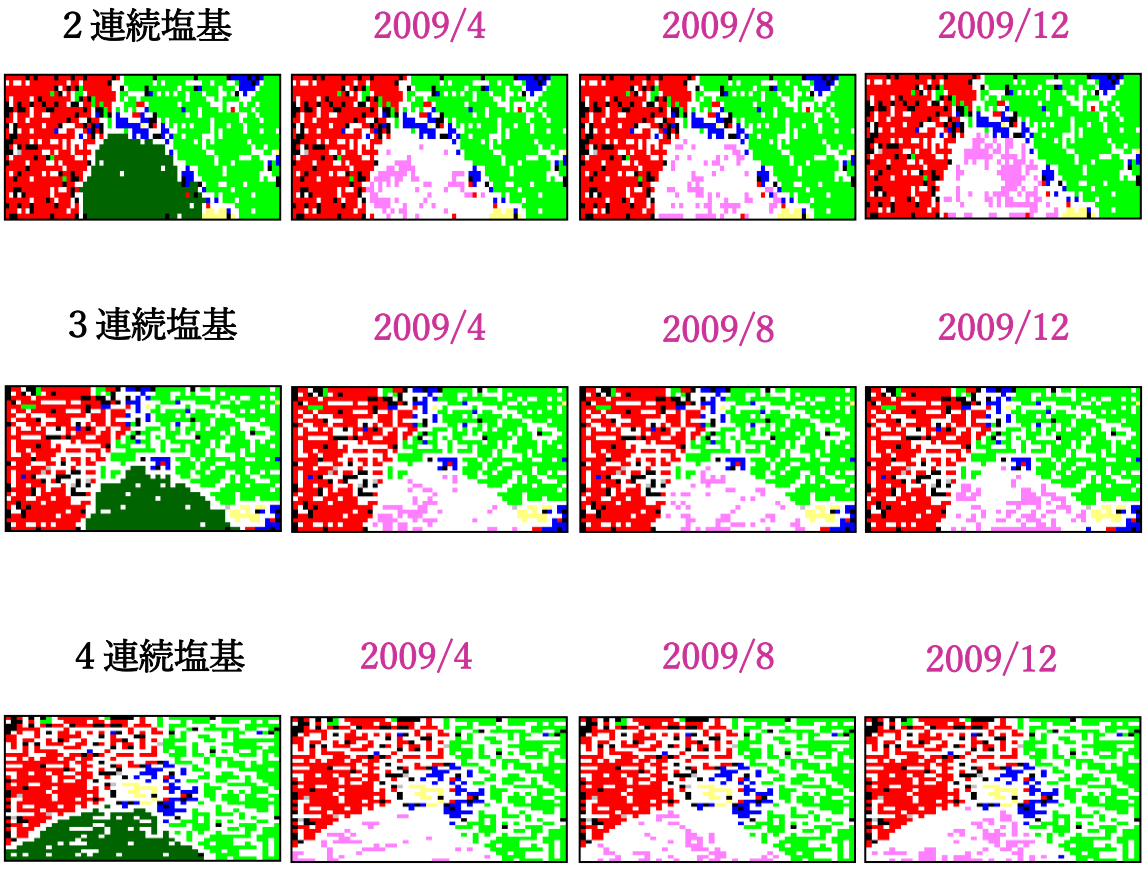


図２　インフルエンザウイルスのオリゴヌクレオチド組成の
方向性のある時間的な変化

BLSOM による教師なしタイプの機械学習の結果

■ ： トリ, 1948 株　　■ ： ヒト, 2788 株　　■ **：新型, 2256 株**　　■ ： ウマ, 68 株
■ ： ブタ, 249 株　　■ ： その他（アザラシ，トラ etc）, 130 株　　■ ： H1N1/09

Iwasaki et al. *DNA Res* 2011; 18: 125-136.

Iwasaki et al. *BMC Infect Dis* 2013; 13: 386.

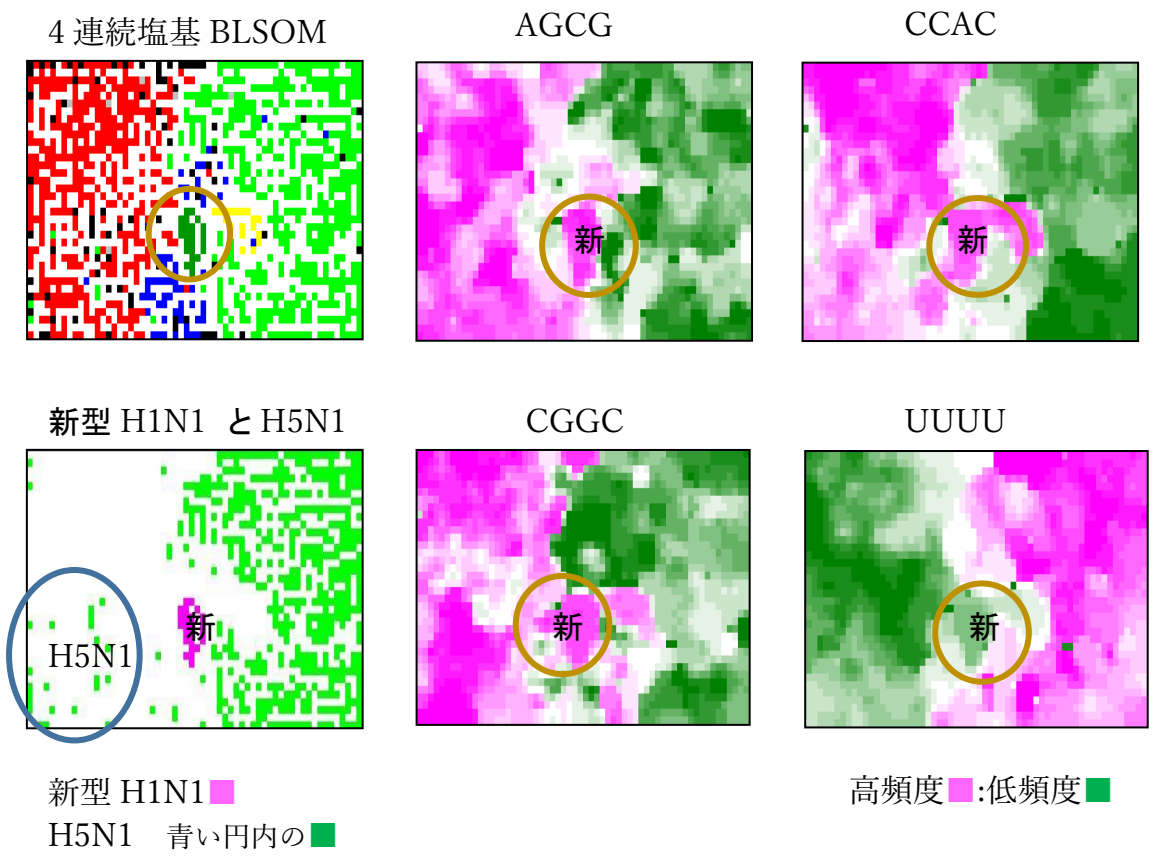


図３　４連続塩基の BLSOM 解析

新型インフルエンザ株のオリゴヌクレオチド組成の一部は、季節性のヒト由来株からずれていて、トリ･豚･馬由来に近い。

Iwasaki et al. *DNA Res* 2011; 18: 125-136.

表１　H1N1/09 で変化が予想される連続塩基配列及びコドン

H1N1/09 で好まれるが、ヒト株で好まれない配列

Codon	GCA, CAG, CUC, AAG, UUC, UCG
Di	AG, CG, GA
Tri	AGA, CAG, CCA, GCG, GUG
Tetra	AAGA, ACGG, AGAG, AGCG, AGGA, AUAA, AUCC, CACG, CCAC, CCAG, CGGC, GACG, GACU, GGCA, GUCG, GUCU, UCCA, UCUU, UGAA, UUCG

H1N1/09 で好まれないが、ヒト株で好まれる配列

Codon	CAA, UUG, AAA, UUU, ACU, GUU
Di	AA, UU
Tri	AAA, AUU, GGG, UCA, UGU, UUA, UUG, UUU
Tetra	AAAA, AAAC, AACU, AGCU, AUAG, AUUA, CAAA, GGGG, GGUU, GUCA, GUUG, UAUG, UGUA, UGUU, UUAA, UUAU, UUGU, UUUG, UUUU

Iwasaki et al.　*DNA Res*　2011; 18: 125-136.
Iwasaki et al.　*BMC Infect Dis*　2013; 13: 386.

第２章　方法

2.1　使用データ

ヒトザイールエボラウイルスと MERS コロナウイルスの株のゲノム配列については、NCBI ウイルス変異データベース（http://www.ncbi.nlm.nih.gov/genome/viruses/variation/）より、2015 年 12 月 20 日および 12 月 31 日のデータを使用した。

最近の西アフリカでのエボラウイルスの大流行はギニアで感染が開始し，シエラレオネ やリベリアへ感染が拡大した。2014 年から 2015 年にヒトから分離され、配列が決定されており、分離された日付が与えられたザイールエボラウイルス 1020 株を使用した。現地の政府によって厳しい滅菌処理が義務付けられたことが主な原因と考えられるが、かなり短めのゲノムも含まれており、配列内に未確認であることを示す「N」塩基が含まれていた[3]。これらの配列はモノヌクレオチドおよびオリゴヌクレオチド組成に影響を与えるので、N 塩基を除いた後に 18.5kb より長い、935 株のゲノムに焦点を当てた（ギニア 244 個、リベリア 156 個およびシエラレオネ 535 個の株に由来する）。また、他の地理的領域から分離された 15 株は除外した。

A 型および B 型インフルエンザウイルスの配列については NCBI インフルエンザウイルスリソース（http://www.ncbi.nlm.nih.gov/genomes/FLU/）[27]より、2015 年 9 月 1 日付の、約 25,000 株、合計 約 200,000 のセグメントの配列を使用した（8 つのセグメントが完全にそろっている株のみを対象とした）。

2.2 解析方法

ゲノム配列ごとにモノヌクレオチド、オリゴヌクレオチドの出現頻度を独自のプログラム（ソースコードは付録を参照）で計算し、それをもとに時系列解析を行った。

エボラウイルスに関しては、ギニア、リベリア、シエラレオネの 3 つの地域で分離した株のオリゴヌクレオチド組成の時系列変化を地域別に解析した。モノヌクレオチドに関しては、分離株ごとの解析と 1 か月ごとの月平均値の解析を行った。2 連続塩基以上については、ランダム突然変異やシークエンシングの不確実性に起因するデータの変化の影響を少なくするために、月平均の解析を行った。

MERS コロナウイルスの解析において、エボラウイルス、インフルエンザウイルスと比較して、MERS コロナウイルスの配列の数が少なかったため、3 株以上を有する月について時系列解析を行い、ある月の株の数が 3 株未満である場合、最も近い隣の月（少ない数の株を有する隣月）に組み込んだ。

インフルエンザウイルスに関しては、8 つのゲノムセグメントにおけるモノヌクレオチドおよびオリゴヌクレオチドの出現数をそれぞれ計算し、株ごとにそれらの出現数を合計した。エボラウイルスよりも 20 倍も多い配列が存在したため、シークエンシングの不確実性によるアーチファクトを軽減するために、1 年間に 10 株以上で 5 年以上のデータがある亜型のみを解析した。

pH1N1 以外のヒト A 型インフルエンザウイルスの亜型、ヒト B 型インフルエンザウイルスおよびトリ A 型インフルエンザウイルスに関しては、年ごとに平均値を計算して時系列解析を行った。ヒト A 型インフルエンザウイルスの亜型 pH1N1 はデータがたくさんあったので、月ごとの平均値を計算して時系列解析を行った[16-18]。ここで、多数の pH1N1 株からの誤った割り当てを防ぐために、2009 年から分離した比較的少数の古典的ヒト H1N1 株を本解析から除外した。

第３章　エボラウイルスのゲノムにおけるオリゴヌクレオチド組成の時系列変化と地域差の解析

3.1　エボラウイルスの概要

エボラウイルス属は重複の多いマイナス 1 本鎖 RNA ウイルスである。1976 年に初めて発見され、今までに、20 回以上の大流行を起こしている。自然宿主はオオコウモリだと考えられている。霊長類に感染力が強く、頭痛、発熱、筋肉痛、嘔吐、下痢、肝機能障害、腎機能障害、出血傾向（エボラ出血熱）などを起こし、致命率が高い。

本研究では、2014 年、西アフリカのギニアから開始し、リベリア、シエラレオネに拡大して大流行を起こしたエボラウイルスの株のゲノム配列について解析を行った。

3.2　エボラウイルスのゲノム配列中の
モノヌクレオチド組成の解析結果と考察

図４はエボラウイルスの各株のモノヌクレオチド（A, C, G, U）組成（%）を、エボラウイルスデータベースに記載されているウイルスが分離された最初の日の 2014 年 3 月 17 日を開始点[24]として、分離された日に従ってプロットし、採取された地域ごとに着色している。

図４より、同じ日に分離された株でも、モノヌクレオチド組成にかなりばらつきが見られた。これは、組成の多様性を示しているが、配列の突然変異がランダムに起きているためと、シークエンシングの不確実性のためであると考えられる。

時間経過による線形回帰直線では、ギニア、リベリア、シエラレオネの３つの地域に共通して A%と U%は減少傾向、C%は増加傾向が見られた。これは２つの研究グループ[2-3]によって見いだされた過剰な U から C への突然変異と一致する。

モノヌクレオチドの回帰分析では、ギニアの U%、G%を除いて、ギニア、リベリア、シエラレオネの３つの地域において、有意水準 0.01 で時間との相関なしの帰無仮説が棄却された。ギニアの U%に関しては有意水準 0.05 で時間との相関なしの帰無仮説が棄却された。

G%に関してはギニアでは減少傾向、リベリア、シエラレオネでは増加傾向が見られた。

ランダム突然変異やシークエンシングの不確実性に起因するデータのばらつきの影響を取り除くために、分離された配列を地域別に１か月ごとにまとめ、各月の平均モノヌクレオチド組成を計算して、月単位での時系列解析を行った（図５）。

それぞれの地域で 5 株以下の株しか入手できなかった数ヶ月間のデータを省略した。

図 5 より、月別の平均モノヌクレオチド組成（%）の時系列グラフでは、ギニア、リベリア、シエラレオネの 3 つの地域に共通した、はっきりとした増加傾向（C%）、減少傾向（A% , U%）が見られた。

時間との相関係数を表 2 に示した。

月別の平均モノヌクレオチド組成（%）の回帰分析では、ギニアの U%、G% を除いて、ギニア、リベリア、シエラレオネの 3 つの地域において、有意水準 0.05 で時間との相関なしの帰無仮説が棄却された。

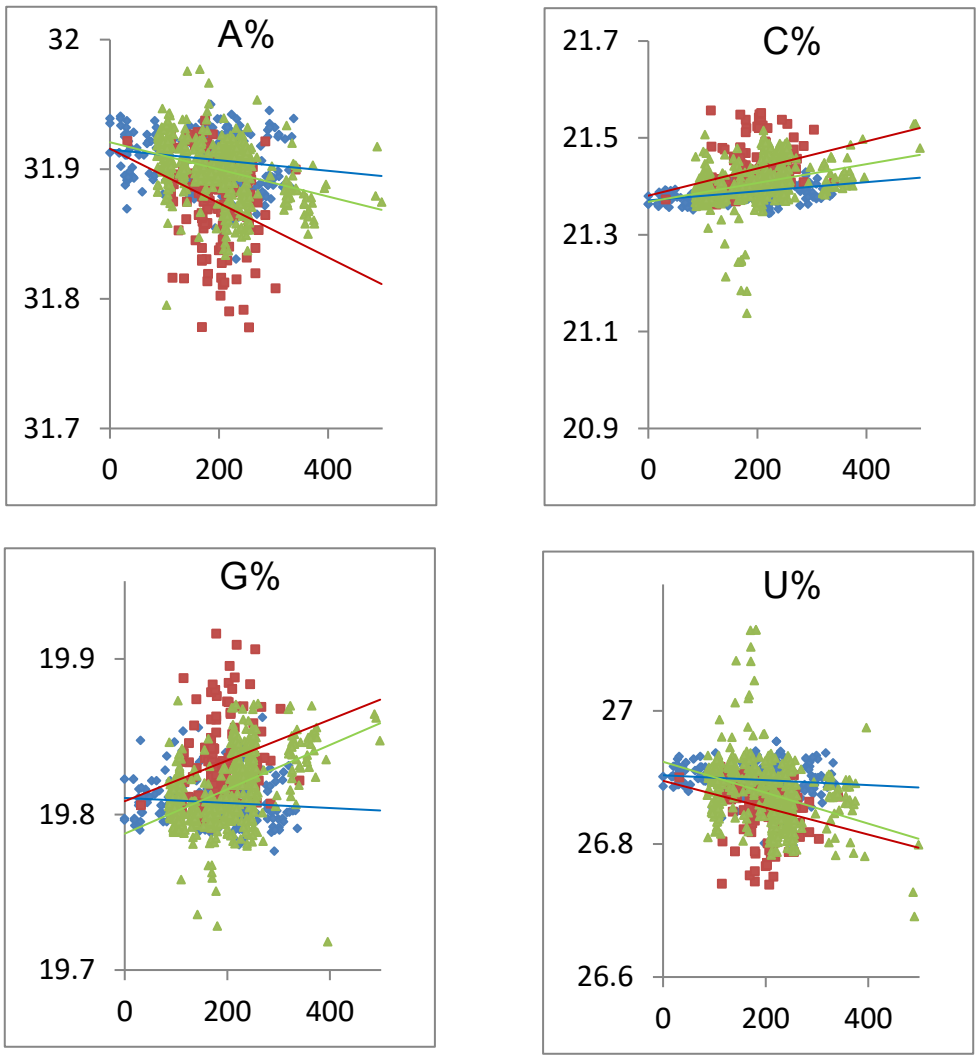


図４　エボラウイルスのゲノム配列中の塩基組成の採取日別の時系列変化

（地域別）ギニア（◆），リベリア（■），シエラレオネ（▲）

横軸は開始点（2014 年 3 月 17 日）からの日数（採取日別にプロット）を表す。

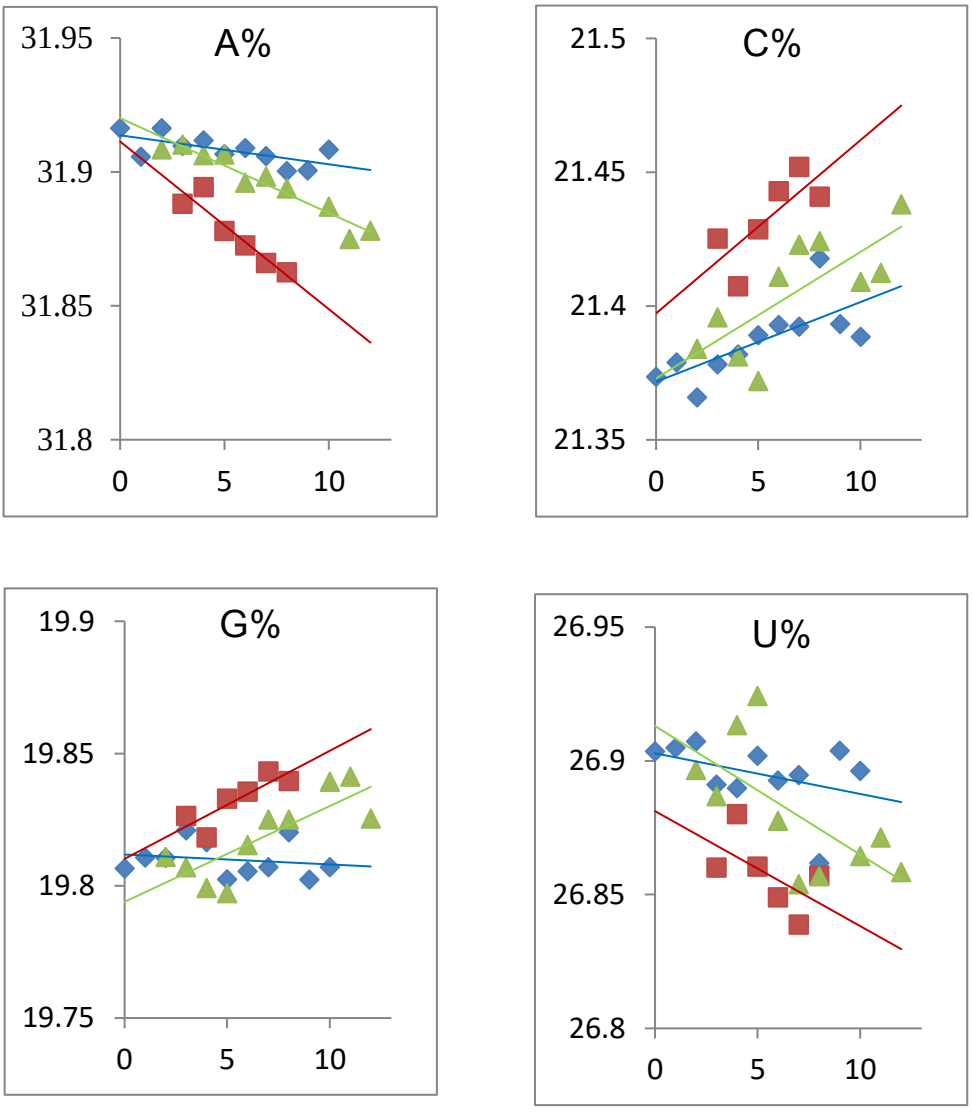


図５　エボラウイルスのゲノム配列中の塩基組成の月平均の時系列変化

（地域別）　ギニア（◆），　リベリア（■），　シエラレオネ（▲）

横軸は開始点（2014 年 3 月 17 日）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

表２　モノヌクレオチドおよび２連続塩基組成の相関係数

（月平均値）

	ギニア	リベリア	シエラレオネ
A	−0.68	−0.94	−0.93
C	0.73	0.92	0.78
G	−0.19	0.87	0.74
U	−0.40	−0.87	−0.69
AA	−0.26	−0.82	−0.87
AC	0.83	0.58	0.57
AG	−0.62	0.73	0.42
AU	−0.82	−0.72	−0.61
CA	0.35	0.6	0.47
CC	0.8	0.84	0.7
CG	−0.77	0.55	0.58
CU	0.84	0.74	0.23
GA	−0.69	0.44	0.19
GC	−0.83	0.73	0.78
GG	0.76	0.56	0.8
GU	0.94	0.68	0.39
UA	0.45	−0.46	−0.33
UC	0.47	0.79	−0.04
UG	0.8	0.63	0.33
UU	−0.86	−0.77	−0.91

3.3　エボラウイルスのゲノム配列中の２連続塩基組成の解析結果と考察

モノヌクレオチドの解析と同様に、2014 年 3 月 17 日を開始点として、地域別に、各月の２連続塩基組成の平均を計算して、平均特性の時系列解析を行った（図６）。

16 種類の２連続塩基組成の半分以上について、3 つの地域で共通の増加傾向または減少傾向がみられた。

増加傾向：　AC%, CA%, CC%, CU%, GG%, GU%, UG%

減少傾向：　AA%, AU%, UU%

月平均の２連続塩基組成の時間との相関係数は表２に示した。

図６の４つの２連続塩基組成（UU%、AU%、AC%、CC%）に関しては、有意水準 0.05 で時間との相関なしの帰無仮説が棄却された。

モノヌクレオチドにおいて、C%が増加傾向、A%、U%が減少傾向を示すので、CC%の増加傾向や AA%、AU%、UU%の減少傾向は構成モノヌクレオチドの累積効果から予測可能であるが、AC%、CU%、GU%、UG%の増加傾向は予測できない。AC%、CU%、GU%、UG%の増加傾向は２連続配列自体の特性と考えられる。

そこで、観測された２連続塩基組成の値（%）とモノヌクレオチド組成から予測される２連続塩基組成の値（%）の比、すなわち、

（観測値）÷（期待値）

を計算して、時系列変化のグラフを作成した（図７）。ただし、ここでは月別平均値をもとに計算した。

３つの地域で共通して、UU%、GC%の減少傾向、AC%、GU%、UG%、UA%の増加傾向が見られた。

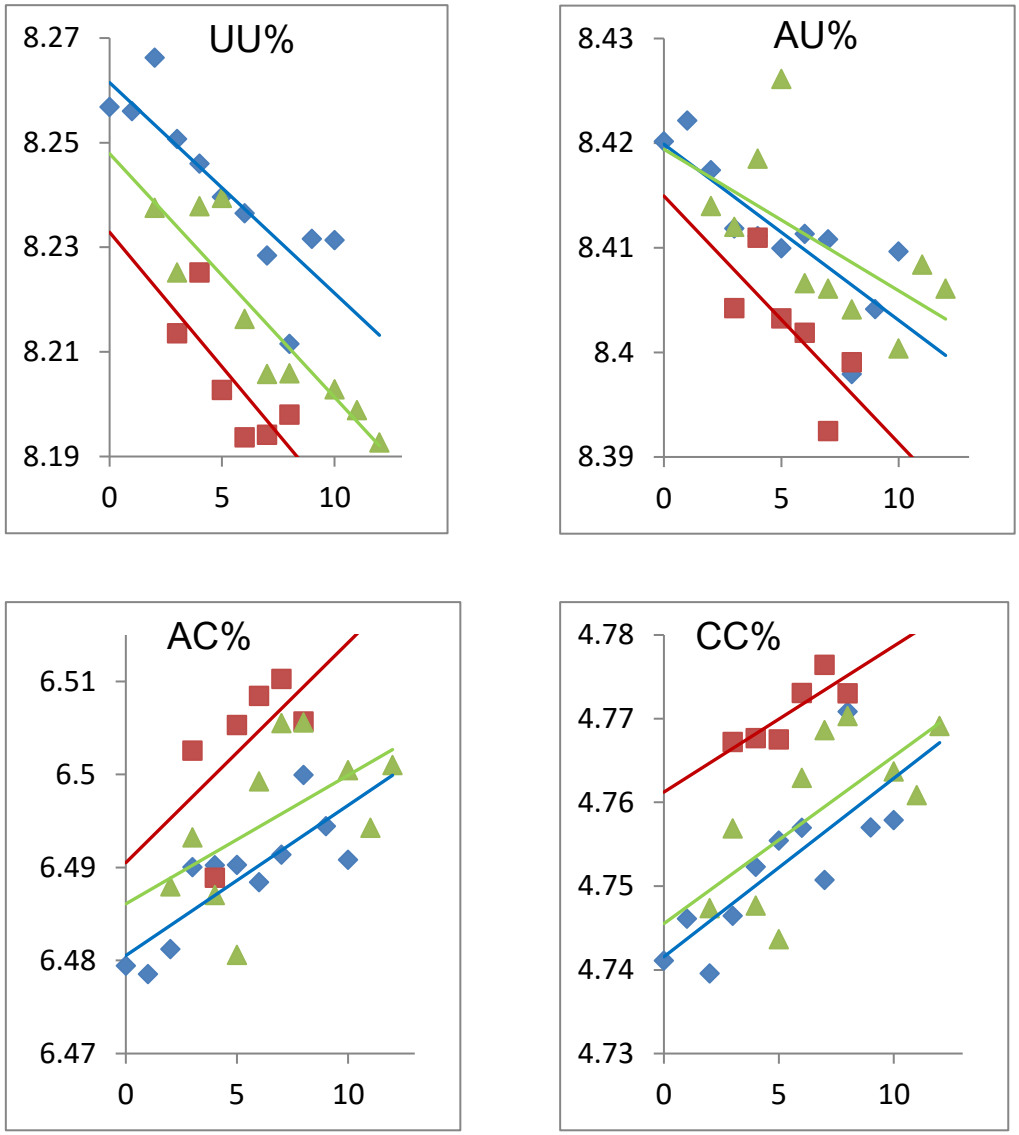


図６　エボラウイルスのゲノム配列中の２連塩基組成の時系列変化

（地域別）ギニア（◆），リベリア（■），シエラレオネ（▲）

横軸は開始点（2014 年 3 月 17 日）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

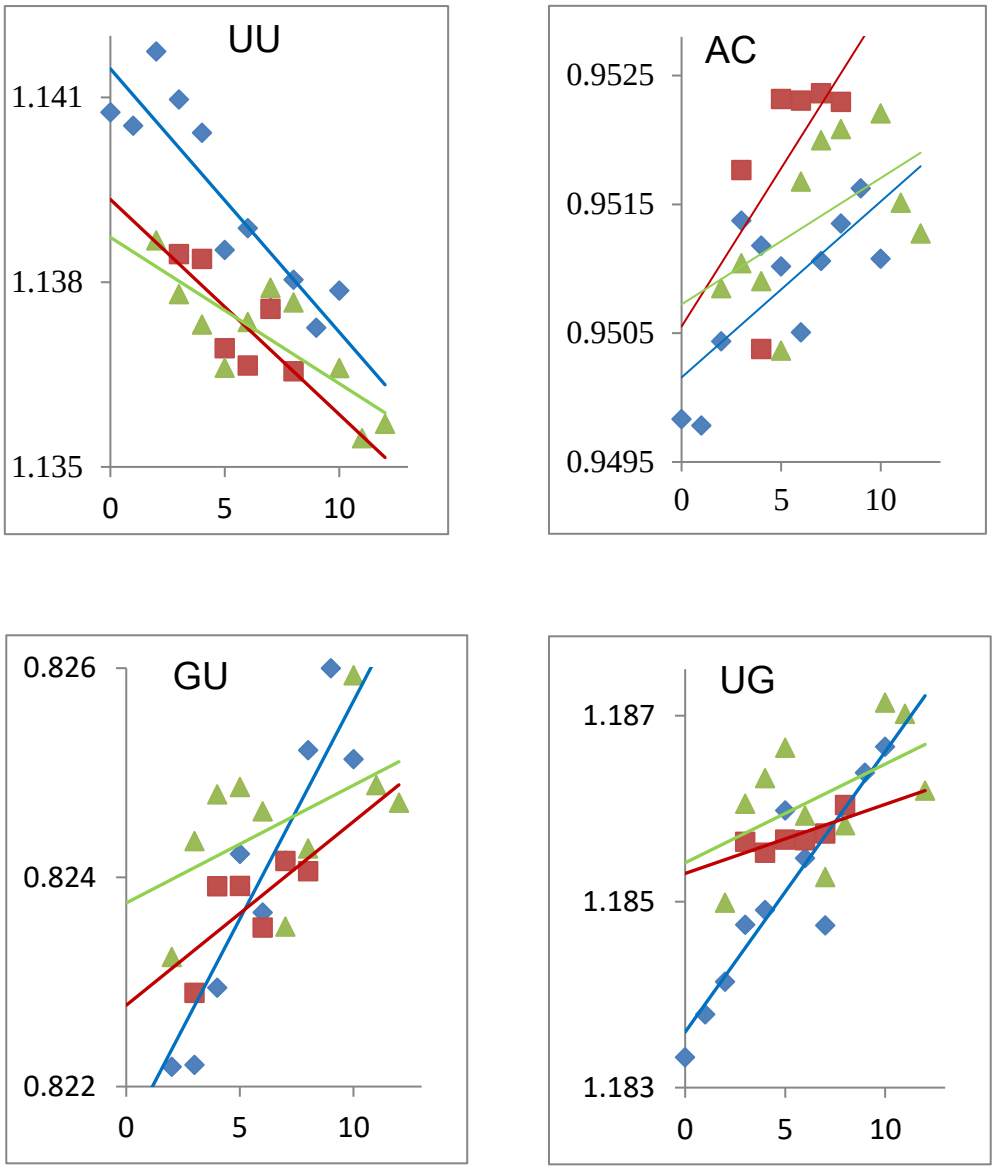


図７　エボラウイルスのゲノム配列中の２連塩基組成の
（観測値）÷（期待値）　の時系列変化
（地域別）ギニア（◆），　リベリア（■），　シエラレオネ（▲）

横軸は開始点（2014 年 3 月 17 日）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

3.4　エボラウイルスのゲノム配列中の
５連続塩基組成の解析結果と考察

最初に、すべての 1024（$= 4^5$）個の月平均の 5 連続塩基組成の相関係数を計算し、ギニア、リベリア、シエラレオネの 3 つの地域の平均相関係数でソートした。3 つの地域で共通して、384 個の 5 連続塩基組成が減少傾向を示し、50 個の 5 連続塩基組成が増加傾向を示した。

図８および図９に、明らかに強い正の相関係数、または明らかに強い負の相関係数を有する 5 連続塩基組成の時系列パターンを示した。

5 連続塩基組成のうち、時間との相関係数の絶対値の大きいものについて表３に示した。

図８および図９の８つの５連続塩基組成に関しては、有意水準 0.05 で時間との相関なしの帰無仮説が棄却された。

UUUUU、UAUUU の配列はモノヌクレオチドのＵ％とＡ％の減少傾向から説明がつくが、２連続塩基についてみられたように、CCCAA、AUUCU など、モノヌクレオチドのＵ％とＡ％の減少傾向およびモノヌクレオチドのＣ％の増加傾向からだけでは、５連続塩基組成の増加傾向の変化を説明できないものがあることが示された。

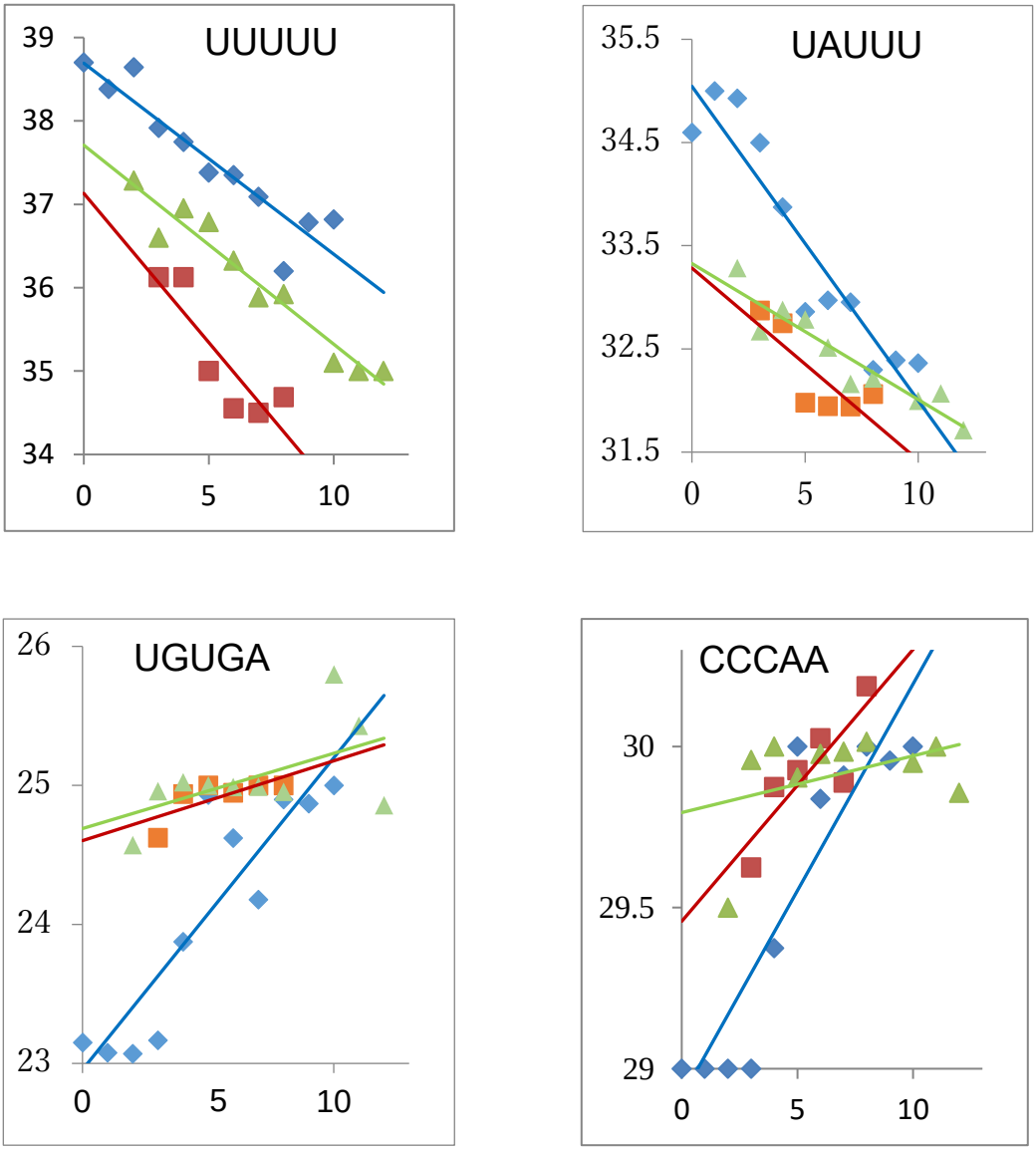


図８　エボラウイルスのゲノム配列中の５連塩基組成の時系列変化（その１）

（地域別）　ギニア（◆），　リベリア（■），　シエラレオネ（▲）

横軸は開始点（2014 年 3 月 17 日）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

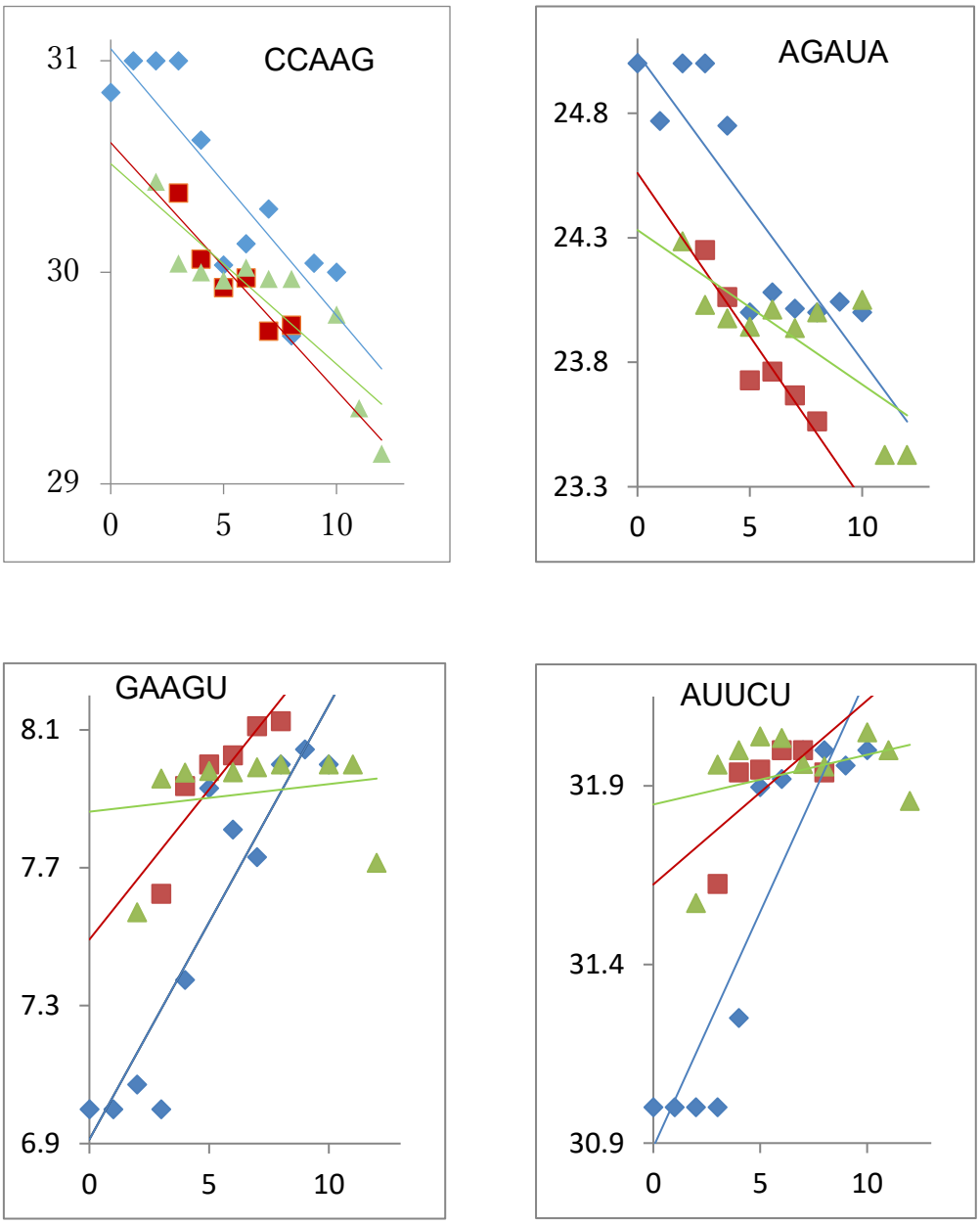


図９　エボラウイルスのゲノム配列中の５連塩基組成の時系列変化（その２）

（地域別）　ギニア（◆），リベリア（■），　シエラレオネ（▲）

横軸は開始点（2014 年 3 月 17 日）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

表３：エボラウイルスの５連続塩基組成の相関係数

	ギニア	リベリア	シエラレオネ
UUUUU	-0.93	-0.88	-0.97
UAUUU	-0.94	-0.81	-0.94
UGUGA	0.89	0.73	0.56
CCCAA	0.90	0.85	0.39
CCAAG	-0.87	-0.92	-0.89
AGAUA	-0.87	-0.94	-0.78
GAAGU	0.92	0.89	0.18
AUUCU	0.91	0.68	0.34

第４章　MERS コロナウイルスのゲノムにおけるオリゴヌクレオチド組成の時系列変化の解析

4.1　MERS コロナウイルスの概要

MERS（Middle East Respiratory Syndrome）コロナウイルス[25-26]は、コロナウイルス科ベータコロナウイルス属のウイルスである。MERS コロナウイルスはプラス鎖の一本鎖 RNA ウイルスであり（参考：国立感染症研究所）、エンベロープを持つ。ヒトコブラクダが保有宿主で[25]、典型的な症状としては発熱、咳、呼吸困難、肺炎などがある。ワクチンや特別な治療法はない。

2012 年 9 月 22 日に英国より WHO に対し、中東へ渡航歴のある重症肺炎患者から後に Middle East Respiratory Syndrome Coronavirus（MERS コロナウイルス）と命名される新種のコロナウイルスが分離されたとの報告があり、中東地域に住んでいる人、中東地域に渡航歴のある人、MERS 患者と接触のある人から、このウイルスによる中東呼吸器症候群（MERS）の症状が継続的に報告されており、医療施設や家族内等でヒト・ヒト感染が確認されている。元の自然宿主はコウモリと考えられているが[26]、この伝染病では非ヒト（主にラクダ）から繰り返しヒトに感染がおこっている[25]。

異なるヒト分離株に対する直接的な非ヒト宿主が分離株間で異なるため、最新の流行におけるラクダ由来株の情報が重要である。

NCBI ウイルス変異データベース[24]から、分離された日付が既知のヒト 91 株およびラクダの 16 株のデータを得た。

4.2 MERS コロナウイルスのゲノム配列中のオリゴヌクレオチド組成の解析結果と考察

各月の配列をグループ化し、各月および宿主についての平均化したモノヌクレオチドおよびオリゴヌクレオチド（２～５連続塩基）を解析した。

図１０は MERS コロナウイルスのモノヌクレオチド（A,C,G,U）組成の時系列変化を 2012 年 4 月からの月数に応じてプロットしている。ヒト由来株は青色の記号で表示し、Camel 由来株は小さな茶色の記号で表示している。回帰直線（青色）はヒト由来株のみで算出した。

図１０より、C％は減少傾向がみられ、U％は増加傾向がみられた。時間との相関係数を表４に示した。時間との相関なしの帰無仮説は、C％およびU％について、ラクダ由来株の有無にかかわらず、有意水準 0.01 で棄却された。G%については時間との相関なしの帰無仮説は有意水準 0.1 で棄却された。

図１１は、MERS コロナウイルスの２連続塩基組成の時系列変化を 2012 年 4 月からの月数に応じて図１０と同様にプロットしている。

ＵＵ％、UC%が増加傾向、GC%、CC%が減少傾向を示した。時間との相関係数を表５に示した。時間との相関なしの帰無仮説は、ラクダ由来株の有無にかかわらず、有意水準 0.01 で図１１のすべてについて棄却された。

図１２は、時間との相関係数が高い４つの５連続塩基組成を示した。UUUUU、GUUCU は増加傾向、ACCUC、CCACU は減少傾向を示した。時間との相関係数を表６に示した。時間との相関なしの帰無仮説は、有意水準 0.01 で図１２の４つの５連続塩基組成について棄却された。

回帰直線（図１０、図１１、図１２の青い線）はヒト株のみについて計算したが、ラクダ系統株（茶色）の発生データは回帰直線の周りに主に位置しているので、最新の伝染病の報告の中で、MERS ウイルスがラクダと人との間に繰り返し感染したという事例とも合致する。

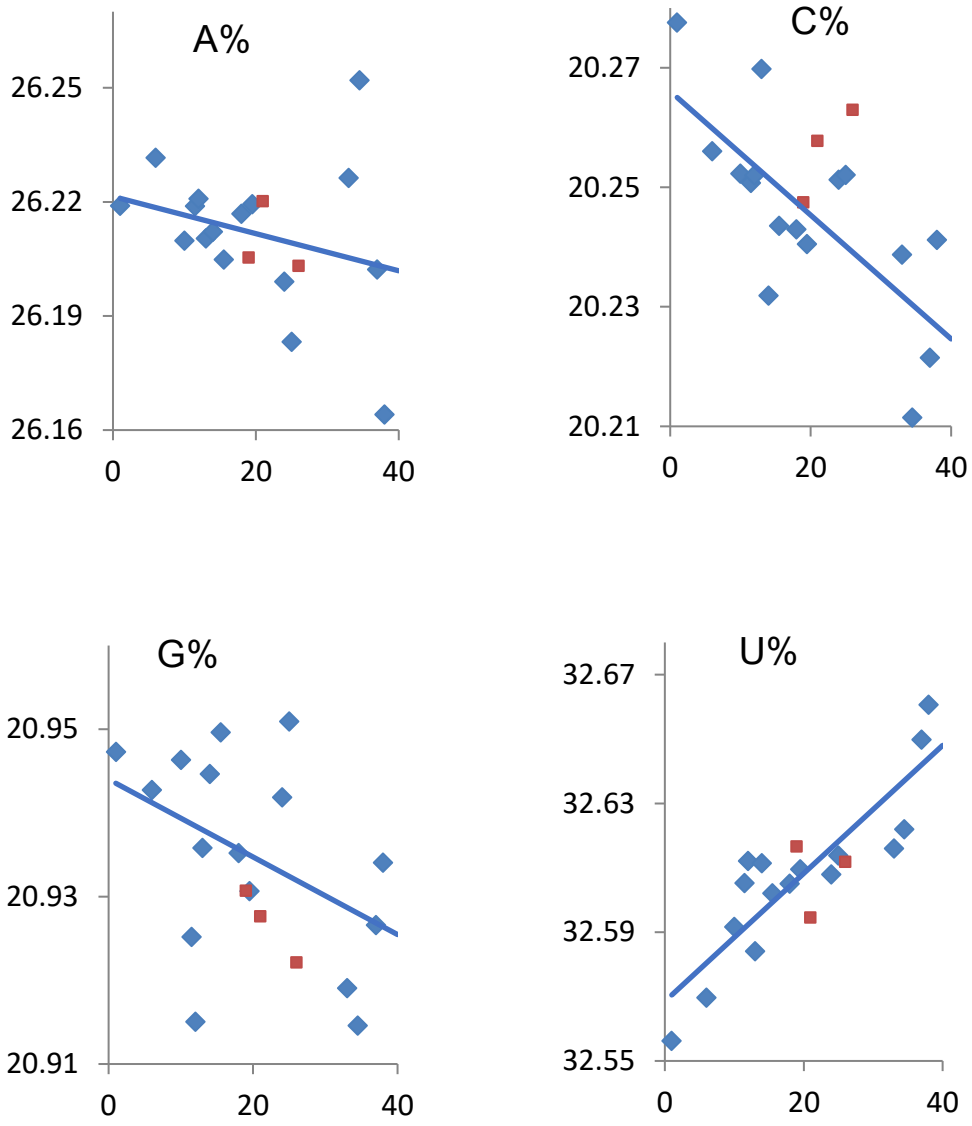


図１０　MERS コロナウイルスのゲノム配列中の塩基組成の時系列変化

横軸は開始点（2012 年 4 月）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

ヒト　（◆）、ラクダ　（■）

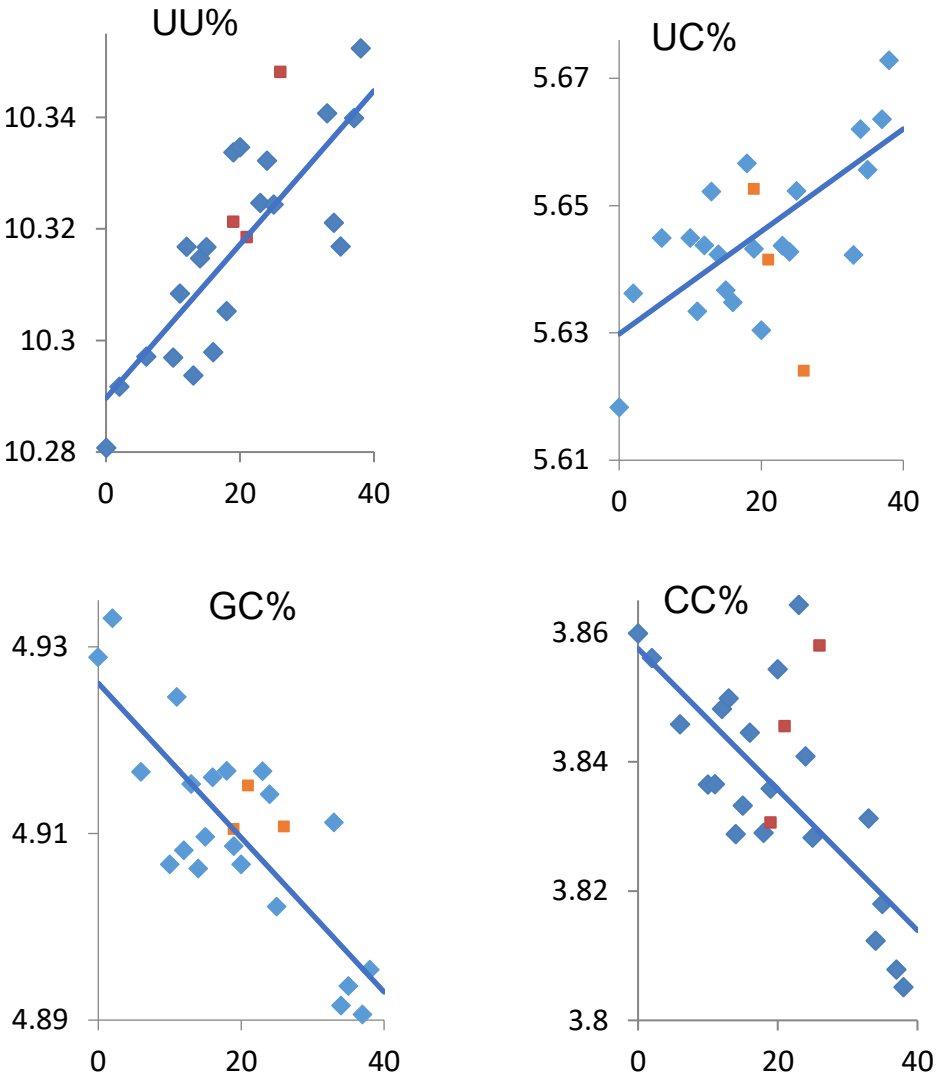


図１１　MERS コロナウイルスのゲノム配列中の
２連続塩基組成の時系列変化

横軸は開始点（2012 年 4 月）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

ヒト　（◆）、ラクダ　（■）

表４．　MERS コロナウイルスのモノヌクレオチドの相関係数

A%	-0.28
C%	-0.73
G%	-0.43
U%	0.88

表５　MERS コロナウイルスの
２連続塩基組成の相関係数

GC%	-0.81
CC%	-0.74
CA%	-0.55
UG%	-0.52
GG%	-0.39
GA%	-0.24
AG%	-0.24
AU%	-0.16
AC%	0.02
AA%	0.09
CU%	0.16
CG%	0.31
UA%	0.36
GU%	0.45
UC%	0.73
UU%	0.82

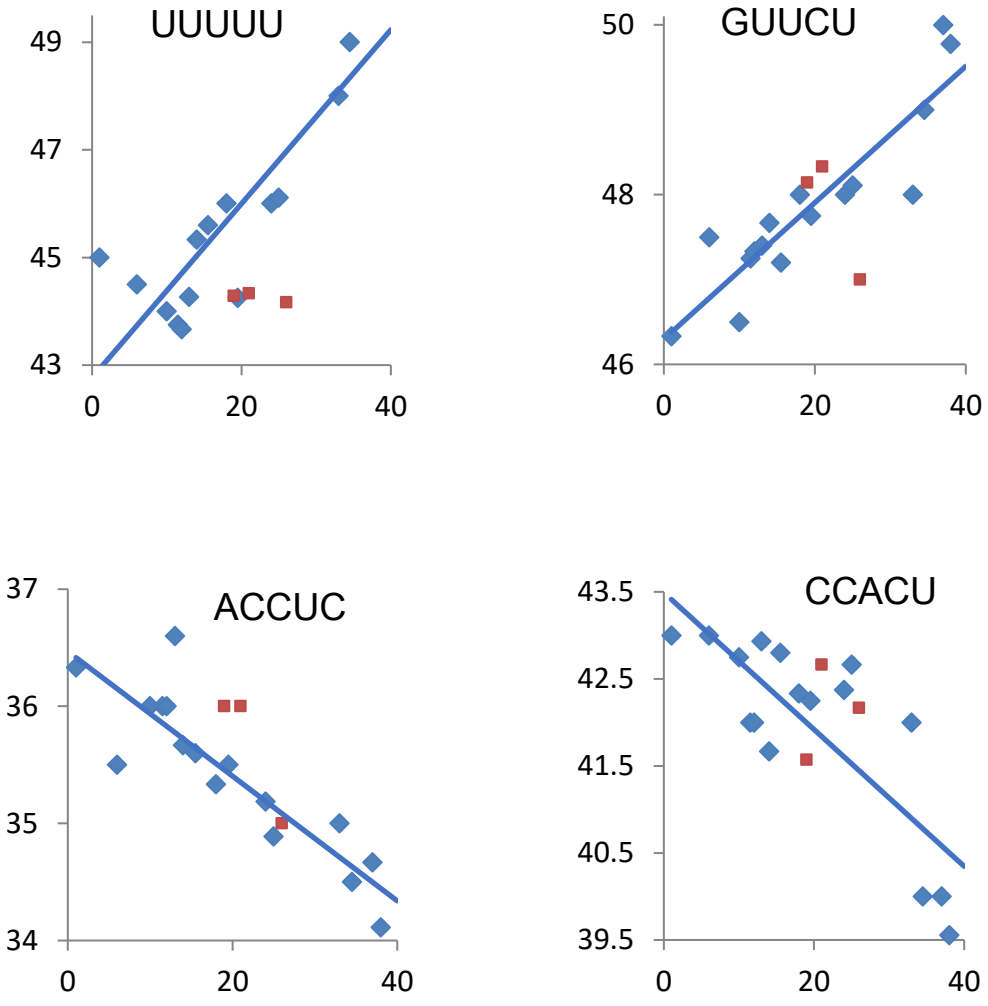


図１２　MERS コロナウイルスのゲノム配列中の 5 連続塩基組成の時系列変化

横軸は開始点（2012 年 4 月）からの月数を表す。

（月別に集計して、その平均値を月単位でプロットした）

ヒト　（◆）、ラクダ　（■）

表６　MERS コロナウイルスの
５連続塩基組成の相関係数

ACCUC	-0.89
CCACU	-0.80
GUUCU	0.90
UUUUU	0.89

第５章　インフルエンザウイルスのゲノムにおけるオリゴヌクレオチド組成の時系列変化の解析

5.1　インフルエンザウイルスの概要

インフルエンザウイルスはエンベロープを持つ、マイナス鎖の一本鎖 RNA ウイルスであり、A 型インフルエンザウイルス、B 型インフルエンザウイルス、C 型インフルエンザウイルスの 3 属がある。

A 型インフルエンザウイルス、B 型インフルエンザウイルスのゲノムは 8 分節であるが、C 型のゲノムは 7 分節になっている。また、エンベロープ表面上のヘマグルチニン（赤血球凝集素 HA）とノイラミニダーゼ（NA）糖蛋白 の（C 型：ヘマグルチニン－エステラーゼ（HE））抗原性の違いにより、亜型、株が分類される。

A 型インフルエンザウイルスの自然宿主は水禽で、そこから他の動物に感染が広まっている。しかし、鳥インフルエンザは通常はヒトには感染しない。ブタはトリ型とヒト型の両方の受容体を持つので、鳥インフルエンザにもヒトインフルエンザにも感染し、そこからヒトへ感染が波及することになる。

A 型インフルエンザでは、数年から数十年ごとに世界的な大流行が見られるが、これは突然別の亜型のウイルスが出現して、従来の亜型ウイルスにとって代わることによって起こる。

B 型インフルエンザウイルスと C 型インフルエンザウイルスの宿主は主にヒトでいずれも亜型は 1 つである。

B 型インフルエンザも世界中で流行を起こしているが、これは抗原性が少しずつ変化しているためである。

一方､C 型インフルエンザでは一度罹患するとその免疫が長く持続するため、2 度目以降は軽症でインフルエンザとは気づきにくい。

5.2 インフルエンザウイルスのゲノム配列中のモノヌクレオチド組成の解析結果と考察

ヒト A 型インフルエンザウイルスに関しては、数十年などの長い間隔で独立して流行してきた異なった亜型について、多数の配列を NCBI インフルエンザウイルスリソースから入手できたので、5 年以上にわたり 1 年間に 10 株以上ある亜型に着目して時系列解析を行った。

ヒト A 型インフルエンザウイルスの亜型（H1N1、H3N2、pH1N1）、トリ A 型インフルエンザウイルスの 9 つの亜型（H1N1、H3N2、H3N8、H4N6、H5N1、H5N2、H6N2、H7N3、H7N9）およびヒト B 型インフルエンザウイルスに焦点を当て、モノヌクレオチド組成の時系列変化を分析した。

50 ページの図 13 は 1930 年からの経過年数に従って各年の株の平均モノヌクレオチド組成をプロットした。pH1N1 に関しては、2009 年 3 月からの経過に従って、各月の株についての平均組成をプロットした。9 種類のトリ A 型インフルエンザウイルスの亜型には個別に印をつけた。

時系列変化に関して、ヒト A 型インフルエンザウイルスの 3 種類の亜型、H1N1、H3N2、pH1N1 の A%、C%、G%、U%の相関係数は、表 7 のようになった。

時間との相関がないという帰無仮説は、H3N2 の U%を除いて、0.01 の有意レベルでヒト A 型インフルエンザウイルスの 3 つの亜型で、すべての 4 つのモノヌクレオチドに対して棄却された。

pH1N1 についてみると、2009 年から 2012 年までに A%は単調に増加したが、突然 2013 年に減少して、2014 年に少し増加している（図 14）。C%,G% に関しては、2009 年から 2012 年までに単調に減少したが、突然 2013 年に増加しているように見える（図 14、図 15）。

また、1 つの大流行に由来する pH1N1 株は、複数の大流行に由来する H1N1 株及び H3N2 株より明らかに急勾配の傾きを示している。この急な変化は、1 つの流行内において、ウイルスがヒト細胞に急速に適応して増殖したことを意味している。増加の後に減少が起こるのは、そのウイルスに対する抗体を持つ人が増えていき、増殖しにくくなるからだと推測される。

表７　ヒトＡ型インフルエンザウイルスの
モノヌクレオチド組成の相関係数

相関係数	H1N1	H3N2	pH1N1
A％	0.91	0.93	0.92
C%	-0.87	-0.81	-0.88
G%	-0.91	-0.88	-0.92
U%	0.81	0.16	0.69

5.3 インフルエンザウイルスのゲノム配列中の 2 連続塩基組成の解析結果と考察

次に、ヒト A 型インフルエンザウイルスの亜型（H1N1、H3N2、pH1N1）、トリ A 型インフルエンザウイルスの 9 つの亜型およびヒト B 型インフルエンザウイルスについて、2 連続塩基組成の時系列変化を分析した。

ヒト A 型インフルエンザウイルスの亜型の H1N1 と H3N2 の場合は経過年数に対しての相関係数に関して、pH1N1 の場合は経過月数に対しての相関係数に関して調べた。この 3 つの亜型について、16 個の 2 連続塩基組成のうち 13 個は、共通に増加傾向または減少傾向を示した。

3 つの亜型について、平均相関係数の絶対値の上位 4 つの 2 連続塩基組成を、図 16 に表示した。ヒト H1N1、H3N2、pH1N1 の AA%、CG%、GC%、AG%の時間との相関係数は、表 8 のようになった。

時間との相関がないという帰無仮説は、3 つの亜型と上記の 4 つの 2 連続塩基（AA%、CG%、GC%、AG%）との 12 通りの組合せのうち、H1N1 の GC%と AG%の 2 組を除き、0.01 の有意水準で棄却された。AA%に関しては 3 つの亜型ともに増加傾向がみられた。AG%、GC%、CG%も含めてみると、モノヌクレオチドの時と同様に、ヒト A 型インフルエンザウイルスの 3 種類の亜型は、トリ由来 A 型インフルエンザウイルス亜型から離れて、ヒト B 型株へ近づく傾向があるように思われる。また、pH1N1 について細かくみると、2009 年から 2012 年までに AA%が単調増加し、2013 年に突然減少している。さらに、pH1N1 の変化は他のヒト A 型インフルエンザウイルスの亜型の変化に比べて傾きが大きい。これもモノヌクレオチドと同様の理由だと思われる。

ウイルス感染の大流行が起きると、ウイルスが広まると同時にその抗体を持つ人が増えるので、次第に流行は収まってくる。そこで、もし、つぎにまたその

病気が流行するとしたら、まだ抗体を持った人が少ないウイルス株が、次の大流行のための良好な出発株になるかもしれない。

表８　ヒトＡ型インフルエンザウイルスの
２連続塩基組成の相関係数

相関係数	H1N1	H3N2	pH1N1
ＡＡ%	0.82	0.94	0.87
CG%	-0.81	-0.70	-0.96
GC%	-0.16	-0.84	-0.90
AG%	-0.47	-0.93	-0.66

5.4　インフルエンザウイルスのゲノム配列中の 20 連続塩基組成の解析結果と考察

診断用 RT-PCR プライマーおよび治療用オリゴヌクレオチドのサイズは主に 15～30 連続塩基配列の範囲である。

そこで A 型ヒトインフルエンザウイルスの 3 つの亜型 H1N1、H3N2、pH1N1 のゲノムにおける 20 連続塩基の頻度を解析した。

合計 4 の 20 乗（約 1.1 兆）種類の 20 連続塩基うちの 140 万種類がこれらのゲノム中に見出された。各亜型について各 20 連続塩基組成の相関係数を算出した後、3 つの亜型に共通して、明らかに強い正の相関係数、または強い負の相関係数（> 0.8 または<-0.8）を有する全ての 20 連続塩基配列を抽出した。全部で 5 種類あり、負の相関係数をもっていた（表 9）。

図 17 はその 1 つの 20 連続塩基配列 ACAGCAGAGUGCUGUGGAUG の時系列の発生を提示している。他の 4 つの 20 連続塩基配列は実質的にこれと同一の減少パターンを示した。ACAGCAGAGUGCUGUGGAUG は M2 の CDS 領域に位置している。

図 17 から、H1N1 及び H3N において、元の 20 連続塩基配列の 9 番目の G は A に突然変異し（非同義置換、Ser ⇒ Asn、AGU ⇒ AAU）、pH1N1 において、元の 20 連続塩基配列の 1 番目の A は G に突然変異していった（Glu、GAA ⇒ GAG の同義置換）ものと推定できる。

表９　ヒトＡ型インフルエンザウイルスの

２０連続塩基組成の相関係数

	H1N1	H3N2	pH1N1
AGGAACAGCAGAGUGCUGUG	-0.89	-0.84	-0.81
GAACAGCAGAGUGCUGUGGA	-0.89	-0.84	-0.81
GGAACAGCAGAGUGCUGUGG	-0.89	-0.84	-0.81
AACAGCAGAGUGCUGUGGAU	-0.89	-0.83	-0.81
ACAGCAGAGUGCUGUGGAUG	-0.89	-0.83	-0.81

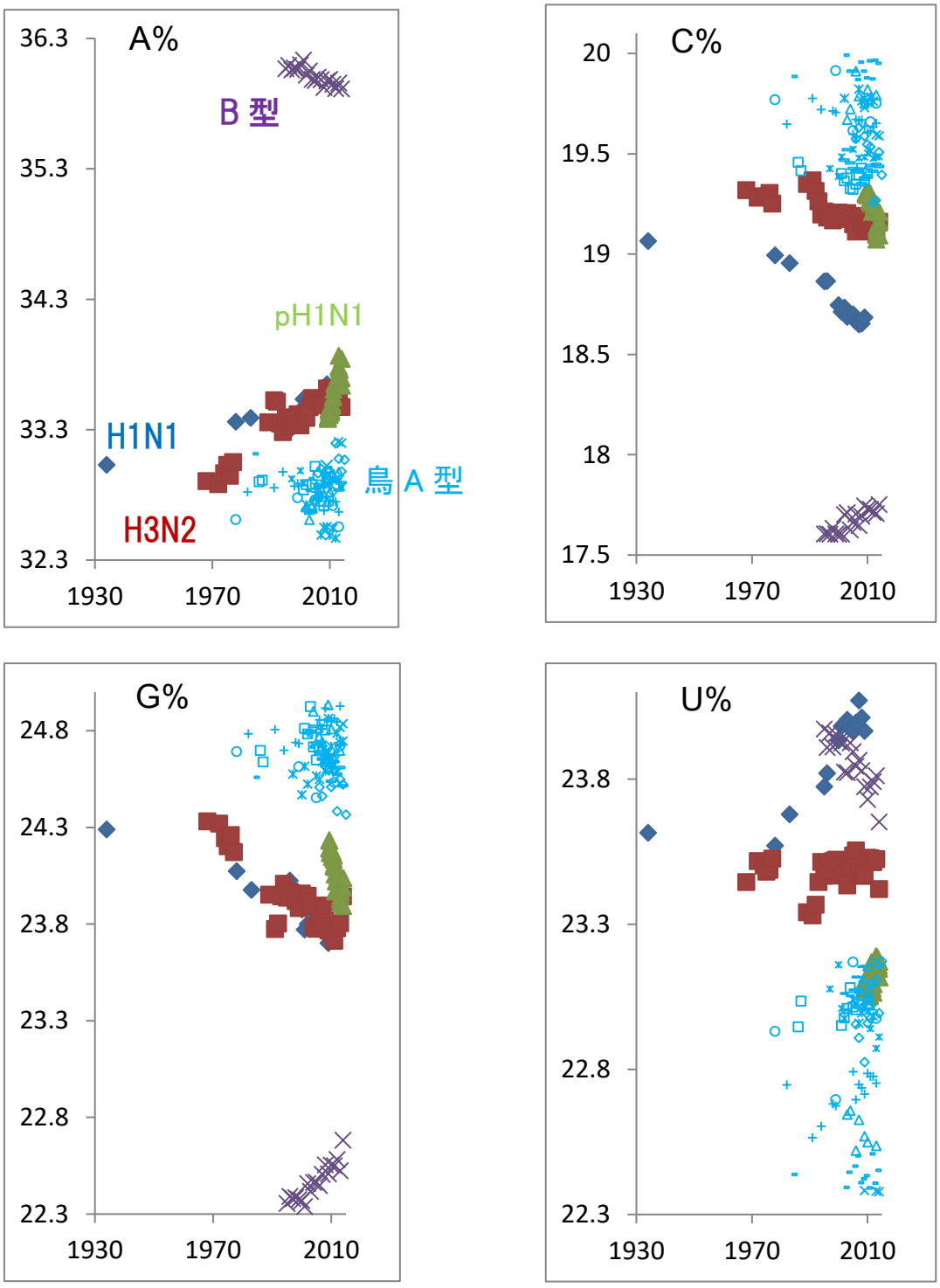


図１３　インフルエンザウイルスのモノヌクレオチド組成（%）の時系列変化（その１）

横軸は年（西暦）を表す。

縦軸は平均モノヌクレオチド組成（pH1N1 は月平均、他は年平均）を表す。

ヒト A 型の亜型：H1N1 (◆)、H3N2 (■) 、pH1N1 (▲)、ヒト B 型 (×)、

９種の鳥類 A 型の亜型：H1N1 (○)、H3N2 (+)、H3N8 (-)、H4N6 (−)、H5N1 (◇)、

H5N2 (□)、H6N2 (△)、H7N3 (×)、H7N9 (∗)

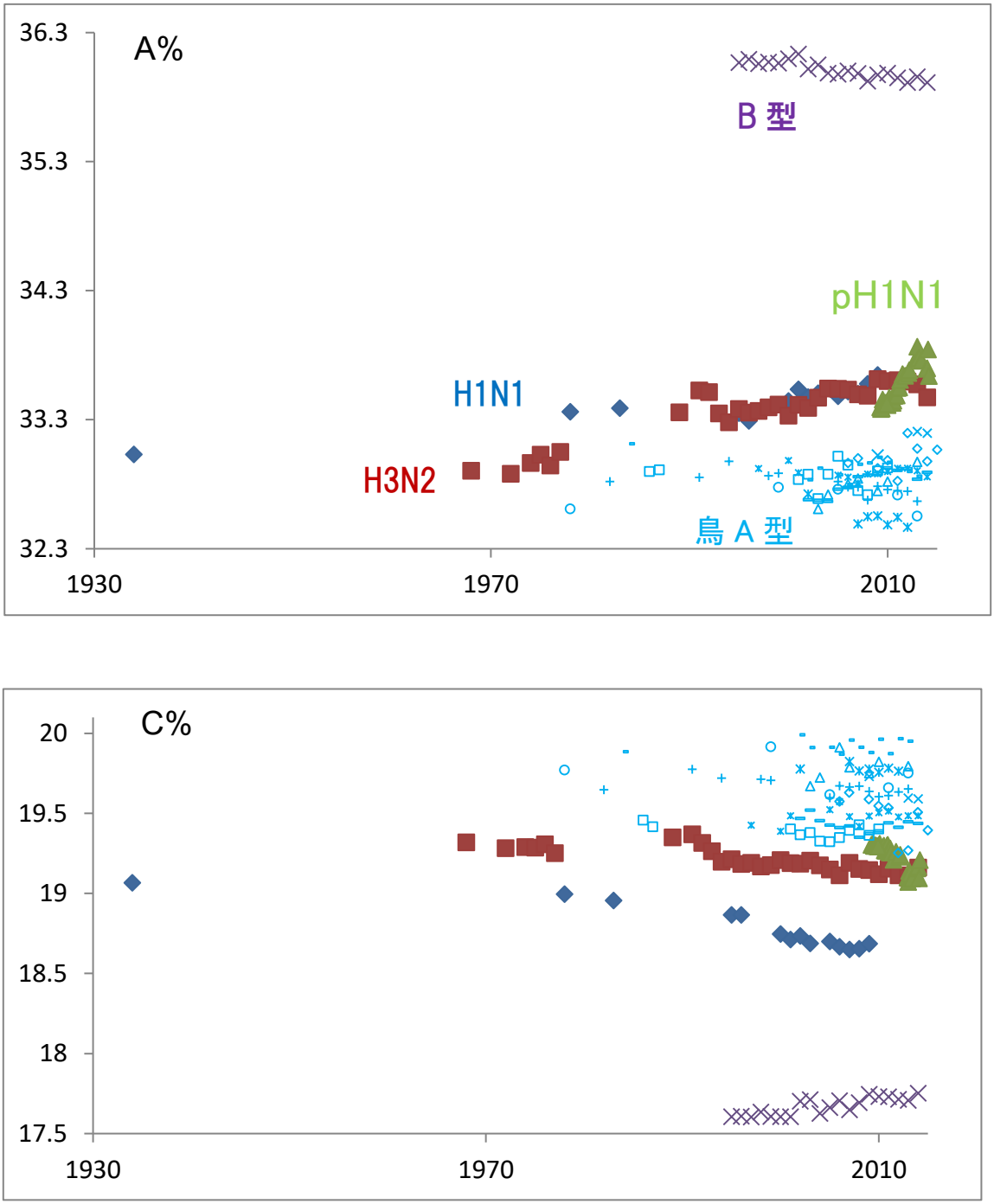


図１４　インフルエンザウイルスのモノヌクレオチド組成（%）の時系列変化（その２）

横軸は年（西暦）を表す。

縦軸は平均モノヌクレオチド組成（pH1N1 は月平均、他は年平均）を表す。

ヒト A 型の亜型：H1N1 (◆)、H3N2 (■) 、pH1N1 (▲)、ヒト B 型 (×)、

９種の鳥類 A 型の亜型：H1N1 (○)、H3N2 (+)、H3N8 (-)、H4N6 (−)、H5N1 (◇)、

H5N2 (□)、H6N2 (△)、H7N3 (×)、H7N9 (∗)

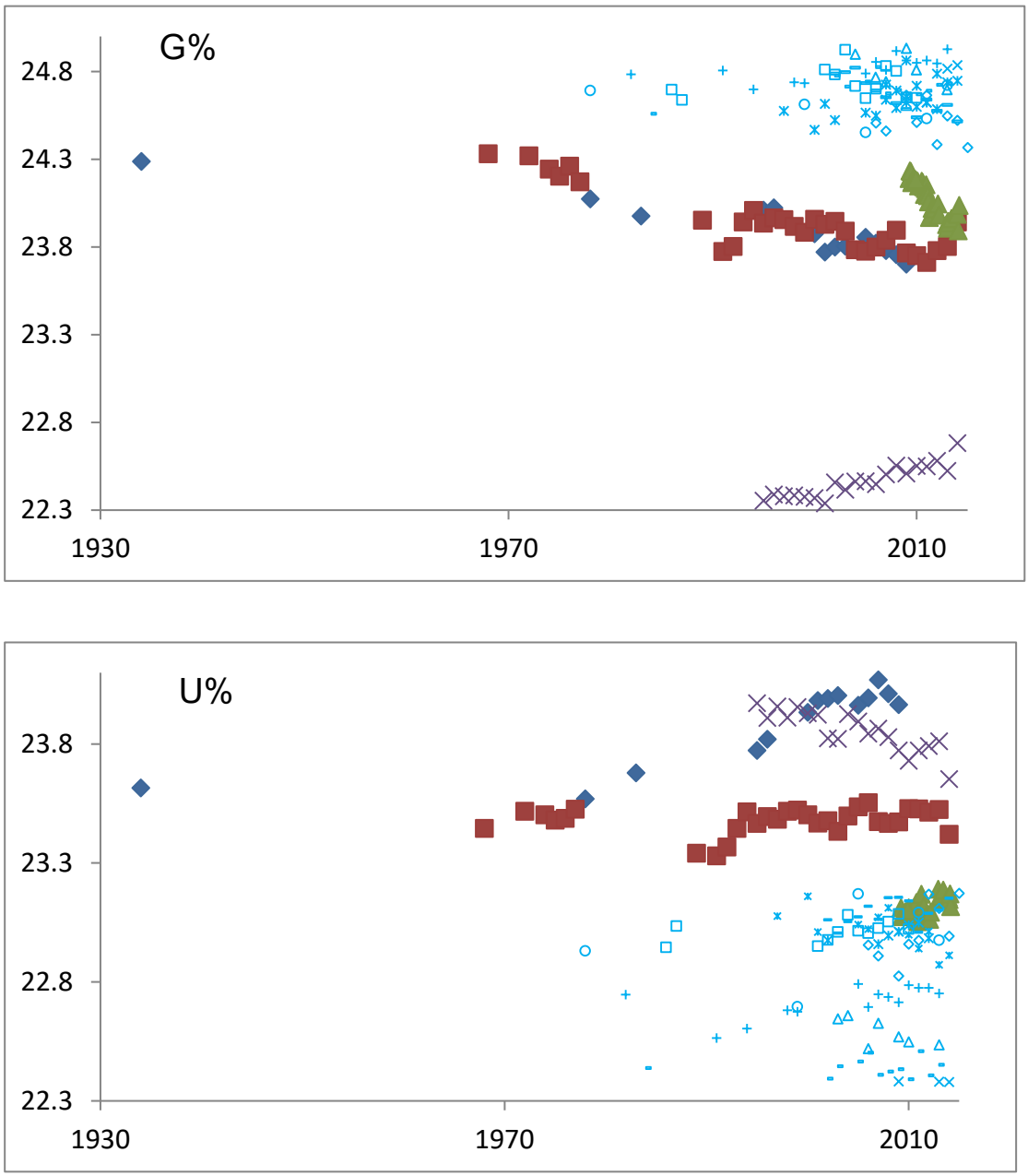


図１５　インフルエンザウイルスのモノヌクレオチド組成（%）の時系列変化（その３）

横軸は年（西暦）を表す。

縦軸は平均モノヌクレオチド組成（pH1N1 は月平均、他は年平均）を表す。

ヒト A 型の亜型：H1N1 (◆)、H3N2 (■) 、pH1N1 (▲)、ヒト B 型 (×)、

９種の鳥類 A 型の亜型：H1N1 (○)、H3N2 (＋)、H3N8 (-)、H4N6 (－)、H5N1 (◇)、

H5N2 (□)、H6N2 (△)、H7N3 (×)、H7N9 (＊)

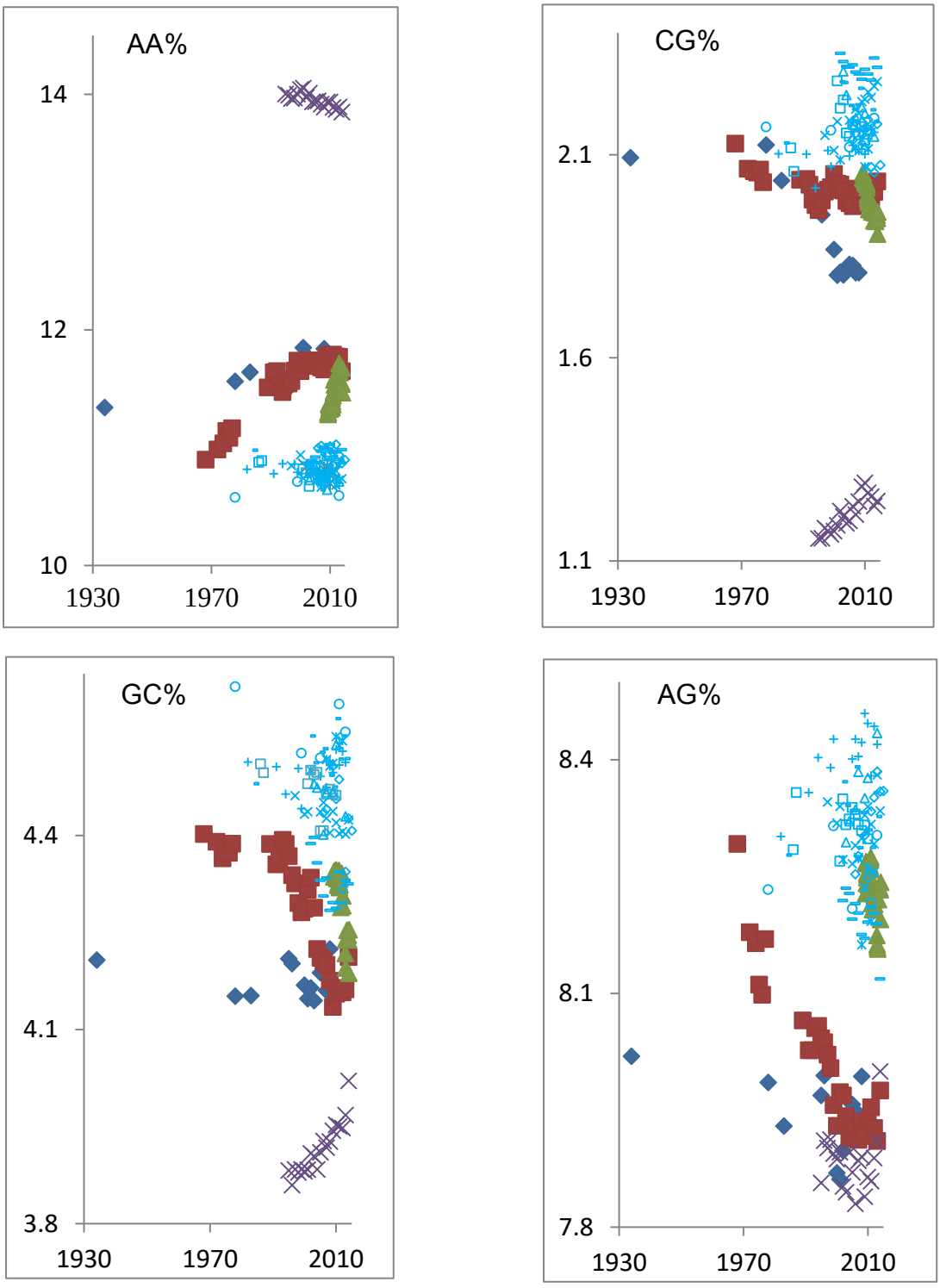


図１６　インフルエンザウイルスの２連続塩基組成（%）の時系列変化

横軸は年（西暦）を表す。

縦軸は平均モノヌクレオチド組成（pH1N1 は月平均、他は年平均）を表す。

ヒトＡ型の亜型：H1N1 (◆)、H3N2 (■) 、pH1N1 (▲)、ヒトＢ型 (×)、

９種の鳥類Ａ型の亜型：H1N1 (○)、H3N2 (+)、H3N8 (-)、H4N6 (−)、H5N1 (◇)、

H5N2 (□)、H6N2 (△)、H7N3 (×)、H7N9 (∗)

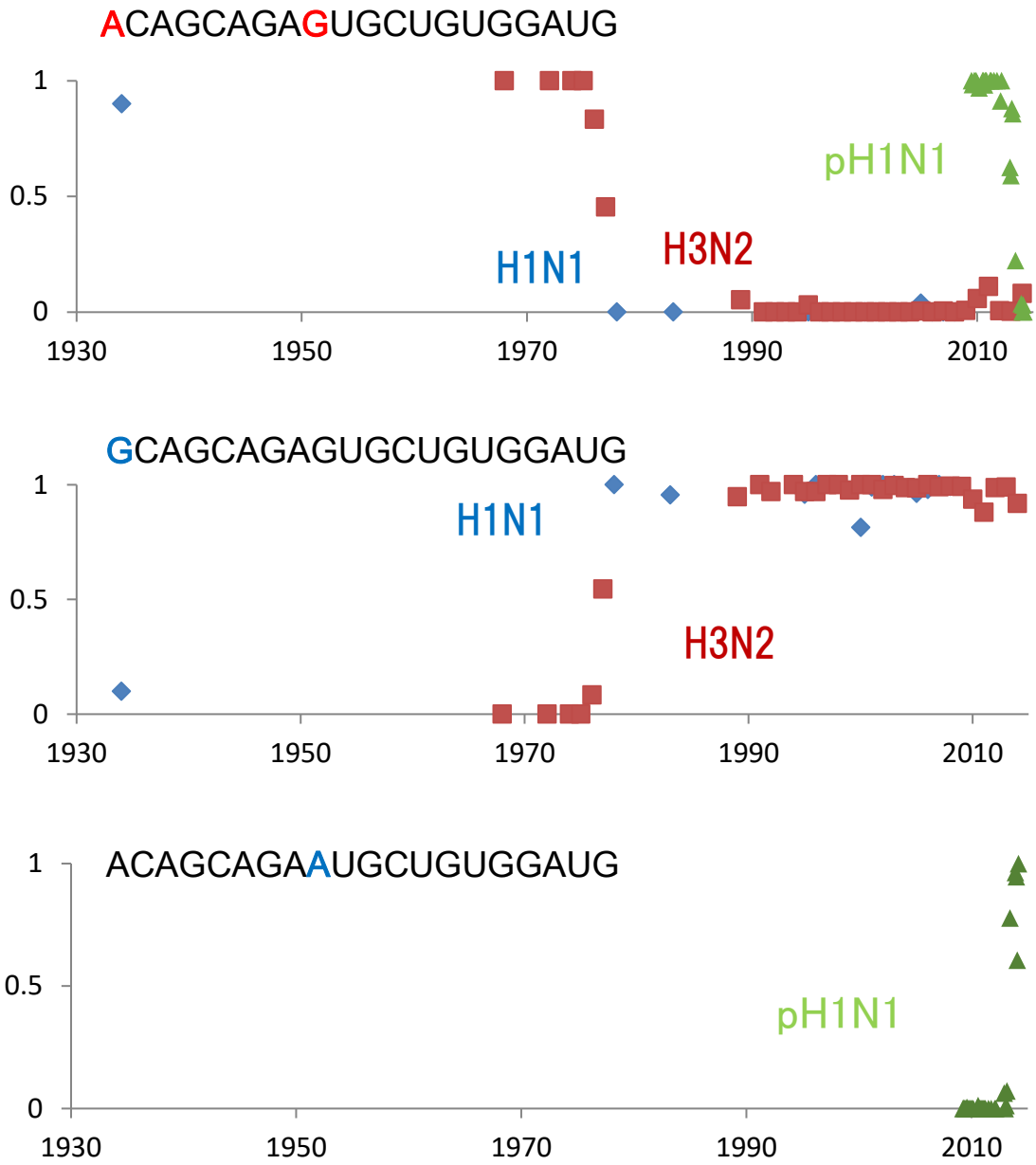


図１７　ヒトＡ型株の特定の２０連続塩基組成（%）の時系列変化

横軸は年（西暦）を表す。

縦軸は平均モノヌクレオチド組成（pH1N1 は月平均、他は年平均）を表す。

ヒトＡ型の亜型：H1N1 (◆)、H3N2 (■) 、pH1N1 (▲)

第６章　考察

人獣共通感染症 RNA ウイルスは　ヒト以外の宿主からヒトに感染して、新興感染症や再興感染症の大流行をひきおこす。RNA ウイルスは急速に突然変異を起こしながら、宿主の免疫機構から逃れ、宿主の細胞環境に適応していくだけでなく、薬剤感受性なども変化する。

そこで、本研究では、西アフリカで大流行をひき起こしたエボラウイルス、中東呼吸器症候群を引き起こした MERS コロナウイルス、インフルエンザウイルスに着目して、ウイルスがヒト以外の宿主からヒトに感染した後で、時間とともに変化するウイルスのゲノム中のオリゴヌクレオチド組成について時系列解析を行った。

エボラウイルスの３つの地域ごとのモノヌクレオチドの時系列変化を調べたところ、採取日別であっても月別であっても、一部を除けば、はっきりとした増加傾向と減少傾向が見られ、C%と G%は増加傾向を示し、A%と U%は減少傾向を示した。

同様に、エボラウイルスの２連続塩基についても、地域によらず、共通の増加傾向（AC%、CA%、CC%、CU%、GG%、GU%、 UG%）と減少傾向（AA%、AU%、UU%）がみられたが、これらの中でも、AC%、CA%、UG%の増加傾向は、モノヌクレオチド組成だけからは予測できない特徴的なパターンであり、2 連続塩基自体の特性と考えられる。

エボラウイルスの５連続塩基に関しては、モノヌクレオチド組成だけからは予測できない特徴的なパターンとして、増加傾向を示す CCCAA%と AUUCU%と、減少傾向を示す CCAAG%のパターンが見つかった。

MERS コロナウイルスでは、モノヌクレオチド組成の C%は減少傾向を示し、U%は増加傾向を示した。

MERS コロナウイルスの 2 連続塩基組成については、CC%、GC%が減少傾向、UC%、UU%が増加傾向を示した。

MERS コロナウイルスの 5 連続塩基組成については、エボラウイルスとは異なり、UUUUU%は増加傾向を示した。

モノヌクレオチド組成だけからは予測できない特徴的なパターンとして、増加傾向を示す UC%、GUUCU%と、減少傾向を示す ACCUC%、CCACU%が見つかった。

インフルエンザウイルスについては、ヒト A 型インフルエンザウイルスの亜型（H1N1、H3N2、pH1N1）、トリ A 型インフルエンザウイルスの 9 つの亜型およびヒト B 型インフルエンザウイルスに焦点を当てた。

モノヌクレオチド組成と 2 連続塩基組成の解析より、ヒト A 型インフルエンザウイルスの 3 つの亜型は、トリ A 型インフルエンザウイルス株から離れて、ヒト B 型インフルエンザウイルス株の方向に移動していく傾向が見られた。

1 つの大流行に由来する pH1N1 株は、複数の大流行に由来する H1N1 株及び H3N2 株よりも明らかに急勾配の傾きを示したが、ウイルスがヒト細胞に適応して増殖するために、急速に変化していることが原因であると考えられる。

診断用 RT-PCR プライマー[20,28]および治療用オリゴヌクレオチド[19,21-23]のサイズは、主に 15～30 連続塩基配列の範囲であることから、ヒト A 型インフルエンザウイルスの 3 つの亜型 H1N1、H3N2、pH1N1 の 20 連続塩基配列の頻度を解析した。20 連続塩基配列の場合、4 の 20 乗（約 1.1 兆）の種類のパターンの可能性があるが、実際には 140 万種類がゲノム中に見出された。これらの中から、3 つの亜型に共通して強い相関を示す上位 5 つのパターンを算出した。いずれも負の相関係数を示し、時間の経過とともに減少する傾向を示した。

本研究で扱ったどの RNA ウイルスに関しても、ウイルスごとに増加傾向、減少傾向の特徴は異なっているが、時間経過とともにオリゴヌクレオチド組成のはっきりした増加傾向や減少傾向がみられる特定の配列パターンが存在した。

非ヒトの宿主から独立した侵入によって異なる流行が始まったにもかかわらず、同じ変動傾向を示すということは、これらのウイルスは、ヒトの細胞に侵入した後のウイルスの進化戦略が似ていることが考えられる。

人獣共通感染症 RNA ウイルスにとって、ヒト細胞は必ずしも増殖に適しているわけでないので、ウイルスはゲノムを変化させることで、ヒトの細胞の環境に急速に適応していく。効率的な増殖のために生じたウイルスのゲノムの変化は、将来、自然保有宿主からヒト細胞にウイルスが侵入した際に、再び似たような変化パターンを繰り返す可能性が高い。したがって、本研究での解析の結果は、診断用 PCR プライマーの開発や、治療効果の高さが長期間にわたって持続する核酸治療薬の設計にとって、きわめて有効な開発戦略をもたらすことが期待できる。

第７章　結論と展望

西アフリカのエボラウイルスの大流行において、オリゴヌクレオチド組成における方向性のある時系列変化が、ギニア、リベリア、シエラレオネの３つの地域で共通して観察された。中東の MERS コロナウイルスの流行においても、オリゴヌクレオチド組成における方向性のある時系列変化が観察された。ヒトＡ型インフルエンザウイルスについては、数十年の間隔で、ヒト以外の宿主からヒト集団へと侵入した３つの亜型 H1N1、H3N2、ｐH1N1 に関して、オリゴヌクレオチド組成に方向性と再現性のある時系列変化が観察された。この３つの亜型に共通して認められた明らかな方向性のある変化を示す 20 ヌクレオチド程度の長いオリゴヌクレオチド類は、ヒトＡ型インフルエンザウイルスの siRNA ターゲット配列のいくつかに対応していた。なお、その siRNA の活性は実験的にも証明されている。自然保有宿主からヒト細胞にウイルスが侵入した際に、再び似たような変化パターンを繰り返す可能性が高い。オリゴヌクレオチド組成の方向性のある時系列変化や再発性を予測することは、診断 RT-PCR プライマーの開発や、長い期間に渡って有効性を保持する治療用オリゴヌクレオチド（核酸医薬）のデザインにおいて必須の技術要素となる。

一方、治療用のオリゴヌクレオチドをデザインする際、薬剤の標的になるウイルスのオリゴヌクレオチドがヒトに存在すると、ヒトの組織も薬剤のターゲットとなってしまう可能性が生じる。

そこで、副作用が起こりにくくするために、なるべく人の組織には存在しないオリゴヌクレオチドをターゲットに選ぶ必要がでてくる。

既に、我々は、ヒトのゲノムに関して 100 連続塩基組成までの使用頻度カウントを行っている。そこで、その結果と照らし合わせながら、ウイルスのオリゴヌクレオチドのターゲットを選定していけば、最適なオリゴヌクレオチドの候補を自動抽出することが可能となる。

したがって、本研究での解析の結果は、診断用 PCR プライマーの開発や核酸治療薬の設計にとって、きわめて有効な開発戦略をもたらすことが期待できる。

謝辞

本研究を遂行するに当たり、奈良先端科学技術大学院大学情報科学研究科の金谷重彦教授より、研究の大局的な戦略から解析手法などの細部にいたるまで、とても丁寧にご指導をいただきました。また、長浜バイオ大学の池村淑道名誉教授には、ゲノムのビッグデータを用いたデータ駆動型の研究方法をはじめとする様々なことに関しまして、長年にわたりご指導をいただきました。両先生よりバイオインフォマティクスの面白さと醍醐味をお教えいただき、大変感謝しております。

博士論文の審査につきまして、副指導教員を担当していただきました、奈良先端科学技術大学院大学情報科学研究科の佐藤嘉伸教授には、異分野にもかかわらず、いろいろと助言をしていただき大変感謝しております。副指導教員を担当していただきました奈良先端科学技術大学院大学情報科学研究科の Md.Altaf-Ul-Amin 准教授も、異分野にもかかわらずご審査いただき、ありがとうございました。副指導教員を担当していただきました奈良先端科学技術大学院大学情報科学研究科の小野直亮准教授には、研究内容につきまして細かい指摘をしてくださり感謝しております。また、奈良先端科学技術大学院大学情報科学研究科の佐藤哲大客員准教授からは情報解析の方法や、論文の作成方法、発表に関してのご指導をいただき大変感謝しております。

これからは、本研究をさらに発展させて、未だに猛威を振るっている人獣感染症のウイルスに対して有効な医薬品の開発などに役立てたいと思っております。

業績

査読付学術論文

1. Yoshiko Wada, Kennosuke Wada, Yuki Iwasaki, Shigehiko Kanaya, Toshimichi Ikemura.

   “Directional and reoccurring sequence change in zoonotic RNA virus genomes visualized by timeseries word count”

   Scientific Reports 6, No.36197 (2016)

   doi:10.1038/srep36197

査読付き国際会議発表

1. Yoshiko Wada, Tetsuo Sato, Naoaki Ono, Tetsuo Katsuragi, Toru Hoshida, Shigehiko Kanaya, Kotaro Minato.

   “Correlation study of the uncinate fasciculus and memory function of healthy individuals using diffusion tensor tractography and MR spectroscopy”

   2015 Joint Conference of IWAIT and IFMIA. January 11-13, 2015

   http://2015iwait-ifmia.web2.ncku.edu.tw/bin/home.php

参考資料

1. WHO. Ebola Response Team. Ebola virus disease in West Africa - The first 9 months of the epidemic and forward projections. *N. Engl. J. Med.* **371**, 1481–1495 (2014).
2. Tong, Y. G. et al. Genetic diversity and evolutionary dynamics of Ebola virus in Sierra Leone. *Nature* **524**, 93–96 (2015).
3. Park, D. J. et al. Ebola virus epidemiology, transmission, and evolution during seven months in Sierra Leone. *Cell* **161**, 1516–1526 (2015).
4. Carroll, M. W. et al. Temporal and spatial analysis of the 2014–2015 Ebola virus outbreak in West Africa. *Nature* **524**, 97–101 (2015).
5. Nichol, S. T., Arikawa, J. & Kawaoka, Y. Emerging viral diseases. *Proc. Natl Acad. Sci. USA* **97**, 12411–12412 (2000).
6. Karlin, S., Campbell, A. M. & Mrazek, J. Comparative DNA analysis across diverse genomes. *Annu. Rev. Genet.* **32**, 185–225 (1998).
7. García-Sastre, A. Inhibition of interferon-mediated antiviral responses by influenza A viruses and other negative-strand RNA viruses. *Virology* **279**, 375–384 (2001).
8. Voinnet, O. Induction and suppression of RNA silencing: insights from viral infections. *Nat. Rev. Genet.* **6**, 206–220 (2005).
9. Randall, R. E. & Goodbourn, S. Interferons and viruses: an interplay between induction, signalling, antiviral responses and virus countermeasures. *J. Gen. Virol.* **89**, 1–47 (2008).
10. Rabadan, R., Levine, A. J. & Robins, H. Comparison of avian and human influenza A viruses reveals a mutational bias on the viral genomes. *J. Virol.* **80**, 11887–11891 (2006).
11. Jimenez-Baranda, S. et al. Oligonucleotide motifs that disappear during the evolution of influenza in humans increase IFN-α secretion by plasmacytoid endritic cells. *J. Virol.* **85**, 3893–3904 (2011).
12. Kanaya, S. et al. Analysis of codon usage diversity of bacterial genes with a self-organizing map (SOM) - characterization of horizontally transferred genes with emphasis on the E. coli O157 genome. *Gene* **276**, 89–99 (2001).

13. Abe, T. et al. Informatics for unveiling hidden genome signatures. *Genome Res.* **13**, 693–702 (2003).

14. Iwasaki, Y., Abe, T., Wada, K., Itoh, M. & Ikemura, T. Prediction of directional changes of influenza A virus genome sequences with emphasis on pandemic H1N1/09 as a model case. *DNA Res.* **18**, 125–136 (2011).

15. Iwasaki, Y., Abe, T., Wada, Y., Wada, K. & Ikemura, T. Novel bioinformatics strategies for prediction of directional sequence changes in influenza virus genomes and for surveillance of potentially hazardous strains. *BMC Infect. Dis.* **13**, 386 (2013).

16. Novel Swine-Origin Influenza A (H1N1) Virus Investigation Team. Emergence of a novel swine-origin influenza A (H1N1) virus in humans. *N. Engl. J. Med.* **360**, 2605–2615 (2009).

17. Neumann, G., Noda, T. & Kawaoka, Y. Emergence and pandemic potential of swine-origin H1N1 influenza virus. *Nature* **459**, 931–939 (2009).

18. Smith, G. J. et al. Origins and evolutionary genomics of the 2009 swine-origin H1N1 influenza A epidemic. *Nature* **459**, 1122–1125 (2009).

19. Crooke, S. T. Progress toward oligonucleotide therapeutics: pharmacodynamic properties. *FASEB J.* **7**, 533–539 (1993).

20. Kageyama, T. Broadly reactive and highly sensitive assay for Norwalk-like viruses based on real-time quantitative reverse transcription-PCR. *J. Clin. Microbiol.* **41**, 1548–1557 (2003).

21. Opalinska, J. B. & Gewirtz, A. M. Nucleic-acid therapeutics: basic principles and recent applications. *Nat. Rev. Drug Discov.* **1**, 503–514 (2002).

22. Meister, G. & Tuschl, T. Mechanisms of gene silencing by double-stranded RNA. *Nature* **431**, 343–349 (2004).

23. Bennett, C. F. & Swayze, E. E. RNA targeting therapeutics: molecular mechanisms of antisense oligonucleotides as a therapeutic platform. *Annu. Rev. Pharmacol. Toxicol.* **50**, 259–293 (2010).

24. Brister, J. R. et al. Virus Variation Resource - recent updates and future directions. *Nucleic Acids Res.* **42** (Database issue), D660–D665 (2013).

25. Azhar, E. I. et al. Evidence for camel-to-human transmission of MERS coronavirus. *N. Engl. J. Med.* **370**, 2499–2505 (2014).

26. Ithete, N. L. et al. Close relative of human Middle East respiratory syndrome coronavirus in bat, South Africa. *Emerg. Infect. Dis.* **19**, 1697–1699 (2013).

27. Squires, R. B. et al. Influenza Research Database: an integrated bioinformatics resource for influenza research and surveillance. *Influenza and Other Respir. Viruses* **6**, 404–416 (2012).

28. Xie, Z. et al. A multiplex RT-PCR for detection of type A influenza virus and differentiation of avian H5, H7, and H9 hemagglutinin subtypes. *Mol. Cell. Probes* **20**, 245–249 (2006).

付録

頻度解析に使用したプログラムのソースコード（言語：C++ 14 with Boost Library）

Filename : counterSPARSE_ULL.cpp

Usage : counterSPARSE_ULL <FastaFile> <OligoLength> <SegmentLength> <OthersThrehsold>

[Example]: counterSPARSE_ULL chr12.fa 8 10000 20

```
/*
*****************************************************************************
*    DNA Pattern Counter SPARSE ULL (Unsigned Long Long) for Linux
*    -------------------------------------------------------------------------
*    int                :  -32,768 ～ 32,768
*    unsigend int       :  0       ～ 4^8 -1 = 65,535
*    unsigned long      :  0       ～ 4^16-1 = 4,294,967,295
*    unsigned long long :  0       ～ 4^32-1 = 18,446,744,073,709,551,615
*    -------------------------------------------------------------------------
*     7 連塩基 + Other  :  1  ～  4^7  + 1 = 16,385
*    15 連塩基 + Other  :  1  ～  4^15 + 1 = 1,073,741,825
*    31 連塩基 + Other  :  1  ～  4^31 + 1 = 4,611,686,018,427,387,905
*****************************************************************************
*/

#include <fstream>
#include <iostream>
#include <cstdlib>
#include <string>
#include <sstream>
#include <vector>
#include <memory>
#include <map>
#include <algorithm>
#include <climits>
#include <chrono>
#include <boost/filesystem/path.hpp>
#include <boost/filesystem/operations.hpp>

```

```
30      using namespace std;
31
32      static const string VERSION = "Ver.2.1";
33
34      static unsigned long long Dimension = 0;  // (4**OligoLength) + 1
35      static int OligoLength = 0;
36      static int OtherThreshold = 0;
37
38      class SegmentData
39      {
40      public:
41          SegmentData(unsigned long long start, const string &segment)
42          {
43              m_start = start;
44              m_segment = segment;
45              m_count.clear();
46
47              m_A = 0;
48              m_C = 0;
49              m_G = 0;
50              m_T = 0;
51              m_O = 0;
52          }
53
54          ~SegmentData()
55          {
56              map<unsigned long long, unsigned long long>().swap(m_count);
57          }
58
59          unsigned long long getStart()
60          {
61              return m_start;
62          }
63
64          map<unsigned long long, unsigned long long> getCounts()
65          {
66              return m_count;
67          }
68
```

```
69          string getSegment()
70          {
71              return m_segment;
72          }
73
74          unsigned long long getCountA()
75          {
76              return m_A;
77          }
78
79          unsigned long long getCountC()
80          {
81              return m_C;
82          }
83
84          unsigned long long getCountG()
85          {
86              return m_G;
87          }
88
89          unsigned long long getCountT()
90          {
91              return m_T;
92          }
93
94          unsigned long long getCountO()
95          {
96              return m_O;
97          }
98
99          void countElements()
100         {
101             for (unsigned long long i = 0; i < m_segment.length(); i++)
102             {
103                 char c = m_segment[i];
104                 switch (c)
105                 {
106                 case 'A':
107                 case 'a':
```

```
108                     m_A++;
109                     break;
110                 case 'C':
111                 case 'c':
112                     m_C++;
113                     break;
114                 case 'G':
115                 case 'g':
116                     m_G++;
117                     break;
118                 case 'T':
119                 case 't':
120                     m_T++;
121                     break;
122                 default:
123                     m_O++;
124                     break;
125                 }
126             }
127         }
128 
129         void countPattern()
130         {
131             for (unsigned long long i = 0; i <= m_segment.length() - OligoLength; i++)
132             {
133                 string str = m_segment.substr(i, OligoLength);
134 
135                 unsigned long long index = getIndexFromSeq(str);
136 
137                 if (m_count.find(index) != m_count.end())
138                     m_count[index] = m_count[index] + 1;
139                 else
140                     m_count[index] = 1;
141             }
142         }
143 
144         bool isUnderThreshold()
145         {
146             return (getOtherPercent() < (double)OtherThreshold);
```

```
147         }
148
149     private:
150         unsigned long long  m_start;
151         string         m_segment;
152         map<unsigned long long, unsigned long long>  m_count;
153
154         unsigned long long  m_A;
155         unsigned long long  m_C;
156         unsigned long long  m_G;
157         unsigned long long  m_T;
158         unsigned long long  m_O;
159
160         double getOtherPercent()
161         {
162             unsigned long long total = m_A + m_C + m_G + m_T + m_O;
163             double percent = 100.0 * m_O / total;
164
165             return percent;
166         }
167
168         unsigned long long getIndexFromSeq(const string &seq)
169         {
170             unsigned long long index = 0;
171
172             for (unsigned long long i = 0; i <= seq.length() - 1; i++)
173             {
174                 unsigned long long code;
175
176                 char c = seq[i];
177                 switch (c)
178                 {
179                 case 'A':
180                 case 'a':
181                     code = 0;
182                     break;
183                 case 'C':
184                 case 'c':
185                     code = 1;
```

```
186                     break;
187                 case 'G':
188                 case 'g':
189                     code = 2;
190                     break;
191                 case 'T':
192                 case 't':
193                     code = 3;
194                     break;
195                 default:
196                     return Dimension;
197                 }
198 
199                 index = 4 * index + code;
200             }
201 
202             return index;
203         }
204     };
205 
206     string getLabelArray(int oligoLength)
207     {
208         string DNA = "ACGT";
209         string labelArray = "";
210 
211         for (unsigned long long i = 0; i < Dimension - 1; i++)
212         {
213             string label = "";
214             unsigned long long max = (Dimension - 1) / 4;
215             unsigned long long n = i;
216 
217             for (int j = 0; j < oligoLength; j++)
218             {
219                 unsigned long long c = n / max;
220                 label += DNA[c];
221                 n -= c * max;
222                 max /= 4;
223             }
224 
```

```
225             labelArray += label;
226             if (i != Dimension - 2)
227                 labelArray += "¥t";
228         }
229 
230         return labelArray;
231     }
232 
233     void printMessage(ofstream &ofs, const string &message)
234     {
235         cout << message;
236         ofs << message;
237     }
238 
239 
240     int main(int argc, char* argv[])
241     {
242         if (argc != 5)
243         {
244             cerr
245             << "Usage: counterSPARSE_ULL <Filename> <OligoLength> <SegmentLength> <OthersThrehsold>"
246             << endl
247             << "[Example]: counterSPARSE chr12.fa 8 10000 20" << endl;
248 
249             return EXIT_FAILURE;
250         }
251 
252         string filename(argv[1]);
253         int oligoLength = atoi(argv[2]);
254         int segmentLength = atoi(argv[3]);
255         int otherThreshold = atoi(argv[4]);
256 
257         if (oligoLength >= 32)
258         {
259             cerr << "Oligo Lenth < 32" << endl;
260             return EXIT_FAILURE;
261         }
262 
263         boost::filesystem::path path(filename);
```

```
264         string stem = path.stem().string();
265
266         string logFilename = stem + ".log";
267         ofstream logfs(logFilename.c_str());
268
269         stringstream message;
270         message.str("");
271
272         message << "countSPARSE_ULL [" << VERSION << "]" << endl;
273         message << "    sizeof(unsigned long long) = " << sizeof(unsigned long long) << endl;
274         message << "    ULLONG_MAX = " << ULLONG_MAX << endl;
275         printMessage(logfs, message.str());
276         message.str("");
277
278         auto time_start = chrono::system_clock::now();;
279
280         ifstream ifs(filename.c_str());
281         if (ifs.fail())
282         {
283             cerr << "Can't find a file [" << filename << "]" << endl;
284             return EXIT_FAILURE;
285         }
286
287         string annotation;
288         getline(ifs, annotation);
289
290         message << "Anotation      = [" << annotation << "]" << endl;
291         printMessage(logfs, message.str());
292         message.str("");
293
294         string line;
295         stringstream ss;
296
297         while (getline(ifs, line))
298             ss << line;
299
300         string allSeq = ss.str();
301         unsigned long long totalLength = allSeq.length();
302         message << "Total Length   = [" << totalLength << "]" << endl;
```

```
303         printMessage(logfs, message.str());
304         message.str("");
305 
306         unsigned long long dimension = 1;
307         for (unsigned long long i = 0; i < oligoLength; i++)
308             dimension *= 4;
309         dimension++;
310 
311         message << "Dimension      = [" << dimension << "]" << endl;
312         message << "OtherThreshold = [" << otherThreshold << "]" << endl;
313         printMessage(logfs, message.str());
314         message.str("");
315 
316         Dimension = dimension;
317         OligoLength = oligoLength;
318         OtherThreshold = otherThreshold;
319 
320         unsigned long long segmentCount = totalLength / segmentLength;
321         vector<SegmentData> segmentVector;
322 
323         for (unsigned long long i = 0; i < segmentCount; i++)
324         {
325             unsigned long long start = i * segmentLength;
326 
327             auto segmentSeq = allSeq.substr(start, segmentLength);
328 
329             auto segment = make_shared<SegmentData>(start, segmentSeq);
330             segment->countElements();
331 
332             if (segment->isUnderThreshold())
333             {
334                 segment->countPattern();
335                 segmentVector.push_back(*segment);
336             }
337         }
338 
339         string outputFilename = stem + ".cnt";
340 
341         message << "OutputFile     = [" << outputFilename << "]" << endl;
```

```
342         message << "Data Count     = [" << segmentVector.size() << "]" << endl;
343         printMessage(logfs, message.str());
344         message.str("");
345         logfs.flush();
346 
347         ofstream ofs(outputFilename.c_str());
348 
349         //ofs << getLabelArray(OligoLength) << endl;
350 
351         ofs << "%WINDOWSIZE" << "¥t" << segmentLength << endl;
352         ofs << "%STEPSIZE" << "¥t" << segmentLength << endl;
353 
354         ofs << annotation << endl;
355 
356         for (unsigned long long i = 0; i < segmentVector.size(); i++)
357         {
358             auto segment = segmentVector[i];
359             unsigned long long start = segment.getStart();
360             unsigned long long length = segment.getSegment().length();
361 
362             ofs << "#" << (start + 1) << "_" << (start + length)
363                 << "¥tA : " << segment.getCountA()
364                 << "¥tC : " << segment.getCountC()
365                 << "¥tG : " << segment.getCountG()
366                 << "¥tT : " << segment.getCountT()
367                 << "¥t- : " << segment.getCountO()
368                 << endl;
369 
370             map<unsigned long long, unsigned long long> m = segment.getCounts();
371 
372             map<unsigned long long, unsigned long long>::iterator it;
373             for (it = m.begin(); it != m.end(); it++)
374             {
375                 ofs << it->first << ":" << it->second << "¥t";
376             }
377 
378             ofs << endl;
379         }
380 
```

```
381         ofs.close();
382
383         auto time_end = chrono::system_clock::now();
384
385         double elapsed =
386             (double)chrono::duration_cast<chrono::milliseconds>(time_end - time_start).count();
387         elapsed /= 1000.0;
388
389             message << "TIME [" << outputFilename << "] : "
390             << elapsed << " (sec)" << endl << endl;
391         printMessage(logfs, message.str());
392         message.str("");
393
394         logfs.close();
395     }
```