汎用ゲノム解析環境
G-language System の設計と開発

環境情報学部　荒川 和晴

Abstract

1996 年に *Mycoplasma genitalium* のコンプリートゲノムが発表されてから今日に至るまで、バイオインフォマティックスが果たした役割は目覚しい。だが、短いながらもその歴史はすでに今後の発展に向けていくつか素通りできない課題を明らかにしている。それは即ち、膨大な情報量を処理することが必然となるバイオインフォマティックスをより効率化させることである。本論は 1. 現在個別に開発されている解析ソフトウェアの統合開発環境を構築、２．既存の解析ソフトウェア・手法・結果を再利用可能な方法で蓄積、３．重複する解析の為の汎用統合解析システムの構築、の三点を実現することでバイオインフォマティックスが現在直面する課題を解決する試みである、*G-language Project* の意義と思想について議論する。また、それに基くシステムの実装と開発について詳細に論じる。

１　はじめに

1.1　本研究の意義

ヒトゲノムのドラフトシーケンス読み取りがついに昨年完了し、生命科学はついに本格的にポスト＝ゲノムの時代への幕を開けた。A・T・G・C 四文字から成るデジタルデータを統合的・網羅的に解析し、根幹から生命の理解を探るゲノム解析はこれからますます重要度を増すことだろう。

1996 年に *Mycoplasma genitalium* [1] のコンプリートゲノムが発表されてから今日に至るまで、バイオインフォマティックスが果たした役割は目覚しい。[2] だが、短いながらもその歴史はすでに今後の発展に向けていくつか素通りできない課題を明らかにしている。それは即ち、システムとしてのバイオインフォマティックスをより効率化させることである。日夜読み取られるゲノムを網羅的に解析し、さらに新たな知見を発見する為には膨大な時間と人材を必要とする。このシステムを効率化させるには、少なくとも以下の点を達成しなければならない。

１．現在個別に開発されている解析ソフトウェアの統合開発環境を構築
２．既存の解析ソフトウェア・手法・結果を再利用可能な方法で蓄積
３．重複する解析の為の汎用統合解析システムの構築

第一点は新規ソフトウェアの開発コスト及び時間を軽減することを目的とする。現在バイオインフォマティックスの研究者は独自にソフトウェアを開発することでゲノムの配列データを情報学的に解析しているが、その際に用いられる基本メソッドまでもが独自に開発されている場合が多い。これは解析ソフトウェアの汎用性という観点から見れば非常に非効率的であり、再利用及び知の共用という観点からも不足が否めない。さらに、多くの研究者が同様のメソッドや基本ライブラリを開発することは重複する作業が頻出することを意味し、これにより生命情報学という学問分野における開発コストの多くが無駄に使用されてしまっていると言えるだろう。ここで、基本メソッドをサポートし、ゲノム解析に適した汎用開発プラットフォームが存在すれば大幅に開発効率を上げることが可能になる。

第二点は、バイオインフォマティックスの研究者達の解析手法及びソフトウェアを統合開発環境の上でパブリックドメインのものとすることで、幅広い手法を解析目的に合わせて多様に選択し利用することを可能にする。[3] これはソフトウェアという容易に複製及び再利用可能な道具を使用し、解析目的の為に幅広い手法を選択できるバイオインフォマティックスならではの効率化と言えるだろう。インターネット上で充実しつつある生命情報学データベースの形でバイオインフォマティックスによる解析結果と連動することで、既存の蓄積を最大限に利用し、重複による非効率を廃した上で発展性の高い解析に専念できる環境を実現することを目的とする。

第三点は前の二点を利用した上で、繰り返し行われる解析をシステム化及び汎用化し、一括で解析を行うことを目的とする。例えば新規コンプリートゲノムが読み取られた時には基礎データを網羅的に取得するが、その解析ソフトウェアが汎用性に欠けていてすぐには新規ゲノムに対応できない場合がある。このような場合に汎用解析システムが既存の蓄積のもとに構築されれば、瞬時に基礎データを取得することが可能になる。また、研究内容により統合解析システムを既存研究に基づき構築すれば、これも多岐に渡り応用することが可能となる。

以上の三点の効率化を積極的に意識したバイオインフォマティックスシステムは前例がない。本研究はその課題に挑戦する *G-language Project* 及びそのコアソフトウェアである Prelude (*G-language System* core version 0.1 to 1.0 の開発コード） について議論する。

1.2 本研究の背景

G-language Project の具体的説明に入る前に、その意義及び課題を明確にする為に、他の同様な研究についてまとめる。

・ Bioperl

Bioperl プロジェクトはプログラミング言語 Perl の Bio::モジュール群として統合開発環境を実現することを試みる。[4] 多様な配列データベースフォーマットに対応し、ゲノム情報構造体とともに有用なメソッド群を提供し、さらにはヒトゲノムなど膨大な情報量を持つものに対しては仮想メモリを使用するなど、細部にいたるまで行き届いた入出力方式を持つ。また、一般的ツール（BLAST や FASTA など）を使用して解析するモジュールも含まれている。前項で述べた課題第一点をおおよそ Bioperl プロジェクトは満たすが､不規則な部分が多々ある Genbank データベースフォーマットに完全に対応しきれていない点、ゲノムのアノテーション情報のパーシング精度の問題、そして基礎入出力に関してはある程度充実しているが、具体的バイオインフォマティッ

クス解析に関する点が既存の一般的ツール以外は意識されていないなど、本研究の問題意識から見ると不十分な点が多い。類似プロジェクトに Biopython、Biojava、Biocorba、Bioruby などがある。

・ EMBOSS

EMBOSS はノルウェーの EMBnet により提供されている、ゲノム解析ソフトウェア集である。[5,6] 多様なコマンドラインで使用されることを想定したソフトウェアの集合として、簡易なスタンダードを設けたのがこの EMBOSS の実態であり、提供される解析手法の量と質は充実している。しかし、統合開発環境及び基礎プラットフォームは欠如しており、さらには本研究の問題意識の一つである統合解析システムという視点も存在しない。類似プロジェクトとして NCBI による NCBI SEALS があげられる。

・ Darwin

Darwin はスイスの ETH により提供されているゲノム解析環境で、独自のスクリプト言語を持ち関数化された解析を可能とする。5 年以上の歴史を持つプロジェクトであるが、柔軟性に欠ける配布ライセンスの問題、Bioperl などと比較して特に優れた解析手法がない点などから広い支持を得ているわけではない。[7]

以上のような類似研究が克服していない課題に挑戦する為、本年度から慶応大学先端生命科学研究所において *G-language Project* が発足した。現段階で開発はプロトタイプである Version 0.5 が完成しており、10 月から http://www.g-language.org/ を中心として、ネット上で公開する予定である。

2 プログラムアーキテクチャ

2.1 設計思想

汎用ゲノム解析環境 *G-language System* の設計において最重要視されることは、その汎用性に他ならない。不特定多数のゲノム解析手法及び内容が統一のシステムで開発され、再利用を前提としたシステム化が行われるのならば、システム設計もそれに伴い柔軟に対応できるものが望まれる。また、ここでは入力情報となるゲノムデータベースファイルの多様性や、既存の優れたオープンソースソフトウェアを幅広く利用できることが不可欠となる。

これらの汎用性という最重要課題を解決する方法として、*G-language System* のプログラムアーキテクチャを図 1 のように設計した。この設計はユーザインタフェース層、スクリプト・モジュール層、コア層の三層構造から成り、それぞれの層が完全に分離可能な自立システムとして構築される。コア層による各種入出力が統一フォーマットからなる構造体を創り出し、その構造体自体がスクリプト・モジュール層に受け渡されることで、コア層を介したデータは、スクリプト・モジュール層ではその形式を気にせず扱うことができる。また、スクリプト・モジュール層からの出力も統一されることで、ユーザインタフェース層では多様なプログラムによる解析結果をその形式の違いに煩わされることなく見ることができる。

三層構造をそれぞれ独立させることには上に述べたようにプログラム間通信の利点もあるが、特に *G-language System* において重要な意義を持つ。それは、状況に応じて *G-language System* の中で非常に簡単に層の入れ換えが可能な点である。全章で述べた通り、類似プロジェクトの中でも特に注目を浴びている Bioperl は言わば *G-language System* のコアに相当する機能を持つ。設計段階においてそれぞれの層を自立可能にすることで、例えば *G-language System* を Bioperl のコアを利用してスクリプト・モジュール層による解析を行うことが可能になる。これにより柔軟に優れた機能を外部から取り入れること、そして利用者がその好みによってシステムを組み合わせ可能になる。*G-language System* がバイオインフォマティックスの統合システムとして画期的な点はそのスクリプト・モジュール層の多様な解析手法の蓄積にあり、さらにそれが非常に汎用性を持っている点である。

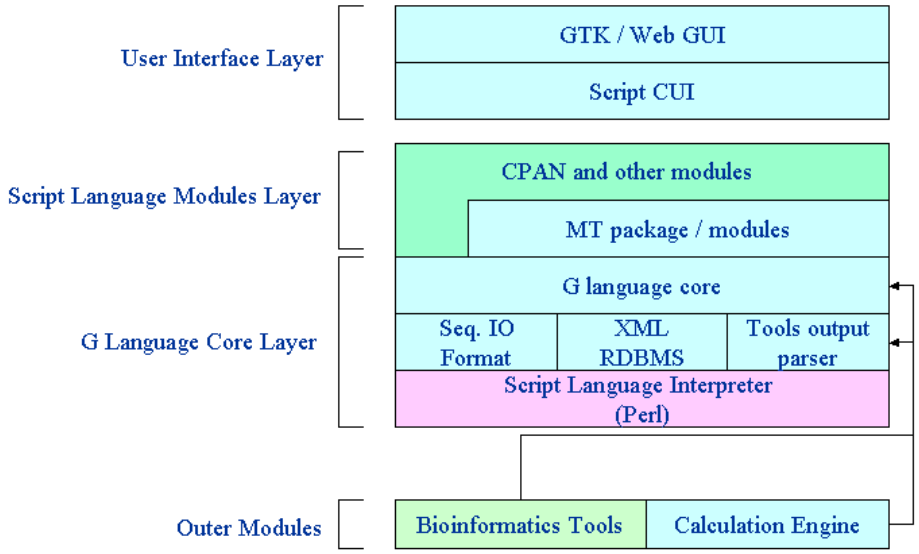

図１： *G-language System* プログラムアーキテクチャ

2.2 アーキテクチャ

G-language System の実体はスクリプト言語 Perl のモジュール群である。[8] Perl は Larry Wall によって開発され[9]、既にスクリプト言語として最も一般的に使われている。記述が非常に柔軟な点、そしてその歴史により数多くの有用モジュールが CPAN[10]などのインターネットアーカイブグループによって蓄積されている点が特徴的な言語である。開発環境として広く汎用性を持たせるにはプログラムの容易さ及び柔軟さ、そして既存モジュールが多いことが大きな利点となる。また、ゲノムデータベースは言わば文書であり、これを解析するには文字列処理を得意とするスクリプト言語が適する。[11] 以上のような理由から、コンパイル言語ではなくスクリプト言語を用いることに大きな利点がある。

ただし、計算効率の問題からコアのみコンパイル言語で構築すること、そしてスクリプト言語でも

Python や Ruby などの新しいより発展した言語を使用することがより望ましい場合も存在する。*G-language Project* としてもこれらの必要性は無視できないが、当面現在の開発者が Perl 以外の言語に不慣れである、という状況などを踏まえ、開発言語を Perl に統一した。

現在ゲノムデータベースフォーマットとして最も一般的なものが Genbank 形式のフラットファイルデータベースフォーマットである。[12] これ以外のフォーマットの入出力は、データベースフォーマット変換ツールである readseq をコアでサポートすることで対応した。また、データ量や検索速度などの観点から *G-language System* ではリレーショナルデータベースを標準で利用可能とし、これを CPAN の Pg モジュール及び Prelude コア の一部である gbk2pg でアクセスする方法をとる。これらファイル入出力を経て得られたデータを *G-language System* コアがプログラム内部で構造体として保持し、それがスクリプト・モジュール層で使用される。

3 *G-language System* の実装

3.1 Prelude

Prelude とは、*G-language System* コア層のうち、前章で述べたようにデータベース情報を構造体として保持する G クラス、リレーショナルデータベースと連動する gbk2pg クラス、そして統計解析言語である R 言語との通信を行う Rcmd クラスから成る。G クラスは *G-language System* の全システムのスーパークラスであり、スクリプト・モジュール層の MT package はこのクラスでライブラリー及びモジュールとしてロードされる。つまり、G クラスのインスタンスは自動的に G クラスを継承するとともに、ネイティブな関数として MT package の関数を利用することができる。これにより多彩な関数を MT package から提供することにより、*G-language System* という開発環境において Perl という言語でそのシステムを拡張しながら解析を行う、という擬似プログラム言語が構築される。この擬似プログラム言語環境は将来的にスクリプト型ユーザインタフェースにより、より高度なものに拡張される。

G-language System の汎用性を利用すれば、コア層は Bioperl で代用できることは前章で述べた。しかし、ここで Prelude を敢えて開発する意義は大きい。それは、Bioperl は統合システムとして設計されていない点、統計解析で有用な R 言語をサポートしていない点、そしてデータベース情報を持つ構造体が Genbank 形式のイレギュラーな書式に対応しきれていないため不十分である点などがあげられる。現在 Prelude はこれらの問題点を克服し、イントロンやエキソン、その他基本的な配列解析手法を多くネイティブでサポートし、さらにヒトゲノムのように巨大なデータベースフォーマットを扱えるようにメモリをファイルハンドルのポインタで操作できるなど、バイオインフォマティックス統合システムのコアの役割を十分に果たせる機能を持っている。G クラスの詳細な機能については巻末にその Perldoc ドキュメンテーションを添付する。

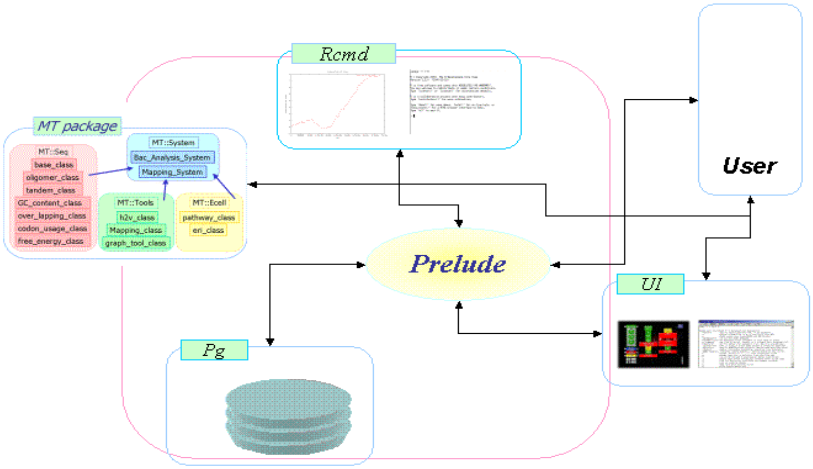

図２：*G-language System* における Prelude コア

3.2 MT package

MT package は様々なバイオインフォマティックス解析手法の集合であり、内部で複数の関連した機能毎にまとめられたモジュールとして構成されている。ただし、前章で述べたようにこれは Prelude の派生クラス及びインスタンスからは、スーパークラスとして G クラスからライブラリとしてロードされるので、実際にはユーザにはネイティブな関数として提供される。

MT package 内の関数はそれらの間でさらに主従関係を持つ。すなわち、最小単位の関数は例えば GC skew を計算するものなどであり、これら関数を複数まとめバッチ処理を行う関数が作られる。現在 *G-language System* に実装されているバクテリア網羅解析システムは例えばこのような MT package 内の関数を統合するものであり、それ自体がまた MT package の関数として提供されている。統合的なシステムソフトウェアはこのような汎用的スクリプト・モジュール層の設計の上に成る。

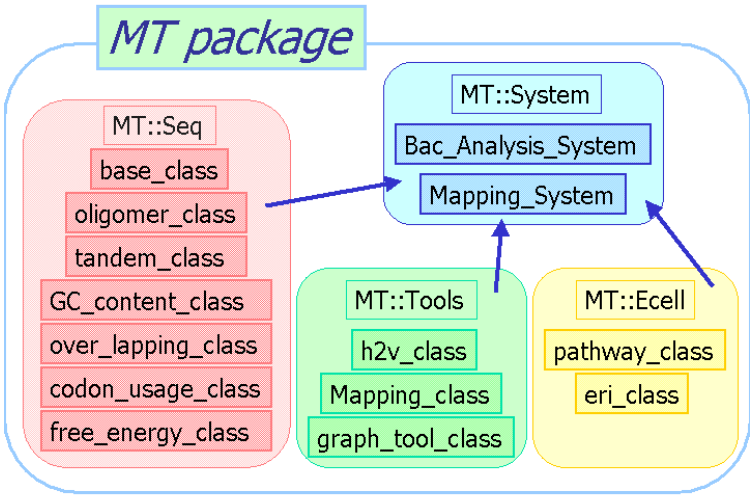

図３： MT package 概要

3.3　システム実装例：　MT::System::mutation_finder

ここで具体例を挙げて *G-language System* がどのように機能するかを紹介する。異なる二つのバクテリアゲノムデータベースから突然変異情報を探るシステム、MT::System::mutation_finder メソッドの流れを図式化すると図４のようになる。つまり、ゲノムデータから遺伝子毎に BLAST による相同性解析を行い、その結果をもとに相同遺伝子の変異を調べ、その変異の部位及びそれが及ぼす翻訳や二次構造への影響を解析し出力する。

まず、MT::System クラスの mutation_finder メソッドが *G-language System* 上で呼び出され、あらかじめ Prelude 起動時に読み込まれたゲノムデータ構造体が引き渡される。このデータをもとに、mutation_finder メソッドが内部で別の MT メソッド群を呼び出し個々の解析を行い出力を返す。個々のメソッドとは、例えば MT::uni-all_blaster、MT::bi-bi_clustalwer、MT::RNAfolder などで、これらはそれぞれさらに内部から外部プログラムである BLAST、ClustalW、RNAfold などを呼び出す。つまり、ここでは Prelude 構造体に基づく汎用メソッド群としての MT package が多重構造を成す事によって複雑な処理を統一された環境のもとに実現することで開発コストが低減し、さらに個別メソッドの再利用性が増す。すなわち、これは最下層に位置する MT メソッドは異なるシステムの構成部品として利用される可能性を意味する。また、MT::System の入出力は Prelude 構造体及び *G-language System* 共通の構造体であることから、異なるフォーマットのファイルに煩わされることなく実行できること、そして汎用的な解析システムとして有効であることを意味する。

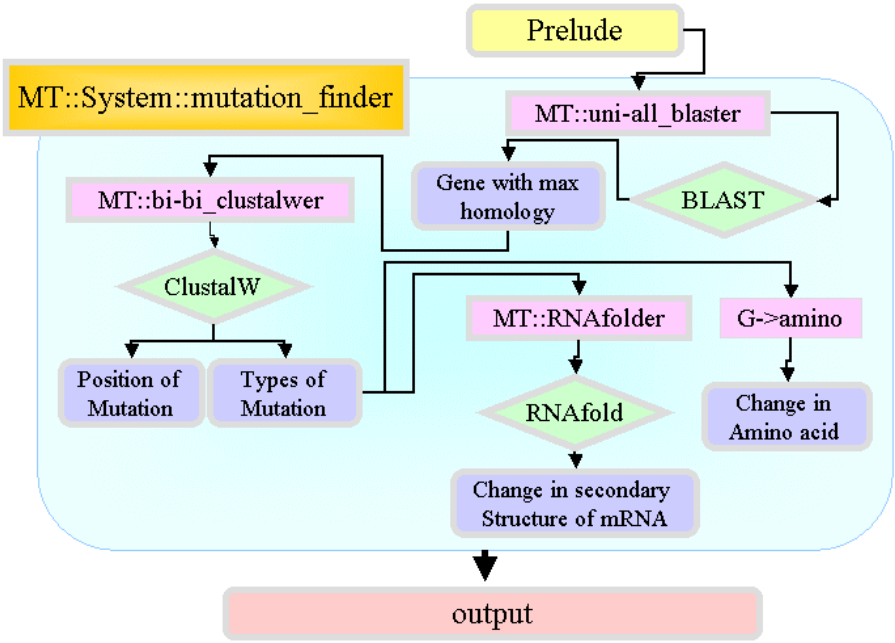

図４：　MT::System::mutation_finder プログラム図

4 議論

4.1 開発予定

現在 *G-language System* は version 0.5 の段階まで開発が進んでいるが、10 月のベータリリースに向けて解決すべき課題は多い。それらは、

- ドキュメンテーションの整備
- 使いやすい UI の開発
- 網羅的な動作テスト
- 既存 MT package からのシステム構築
- MT::System クラスの充実

などである。また、12 月の正式リリースに向けてはより多くの基礎解析及びツールの充実を図らなければならない。

4.2 差別化

ポスト＝ゲノム時代が幕を開けたと同時に、多くの類似研究が台頭しつつある。この中で、*G-language Project* はそれら類似研究との差別化を図り、その充実化に努めなければならない。ここで重要だと思われる点は、一章で述べたバイオインフォマティックスが直面する 3 つの課題をすべて解決すること、ミドルエンドではなくローエンドとハイエンドのユーザを共に満足させるシステムを作ること、そして、パブリックドメインから多くの参加を募ること、であると思われる。

Prelude と MT package という二つの非依存型のソフトウェアを統合することで、*G-language System* は例を見ない汎用解析環境及びシステムを構築するが、この点をより明確にするためにも、より多くの、そして充実した MT package 及び解析システムを開発する必要がある。また、現在の環境ではハイエンドユーザ以外は扱いにくいと思われるので、使いやすい UI によってローエンドユーザもターゲットとする。さらに、これらによって構築されたシステムを GPL によってオープンソース・コピーレフトの概念のもとにパブリックドメインに公開し、幅広く参加を募る。こうして、バイオインフォマティックスにおける共通の汎用解析環境を提供することを目的としたい。

5 謝辞

G-language System の開発は総合政策学部の森航哉氏を始め *G-language Project* のメンバの助力を頂いた。また、本研究においては環境情報学部の中山洋一専任講師、理化学研究所の斎藤輪太郎氏、政策メディア研究科の高橋恒一氏、同三由文彦氏に多くの適切なアドバイスをいただいた。この場をかりて感謝の意を表したい。

さらに、本研究の進行に対してあらゆる面から多大なバックアップをしていただいた冨田勝教授に特に感謝したい。

Bibliography

[1] Tully, J.G., Taylor-Robinson, D., Rose, D.L., Cole, R.M., and Bove, J.M. "Mycoplasma genitalium, a new species from the human urogenital tract." Int. J. Syst. Bacteriol. (1983) 33:387-396.

[2] Cynthia Gibas and Per Jambeck, “Developing Bioinformatics Computer Skills” O'Reilly & Associates. (2001)

[3] Richard Stallman “Why Software Should be Free” available at http:// www.fsf.org/philosophy/shouldbefree.html (1992)

[4] Bioperl project “Bioperl” available at http://bio.perl.org/ (2001)

[5] Rice P, Longden I, Bleasby A. “EMBOSS: the European Molecular Biology Open Software Suite.” Trends Genet. 2000 Jun;16(6):276-7.

[6] Letondal C. “EMBOSS: the European Molecular Biology Open Software Suite.” Trends Genet. 2000 Jun;16(6):276-7.

[7] Gonnet GH, Hallett MT, Korostensky C, Bernardin L. “Darwin v. 2.0: an interpreted computer language for the biosciences.” Bioinformatics. 2000 Feb;16(2):101-3.

[8] Joseph N. Hall, Randal L. Schwartz “Effective Perl Programming Writing Better Programs with Perl”, Addison-Wesley (1998)

[9] Larry Wall, Tom Christiansen, and Randal L. Schwartz with Stephen Potter “Programming Perl Second Edition”, O'Reilly & Associates (1998)

[10] Comprehensive Perl Archive Network available at http://www.cpan.org/ (2001)

[11] Lincoln Stein “How Perl Saved the Human Genome Project” The Perl Journal, Readible Publications (1996) available at http://bio.perl.org/GetStarted/tpj_ls_bio.html

[12] Benson DA, Boguski MS, Lipman DJ, Ostell J, Ouellette BF, Rapp BA, Wheeler DL. “GenBank.” Nucleic Acids Res. 1999 Jan 1;27(1):12-7.