遺伝子解析支援データベースの構築

A new DNA database with flexible and quick homology search

清水　佳奈 [1)]　　秋岡　明香 [2)]
Kana SHIMIZU　Sayaka AKIOKA

1) 早稲田大学 大学院 理工学研究科 情報科学専攻 （〒162-0044 東京都新宿区喜久井町 17 理工学総合研究センター第二研究棟 301　E-mail: kana@muraoka.info.waseda.ac.jp ）

2) 早稲田大学 大学院 理工学研究科 情報科学専攻 （〒162-0044 東京都新宿区喜久井町 17 理工学総合研究センター第二研究棟 301　E-mail: akioka@muraoka.info.waseda.ac.jp ）

ABSTRACT. Homology search is one of the most important service which a DNA database provides. And each research worker searches homologous sequences for his or her own purpose, because there are many different fields of study when we analyze DNA sequences. Therefore flexible search is required. However, huge amount of DNA data and increases of data entries make it difficult to manage a DNA database, and researchers can not search homologous sequence related with annotation information such as gene name, detail species, locus, etc if they use existing DNA databases. Also, speedy search is required, because we have much data which should be analyzed. However, when we search many sequences, it costs much time, if we use a tool which need more processing power to calculate homology. In this paper, we propose a frame work for a new database which provides more flexible and quick homology search. We use GRID computing to calculate homology of all DNA sequences to All, and store all calculation results in a database. Then the database provides users with quick homology search with annotation information.

1　背景

現在のゲノムデータベースでは様々なサービスが提供されているが, その中でも特に重要なサービスが相同性検索である。大多数のゲノム研究者は, 研究のファーストステップとして, 自分がターゲットとする遺伝子配列の類似配列をデータベース上から検索し, それを出発点として様々な解析を行っていく。相同性検索のツールとしては一般的に,BLAST [1],FASTA [2],PSI-BLAST [1] 等が用意され, ユーザは CGI 等によって問い合わせたい配列の類似配列情報を得ることができる。

一口にゲノムの解析といっても様々な分野があり, 各研究者はそれぞれ異なる目的を持って相同性検索を行っている。例えば, 単一の生物種, 単一の遺伝子に着目して解析を行っている研究者もいれば, 多様な生物を扱い, 種間の関係に着目している研究者もいる。そのため検索の際には, 各研究者のアイデアや目的に合わせて, 検索の対象とする生物種, 遺伝子名, データベース等の付加的な情報と関連付けて柔軟に検索を行うことにより, 遺伝子情報をより多角的な方面から解析することができる。

しかしながら, ゲノムのデータベースは莫大な量であること, 日々増え続けていることがその管理を困難にし, 現在公共のデータベースにおける相同性検索サービスでは, 詳細に範囲を指定したり, 付加的な情報とあわせて検索を行うことができない。（データベース側が決めた大まかな分類にのっとった検索のみが可能。）

また, どのような研究であっても効率的に解析を進めるには大量のデータを一度に短時間で検索することが理想的であるが, ツールによっては計算コストが大きく, たくさんの配列の検索結果を得ようとするとかなり時間がかかってしまうことがある。

情報生物学の分野において以上のような背景がある一方で, コンピュータサイエンスの分野では GRID コンピューティングが注目を集めている。GRID には様々な形態, 思想があるが, インターネット上の遊休計算機を利用した成功例として SETI [5],folding@ [6] が挙げられる。両者はともにプロジェクトに賛同する個人が所有する PC を GRID 上のノードとして利用している。後者はタンパク質の折りたたみ構造の解析に関するプロジェクトであり, 情報生物学の分野において GRID が大きな役割を果たしていることを示している。

2　目的

本プロジェクトでは, 付加的な情報（アノテーション情報）と関連させて, 大量のデータを高速に相同性検索することが可能なゲノムデータベースの構築を目的とする。

高速な検索を実現するため, 現存する全遺伝子対全遺伝子の類似度を計算してデータベース化する。計算方法によっては類似度の算出にかなりのコストがかかることをふまえ, 計算には GRID コンピューティングを用いる。

具体的な目標としては (1)GRID を用いて類似度の計算を分散させ,(2) 計算結果を回収してデータベースに登録し,(3) 使いやすいユーザインターフェースによる相同性検索を提供するための, フレームワークの構築を行う。

3　プロジェクトの概要

本節ではプロジェクトの概要について述べる。図 1 は提案するシステムのイメージ図である。提案するシステムでは, 定期的に公共のゲノムデータベースから最新の遺伝子データを取得し, それを管理する。そして取得したデータを GRID 上のノードの要求に応じて切り分けて分散させ, 各ノード上で類似度の計算を行う。計算が終わったものから結果をチェック・回収し, データベース上で管理する。このようにして構築したデータベースを, 遺伝子解析支援

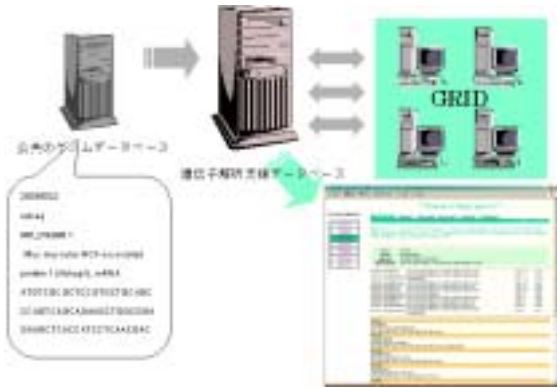

図 1: 全体のイメージ図

データベースとして公開し, ユーザに対して自由な検索を提供する。

提案するシステムのサーバーサイドの特徴を以下にまとめる。

1) GRID を用いて計算を行う。
2) 拡張性が高い（クライアントソフトウェアに組み込むモジュールを入れ替えるだけで, そのモジュールが算出する遺伝子間情報を全データ×全データについて得ることができる。）

提案するシステムのユーザーインターフェースーサイドの特徴を以下にまとめる。

1) アクセッションナンバーを入力するだけで類似配列を検索できる。
2) 一度に複数の配列に対して類似配列を検索できる。
3) 生物種, データベースの種類, キーワード（遺伝子の名前, 機能等）と類似度を関連付けて検索できる。
4) 生物種ごと, アクセッションナンバーごとに分類されて表示される。

4 システムの概要

本節では, 本プロジェクトで提案するシステムについて述べる。まずは, 処理シーケンスの概要を以下に列挙する。なお, サーバーサイドの処理は図 2 に示す。

1) 公共のゲノムデータベース（DBBJ[4]）よりデータをダウンロードする。
2) ダウンロードしたデータを, 遺伝子解析支援データベースサーバ（以下サーバーと略す。）に格納する。
3) GRID 上のクライアントソフトウェア（以下クライアントと略す。）から, 計算要求を受けるとスケジューラが未計算データの ID をクライアントに対して通知する。
4) クライアントは,ID に対応するデータを受け取り, 計算を行う。
5) クライアントは, 計算結果がそろった時点でサーバーに計算結果を送信する。
6) サーバーは受け取った計算結果が正しいかどうかチェックし, 正しいと判断された場合のみ, 計算結果を格納する。
7) サービスを受けたいユーザは,web ブラウザでサーバーにアクセスし, キーワードや生物種などを自由に決めて, 遺伝子情報の検索を行う。

今回のプロジェクトでは, 類似度計算モジュールとして NCBI[3] で提供されている,blast[1] を利用した。

4.1 計算データの分割について

本プロジェクトの最も重要な課題は, 現存する全ての遺伝子データ間の類似度を計算する機構を構築することである。類似度の計算量は莫大であるため, 本プロジェクトでは,GRID を用いて類似度の計算を行う。その際には,「どのデータが配布済であるか」,「どのデータの計算結果が返却済か」などの情報を保持するテーブルが必要となる。

本プロジェクトが対象とする遺伝子データは 100 万のオーダであり, 仮に, 全遺伝子データ×全遺伝子データのテーブルを保持しようとすると, それだけでハードディスクがあふれてしまう。そのため, 本システムでは配列データをブロック単位で管理し, テーブルのサイズを縮小することにした。動的に, テーブルを変更する方法も考えられるが, それについてはまだ実験している段階である。

それでは, ブロックについて説明する。図 3 は 1 ブロックの 1 辺のサイズを 30 とした場合のデータの割り振りである。データの ID は, 追加された順にシリアル番号を与えられる。

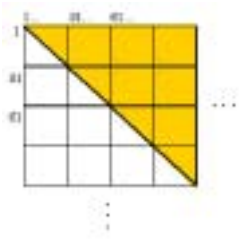

図 3: データ分割 1

最初のブロックは図 4 のオレンジ色で示した部分であり, このブロックに含まれるデータは ID1〜ID30,ID31〜ID60 までである。次のブロックは図 5 のオレンジ色で示した部分であり, このブロックに含まれるデータは ID1〜ID30,ID61〜ID90 までである。

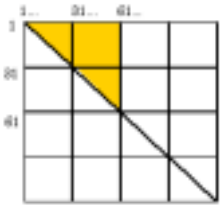

図 4: データ分割 2

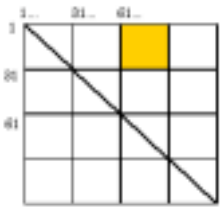

図 5: データ分割 3

さらに, 次のブロックは図 6 のオレンジ色で示した部分であり, このブロックに含まれるデータは ID31〜ID60,ID61〜ID90 までである。

このように, ブロックを定義することで, 新しいシークエンスデータが随時追加されても同様のフレームワークでブロックを生成することができる。

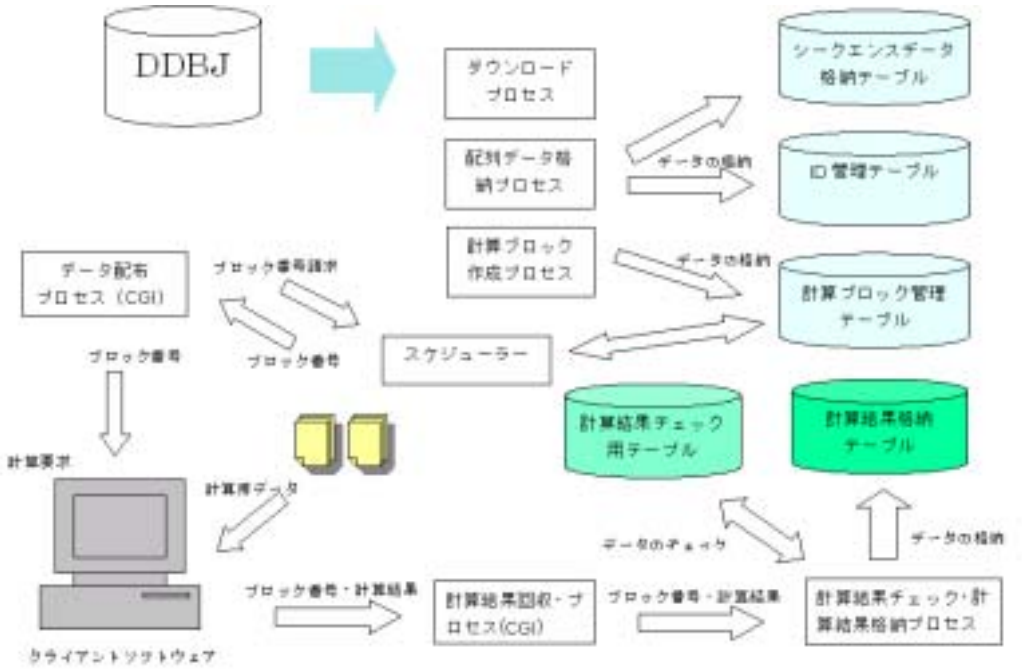

図 2: サーバーサイドの処理の流れ

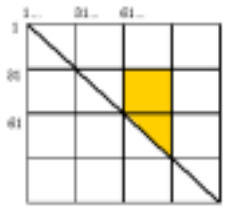

図 6: データ分割 4

4.2 各モジュールの機能

各モジュールの機能について, 以下に説明する。

1) ID 管理テーブル
シークエンス固有に割り振った ID とアクセッションナンバーを対応付けて, 管理するテーブル
2) 計算ブロック管理テーブル
各計算ブロックについて, 計算結果が終わったか, また最後に配布されてからどのくらいの時間が経過しているかなどを保持するテーブル
3) 計算結果チェック用テーブル
計算結果が正しいかどうかチェックするために初回の計算結果を仮に保持しておくテーブル
4) 計算結果格納テーブル
計算結果を格納するテーブル
5) ダウンロードプロセス
DDBJ に定期的にアクセスし, 新規データをダウンロードする。
6) 配列データ格納プロセス
テキストデータから, 必要な情報を抜き出して配列データを格納する。
新規データに新しい ID を振り分け, アクセッションナンバーと対応付ける。
7) ブロック作成プロセス
本節の冒頭で述べたような決まりでブロックを定義して新規ブロック ID を発行し, ブロックデータ（何番目と何番目の配列 ID から始まるか）を計算ブロック管理テーブルに格納する。
8) ブロックデータ作成プロセス
ブロックの 1 辺の単位ごとに, シークエンスデータと ID データのみを抜き出してブロックデータを作成する。
9) スケジューラー
ブロック管理テーブルの情報をふまえて, 計算すべきブロックを選択して伝える。
長時間計算結果が返ってこないブロックの再配布などもする。
10) データ配布プロセス（CGI）
クライアントから計算要求を受け付けてスケジューラーに伝える。
スケジューラーから計算すべきブロック ID を受け取り, クライアントに伝える。
11) 計算結果回収プロセス
5000 番ポートでクライアントから計算結果を受け取り, 計算結果格納プロセスに渡す。
12) 計算結果格納プロセス
計算結果のチェックを行って, 正しい計算結果のみを計算結果格納テーブルに格納する。
13) クライアントソフトウェア
http でデータをダウンロードした後に,bl2seq 用に分割し,bl2seq（NCBI[3] が配布しているバイナリ）で計算を行う。
計算結果から, 必要な部分を抜き出して,http でサーバーに返送する。
14) ユーザーインターフェース
CGI によって, キーワードや生物種などを自由に決めて, 遺伝子情報の検索を行うことができる。

5 実装

前節までに述べたシステムを, 実装した。

サーバーサイドの OS は solaris を使用した。開発言語には,C,perl を使用し, データベースソフトウェアとしては

PostgreSQL[7] を利用した。クライアントソフトウェアは windows 端末を対象とし,C#で開発を行った。また, ユーザーインターフェースは手軽に使えることを重視し,CGI で作成した。右側のフレームにユーザが検索を要求したアクセッションナンバーのリストを表示し, 左側のフレームには検索結果をタブ形式で生物種ごとに分類した。図 7 に検索結果の画面を示す。

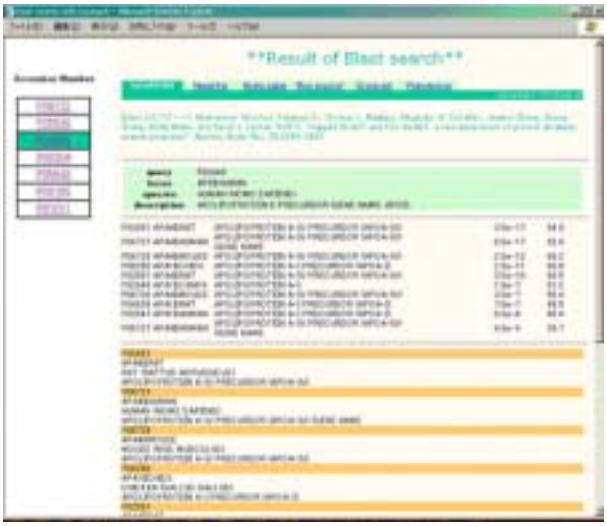

図 7: 検索結果の例

6 実験

実装したシステムを用いて、テストデータベースを構築する実験を行った。全配列間の類似度を計算し終えるまでには約 99 時間かかった。

また、構築したデータベースに対してアクセッションナンバーを指定しての相同性検索を 100 回試行したところ、1 回の平均検索時間は約 1.01 秒であった。

6.1 使用データ

データ名称	swiss-prot[8] (sprot09)
配列総数	8700
ブロック総数	4,1905
計算結果総数	222,237

6.2 実験環境

類似度の計算には、以下の１５台の計算機が参加した。

CPU	メモリ	台数
pentium4 2.4Ghz	1GB	2
Celeron 797Mhz	256MB	7
Celeron 700Mhz	128MB	1
Celeron 700Mhz	64MB	5

7 考察

実装したシステムでテストをした結果,GRID 上のノードを増設するほど全体としての計算速度が向上することが分かり, 本システムが有効に働いていることが分かった。クライアントソフトウェアの計算のさせ方を工夫することによってパフォーマンスの向上が見込めるため, 毎日のアップデートに対応するには 10000 万台程度のノードがあれば対応できると考えている。

また, ユーザーインターフェースの検索速度も, 実用上問題の無いパフォーマンスを実現できた。今後, データベースを拡張して行ったときに検索速度が遅くなることが懸念されるが, データベース自体を並列化するデータグリッド構造にすることによって, それを回避できると考えている。

8 今後の課題

今回のプロジェクトでは, 遺伝子解析支援データベースを構築するフレームワークの基礎的な部分を完成させることができた。しかし, 完全なシステムにするためには, データ配布のスケジューリング, 計算結果のチェックなどについて改良する必要がある。また, ゲノムのデータ量は膨大であるため, 全遺伝子対全遺伝子の計算結果情報の格納にはより大きなデータベースが必要となる。検索効率などを考えると, データグリッドなどを用いることが必須になるであろう。さらに, クライアントソフトウェアに,PSI-Blast などの, よりマシンパワーの必要とするモジュールを組み込んでデータベースを構築することで, より価値のある情報を提供することも考えられる。

9 まとめ

本プロジェクトでは, 以下にあげる項目を開発することにより, 目的とする遺伝子解析支援データベースの構築を行った。

1) サーバーサイドの開発遺伝子配列データを取得・管理し,GRID 上の計算機に配布し, 計算結果をチェックした上でデータベースに格納するシステムを開発した。
2) クライアントサイドの開発サーバーから受け取ったデータに対して計算を行い, 結果をサーバに返送するソフトウェアを開発した。
3) ユーザインターフェースの開発計算結果をアクセッションナンバー, 生物種, キーワードなどによって検索し, アクセッションナンバー及び生物種によって分類された情報をユーザに提供するインターフェースを開発した。
4) 実際にデータを用いてのテスト作成したシステムを用いて, データベースを構築し, マシンスペックごとの計算効率, ユーザインターフェースを用いての検索時間などの測定を行った。

また, 以上に述べた開発によって遺伝子解析支援データベースのフレームワークを作り上げることができた。そのため, サーバーマシンのスペックを上げることによって, より大規模な計算を行うことが可能である。また, クライアントソフトウェアに組み込むモジュールを入れ替えるだけで, そのモジュールを走らせることで得られる情報を全遺伝子対全遺伝子について得ることができる。

参考文献

[1] Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25:3389-3402.1997,Stephen F. Altschul
[2] W. R. Pearson and D. J. Lipman. "Improved Tools for Biological Sequence Comparisons". Proceedings of National Academy of Sciences (PNAS), 85, 2444-2448, 1988.
[3] http://www.ncbi.nlm.nih.gov/
[4] http://www.ddbj.nig.ac.jp/Welcome-j.html
[5] http://setiathome.ssl.berkeley.edu/
[6] http://folding.stanford.edu/
[7] http://www.postgresql.org/
[8] http://www.ebi.ac.uk/swissprot/