Mining the NCBI Influenza Sequence Database: adaptive grouping of BLAST results using precalculated neighbor indexing

October 30, 2009

Leonid Zaslavsky, Tatiana Tatusova


Abstract

The Influenza Virus Resource and other Virus Variation Resources at NCBI provide enhanced visualization web tools for exploratory analysis for influenza sequence data. Despite the improvements in data analysis, the initial data retrieval remains unsophisticated, frequently producing huge and imbalanced datasets due to the large number of identical and nearly-identical sequences in the database.

We propose a data mining algorithm to organize reported sequences into groups based on their relatedness to the query sequence and to each other. The algorithm uses BLAST to find database sequences related to the query. Neighbor lists precalculated from pairwise BLAST alignments between database sequences are used to organize results in groups of nearly-identical and strongly related sequences. We propose to use a non-symmetric dissimilarity measure well crafted for dealing with sequences of different length (fragments).

A balanced and representative data set produced by this tool can be used for further analysis, i.e. multiple sequence alignment and phylogenetic trees. The algorithm is implemented for protein coding sequences and is being integrated with the NCBI Influenza Virus Resource.

Introduction

The amount of influenza sequence data is increasing rapidly due to the collaborative genome sequencing efforts[1], [2]. In support of this work, The Influenza Virus Resource[3] has been created at NCBI and is available for public access. This and other Virus Variation Resources at NCBI provide enhanced visualization web tools for exploratory analysis[4], [5], [6] allowing the representation of phylogenetic trees in an aggregated form with a mechanism for refinement.

Despite the improvements in data analysis, the initial data retrieval remains straightforward in many cases. For example, a database query to NCBI Influenza database is formed as a simple combination of metadata (type of the virus, serotype, segment/proteins, time interval for the date of isolation, etc.)[3] and returns all the sequences that satisfy the query. With the increased number of sequences, it becomes more difficult to operate in this manner: a priori knowledge and experience are required to confirm that all the relevant sequences are included. The dataset extracted in response to the query may be huge and imbalanced due to the large number of identical and nearly-identical sequences in the database (that may differ in metadata though and could not be simply eliminated at the database level). Manual selection of a representative subset is time consuming and difficult. This may create a bottleneck during ongoing pandemic (see[7] for related analysis and references).

In this paper, we describe an algorithm allowing to select the sequences with various degrees of similarity to the query and group the results based on their relatedness to each other. The algorithm uses BLAST scores to find and rate the database sequences related to the query. Pre-calculated neighborhood lists based on pair-wise BLAST hits between the database sequences allow to report the results in groups of identical and nearly identical sequences.

The algorithm is implemented for protein coding sequences and is being integrated with the NCBI Influenza Virus Resource.

Method

Neighbor-relationships between sequences

The standard nucleotide and protein sequence scoring models satisfy conditions (1) – (11) in[8], and the Smith-Waterman local alignment scores satisfy the inequality

\[ S^L(x,y) + S^L(y,z) \leq S^L(x,z) + S^L(y,z), \]

PLOS Currents Influenza
Define dissimilarity measure between sequences $x$ and $y$:

$$
\delta^L(x,y) = 1 - \frac{S^L(x,y)}{S^L(y,y)} \quad (2)
$$

Sequence $y$ is an $\varepsilon$-neighbor to sequence $x$ if

$$
\delta^L(x,y) \leq \varepsilon \quad (3)
$$

where $0 < \varepsilon \ll 1$. The maximum value of $\delta^L(x,y)$ is 1 and it is realized on two unrelated sequences. For identical sequences the value is 0. The dissimilarity measure in (2) is non-symmetric:

$$
\delta^L(x,y) \neq \delta^L(y,x).
$$

That allows a short sequence to be an $\varepsilon$-neighbor to a long sequence, but not vice versa.

In practice, the highest BLAST bit score between two sequences (“BLAST score”), $S^{BLAST}(x,y)$, will be used. BLAST dissimilarity measure

$$
\delta^{BLAST}(x,y) = 1 - \frac{S^{BLAST}(x,y)}{S^{BLAST}(y,y)} \quad (4)
$$

will be used instead of (2). If BLAST does not find any similarity between $x$ and $y$, $\delta^{BLAST}(x,y) = 1$.

Since we are interested in close relationships only and do not consider distant relatives, using fast BLAST algorithm rather than the Smith-Waterman alignment seems reasonable in this context.

Creating display groups

A complete list of all reported BLAST hits to a given query sequence may be difficult and sometimes impractical to use due to the swarms of identical and nearly-identical sequences in the search set. The resulting list may be either too long or incomplete depending on the selected threshold. It will be useful to identify redundant and nearly redundant sequences and produce a near-redundant group report rather than a raw list of all BLAST hits. In our calculations, both relatedness to the query sequence and pairwise similarity of the database sequences are taken into account.

An important requirement to the algorithm is to allow either query or subject sequence to be incomplete. For example, comparing a coding sequence of HA1 domain of hemagglutinin with complete dataset[11] .

Suppose that the subject sequences $x_0, x_1, ... , x_N$ are ordered by score to the query sequence $Q$:

$$
S(x_0,Q) \geq S(x_1,Q) \geq ... \geq S(x_N,Q).
$$

**Iterations.** During the steps of the algorithm we select the sequences from the ordered list $\Theta$. Initially, set $\Theta = \{1, 2, ..., N\}$. At each step we select an anchor sequence $x_k$. The anchor sequence and all its $\varepsilon$-neighbors for group $\Lambda_k$ are removed from the list $\Theta$. The procedure is applied until the list $\Theta$ becomes empty.

**Selecting an anchor sequence.** An anchor sequence is selected as follows. A top-score sequence $x_i$ is considered first (e.g., $i$ is the first in the list $\Theta$), and the anchor sequence $x_k$ is selected among the sequences in $\Theta$ if it is an $\varepsilon$-neighbor to $x_i$. Specifically, all sequences $x_k$ in the list $\Theta$ are selected if $x_i$ is an $\varepsilon$-neighbor and satisfy the criterion

$$
S(x_k,Q) \geq S(x_i,Q) - \gamma S(Q,Q), \quad (5)
$$

where $0 < \gamma \leq \varepsilon$, are considered. The sequence $x_k$ of a maximum length is selected as an anchor (see Figure 1).
Forming a group. The anchor sequence \( x_k \) is selected, then group \( \Lambda_k \) is built: all \( \epsilon \)-neighbors to sequence \( x_k \) including sequence \( x_k \) itself are placed in group \( \Lambda_k \) and removed from the list \( \Theta \). The iterations are performed until the list \( \Theta \) is empty.

**Algorithm. Creating display groups.**

<table>
<thead>
<tr>
<th>Sort reported sequences by score:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S(x_0, Q) \geq S(x_1, Q) \geq \cdots \geq S(x_N, Q) ).</td>
</tr>
</tbody>
</table>

Set \( \Theta = \{1, 2, \ldots, N\} \).

While (\( \Theta \neq \emptyset \))

1. Take the first element \( i \) in the list \( \Theta \) and consider the list
   \[ \Pi_i = \{ k' \in \Theta \mid \delta(x_{k'}, x_i) \leq \epsilon \ \& \ S(x_{k'}, Q) \geq S(x_i, Q) - \gamma S(Q, Q) \} \].

2. Find the anchor sequence
   \[ x_k = \arg \max \{ l(x_{k'}) \mid x_{k'} \in \Pi_i \} \].

3. Create group
   \[ \Lambda_k = \{ x_j \in \Theta \mid \delta(x_j, x_k) \leq \epsilon \} \].

4. Reset \( \Theta \leftarrow \Theta \setminus \Lambda_k \).

**Regularization.** Taking the longest possible sequence \( x_k \) as an anchor instead of sequence \( x_j \) allows to create big and stable groups around anchors and avoid fracturing.

**Filtering low-score and low-overlap sequences.** After the groups \( \Lambda_k \) are created, the members of the group that have low-overlap with the query sequence or have much lower score to the query sequence than the anchor, could be filtered out.

**Indexing.** Steps (i) and (iii) of the Algorithm operate with the sets of \( \epsilon \)-neighbors for a sequence. These lists are pre-calculated for the database for preselected values of \( \epsilon \). As a result, only BLAST scores to the query sequence are calculated on-the-fly.

**Showing display groups**

Each group \( \Lambda_k \) can be displayed at various levels of detail: an anchor sequence and a number of sequences in the group (default) with or without aggregated metadata; groups of identical sequences; and individual sequences. In an aggregated form a group of sequences is shown as a row in the table containing aggregated metadata. For Influenza sequences, it may be score range, identity range, serotype(s), length range, country/countries.

**Results**

To demonstrate the work of the algorithm, we performed a search for the protein coding sequence of hemagglutinin [Influenza A virus (A/New York/3223/2009(H1N1))], coding for the protein with accession number ACR40288 (all data as of September 20, 2009). The request interface is shown in Figure 2 (Protein accession number[3][12] or protein coding sequence in either text or FASTA formats [13], [14] can be entered).
The results are shown in Figure 3. First group contains 943 near-redundant H1N1 2009 hemagglutinin protein coding sequences with identity to the query sequence varying from 100% to 99%. The score to the query sequence is varying from 3063 to 1269 due to the sequence length variation from 1701 – 708. While complete sequences in the group contain entire HA coding sequences, partial sequences cover coding sequences of either HA1 or HA2 regions (low-coverage filtering was not turn on in this test).

Two levels of resolution are shown in Figure 2: table

![Fig. 2: Query Interface](image)

Fig. 2: Query Interface

![Fig. 3: Displaying the search results.](image)

Fig. 3: Displaying the search results.

**Discussion**

The algorithm presented allows the adaptive creation of display groups, taking into account both relatedness of the reported sequences to the query and their similarity to each other. The groups are built in an efficient manner using pre-calculated neighbors for each sequences. Moreover, additional filtering can be performed, allowing exclusion of members of each group that are less relevant to the query than top members. A balanced and representative set of sequences selected using this tool can be passed to the analysis tools within the NCBI Influenza Virus Resource for calculating multiple sequence alignment and trees[3]. We plan to extend the technology to processing other viruses in the NCBI Virus Variation Resources[4].

The developed technology is based on the non-symmetric definition of dissimilarity in (2) and (3). Measuring dissimilarity in this way is adequate when the absence of the data is considered a probable match (a short sequence is very close to a longer one over the whole length of the former). In some situations, such assumption would not be valid, and a more strict similarity measure, such as the alignment metric (1) should be used[8].


**Acknowledgements**

The authors are thankful to David J. Lipman, Stephen Altschul, Yiming Bao, James R. Brister, Stacy Ciufo, William Klimke, Tom Madden, Kathleen O’Neill and Sergey Resenchuk for productive discussions. We appreciate work of Sergey Resenchuk on supporting a BLAST database for influenza virus protein coding sequences.

**Funding Information**

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

**Competing interests**

The authors have declared that no competing interests exist.

**References**


