Multiresolution Approaches to Representation and Visualization of Large Influenza Virus Sequence Datasets

Leonid Zaslavsky
National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Bethesda, MD 20894
zaslavsk@ncbi.nlm.nih.gov

Yiming Bao
National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Bethesda, MD 20894
bao@ncbi.nlm.nih.gov

Tatiana A. Tatusova
National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Bethesda, MD 20894
tatiana@ncbi.nlm.nih.gov

Abstract

Rapid growth of the amount of genome sequence data requires enhancing exploratory analysis tools, with analysis being performed in a fast and robust manner. Users need data representations serving different purposes: from seeing overall structure and data coverage to evolutionary processes during a particular season. Our approach to the problem is in constructing hierarchies of data representations, and providing users with representations adaptable to specific goals. It can be done efficiently because the structure of a typical influenza dataset is characterized by low estimated values of the Kolmogorov (box) dimension. Multiscale methodologies allow interactive visual representation of the dataset and accelerate computations by importance sampling.

Our tree visualization approach is based on a subtree aggregation with subscale resolution. It allows interactive refinements and coarsening of subtree views. For importance sampling large influenza datasets, we construct sets of well-scattered points (ϵ-nets). While a tree build for a global sample provides a coarse-level representation of the whole dataset, it can be complemented by trees showing more details in chosen areas. To reflect both global dataset structure and local details correctly, we perform local refinement gradually, using a multiscale hierarchy of ϵ-nets.

Our hierarchical representations allow fast metadata searching.

1. Introduction

The number of influenza virus sequences available in public databases is rapidly growing due to collaborative genome sequencing efforts by the National Institute of Allergy and Infectional Diseases, the Centers for Desease Control and Prevention, St. Jude Children's Research Hospital, and many others [6], [7]. The National Center for Biotechnology Information (NCBI) has developed the Influenza Virus Resource [10] providing a public access to influenza sequence data and convenient interface for constructing and viewing multiple sequence alignments and trees, as well as performing other kinds of data analysis.

Dealing with large amounts of data requires more sophisticated exploratory analysis to be available through web resources. An interactive web tool should provide a fast performance and represent the results in an easy-to-comprehend form that allows convenient manipulation of the data. The approaches based on manipulation of individual sequences are not very useful for large datasets for two reasons: first, representation of the whole dataset with a fine level of detail, such as shown in Figure 1, is very difficult to comprehend [9], [1]. Second, building large phylogenetic trees with all details resolved requires excessive computational resources. In the case of preliminary data analysis, this formulation leads to a problem that is much more complex than required to serve most of the user's exploratory needs.

Our approach is to offer a hierarchy of data represen-

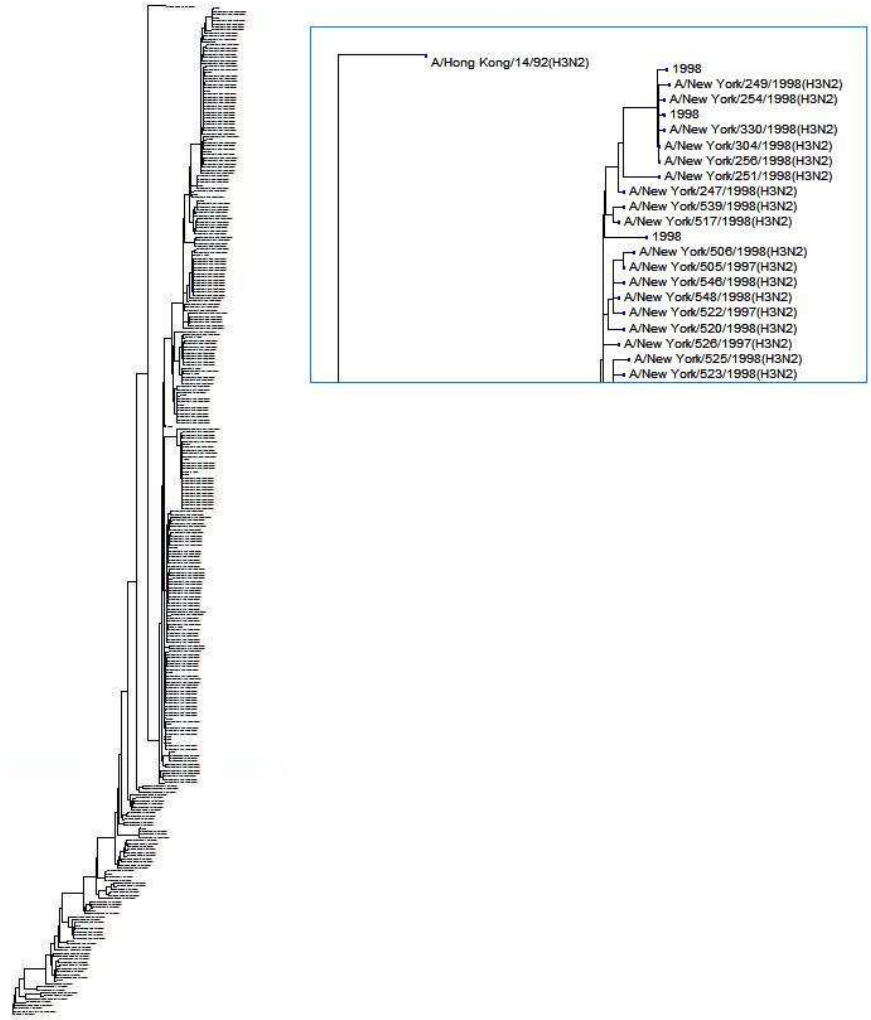

Figure 1. An full-resolution tree built using the neighbor-joining method for 380 HA protein sequences for Influenza A H3N2 viruses extracted from human hosts during 1968-1998. The top of the tree is enlarged in the small window.

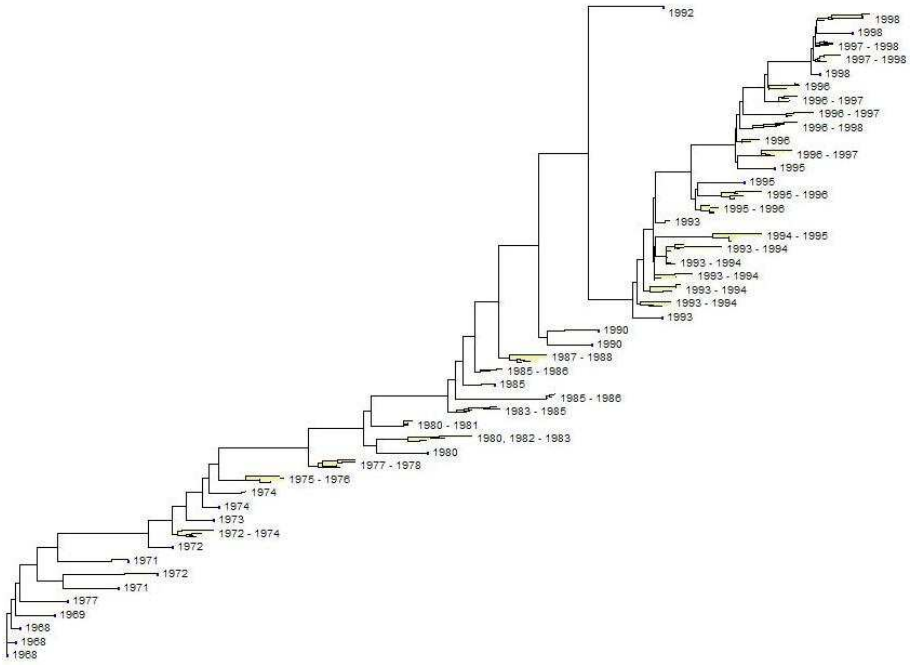

Figure 2. An aggregated tree built for the same dataset as in Figure 1 (from the same full tree calculated using the neighbor-joining method).

tations with different data coverages and resolutions. This can be done efficiently because the structure of a typical influenza dataset is characterized by low estimated values of the Kolmogorov (box) dimension (between 1 and 2 for typical HA protein datasets, as demonstrated below).

In [11], the authors introduced a visualization methodology based on the adaptive tree aggregation. This approach is discussed in section 2.

In section 3 we discuss the advantages of importance sampling for exploratory data analysis of influenza sequence datasets: while a tree build for a global sample provides a coarse-level representation of the whole dataset, it can be complemented by trees showing more detail in chosen areas. A hierarchy of sets of well-scattered points (ϵ-nets) is constructed. We show how to perform a local refinement gradually, using a multiscale hierarchy of ϵ-nets. It allows to take into account both global dataset structure and local details.

2. Adaptive tree visualization

In [11], we proposed an adaptive approach to visualize the dataset in an aggregated form adapted to the user's screen. It allows the user, depending on the task and need for details, to interactively refine or aggregate visualization of different parts of the dataset. The essential parts of our technique are the follows:

- Representation of a large tree by a smaller tree having aggregated groups as terminal nodes;
- Placing a specially constructed tree to show the structure of each aggregated group at the sub-scale resolution level.
- Creating metadata description for each aggregated group from the original metadata.

An example of aggregated tree is shown in Figure 2.

Sub-scale resolution. An aggregated tree represents groups of sequences as named terminal groups, with name of the group shown on the screen. Each named group occupies vertical space that determined by the height of the font used in annotation. Available vertical space, usually 10-12 px, can be used to show the structure the named group as a small subtree that we call *sub-scale resolution subtree*. Figure 3 illustrates transformation of a subtree in its sub-scale resolution representation. When aggregated groups of sequences are created, they form an abstracted description of the group to annotate the tree. We can summarize the group using descriptive characteristics: virus type, subtype, year of extraction; season of extraction, geographical location, etc.

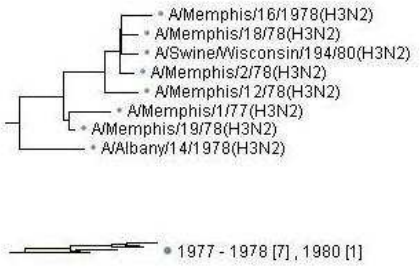

Figure 3. A subtree of the full tree(top) and its sub-scale resolution representation in the aggregated tree (bottom).

In addition to generating group names based on the metadata, datasets represented by trees to be searched using both structured and unstructured metadata, including sequence names. The search results are shown as individual sequences, where resolved, or number of sequences in named groups satisfying the search criteria.

3. Adaptive focusing using importance sampling from the dataset

While the adaptive aggregation technique described in section 2 addresses the visual representation issues, it does not address the issue of computational complexity. When this approach is applied straightforwardly, a full tree is calculated and then aggregated. In many situations, a significant part of the computational work would be unnecessary and wasteful: refinements of most of the subtrees of the aggregated tree will never be requested by the users and the details hidden there will never be seen. The approach we are exploring for large datasets is to provide an overview of the whole dataset first, and the ability to focus on details as needed.

Of course, the ability to focus efficiently requires a dataset to have a certain structure. This is the case for a typical influenza dataset, when either protein sequences or nucleotide sequences for the coding regions for the same segment are considered for viruses of the same serotype, and sequences are full length or almost full length. The property we are interested in is the Kolmogorov dimension of the dataset, also known as a box-dimension, box-counting dimension and metric dimension [3], [5].

For a non-empty bounded subset F in R^n, let $N_\delta(F)$ be the smallest number of sets of diameter δ which can cover F. The lower and upper box-counting dimensions are defined in [5] as

$$\underline{\dim}_B F = \underline{\lim}_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta},$$

$$\overline{\dim}_B F = \overline{\lim}_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta},$$

If these are equal, the common value is called *box-counting dimension* of F.

When these mathematical constructions are applied to real life examples, only relevant diapason of scales can be considered and the limits are not defined. Still, one can use the formula

$$\dim_B(F, \Lambda_\delta) = \frac{\log N_\delta(F, \Lambda_\delta)}{-\log \delta}, \qquad (1)$$

to estimate box-counting dimension using δ-cover Λ_δ of a compact set F in a metric space (with $N_\delta(F, \Lambda_\delta)$ being number of δ-balls in that cover). Figure 4 shows esti-

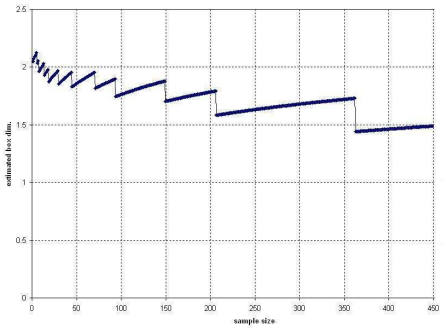

Figure 4. Estimated box dimension

mates of the box-counting dimension for a typical Influenza A dataset from 1968-2006 containing 1006 hemagglutinin protein sequences. Box-counting dimension estimates are between 1 and 2 for a variety of scales.

This result can be interpreted in a more pracrical way: the amount of d-balls required to cover set F is roughly

$$N_d \approx (D/d)^\alpha,$$

where $0 < \alpha \leq 2$, D is the diameter of the dataset (we used the value $\delta = d/D$ in (1)). This mathematical description of a typical influenza dataset structure corresponds to a biological description of influenza evolution as having mostly directed evolution with small lineages branching out and dying. A more attentive observer will find even smaller lineages branching out from minor branches and dying faster.

The multiscale structure of the influenza datasets can be used to focus on the area of interest by refining an initial coarse-level representation.

Let us introduce neccessary mathematical constructions. In mathematics [8], [3], [5], a subset M_ϵ of a set F in a metric space is called ϵ-net, if ϵ-balls with centers in elements of M_ϵ cover F. Of course, we are interested to find a minimal ϵ-net or at least a small one.

In a case of set F in a finite metric space with distance function $d(\cdot,\cdot)$, it is possible to find a finite ϵ-net M_ϵ whose elements are separated by from each other at least by distance ϵ using the following algorithm:

Algorithm 1. Building an ϵ-net

Set $\Lambda = F$ **and** $M_\epsilon = \emptyset$**.**
While ($\Lambda \neq \emptyset$ **)** {
  Select an arbitrary $\zeta \in \Lambda$ **and move it to** M_ϵ**;**
  Exclude from Λ **all** $\eta \in \Lambda$ **such that** $d(\eta,\zeta) \leq \epsilon$
  and create set $\Omega_\epsilon(\zeta)$ **containing these points.**
}

As a result, ϵ-net M_e is built and ϵ-neighnborhood $\Omega_\epsilon(\zeta)$ is defined for each $\zeta \in M_\epsilon$. Define $\gamma_\epsilon(x)$ as a center of ϵ-neighborhood the point x is assigned to (e.g., $x \in \Omega_\epsilon(\gamma_\epsilon(x))$).

We build a hierarchy of ϵ-nets with increasing resolution, starting with $\epsilon_0 = D/2$, where D is an estimate of the diameter of the dataset, and taking $\epsilon_{k+1} = \epsilon_k/2$ for $k = 0, 1, 2, \ldots$. The diameter of the dataset, $D(F)$, can be estimated by

$$D_x(F) = \max_{y \in F} d(x,y).$$

In a metric space, $D(F)/2 \leq D_x(F) \leq D(\Omega))$. We will obtain an estimate using an arbitrary $x \in \Omega$ and refer to that estimate as D.

If a 2ϵ-net is known, one can perform a *local search* while building M_ϵ. It is obvious that for any $x, y \in F$ such that $d(x,y) \leq \epsilon$, there are $\zeta_x, \zeta_y \in M_{2\epsilon}$ such that $x \in \Omega_{2\epsilon}(\zeta_x)$, $y \in \Omega_{2\epsilon}(\zeta_y)$, and $d(\zeta_x,\zeta_y) \leq 5\epsilon$.

Algorithm 2. Building an ϵ-net when a 2ϵ-net is known

Set $\Lambda = F$ **and** $M_\epsilon = \emptyset$**.**
For (**all** $\zeta \in M_{2\epsilon}$ **)**{
  Move ζ **from** Λ **to** M_ϵ**;**
  Set $\Theta = \{\varsigma \in M_{2\epsilon} | d(\varsigma, \gamma_{2\epsilon}(\zeta)) \leq 5\epsilon\}$**;**
  Exclude from Λ **all** $\eta \in \Lambda \cap (\cup_{\varsigma\in\Theta}\Omega_{2\epsilon}(\varsigma))$
  and create set $\Omega_\epsilon(\zeta)$ **containing these points.**
}
While ($\Lambda \neq \emptyset$ **)** {
  Select an arbitrary $\zeta \in \Lambda$ **and move it to** M_ϵ**;**
  Set $\Theta = \{\varsigma \in M_{2\epsilon} | d(\varsigma, \gamma_{2\epsilon}(\zeta)) \leq 5\epsilon\}$**;**
  Exclude from Λ **all** $\eta \in \Lambda \cap (\cup_{\varsigma\in\Theta}\Omega_{2\epsilon}(\varsigma))$
  and create set $\Omega_\epsilon(\zeta)$ **containing these points.**
}

We are not discussing the data structures allowing the fast local search. These are described in computer science literature (see [3], [4] and references there).

Importance sampling. The immediate application of the created hierarchy of ϵ-nets is systematic importance sampling. One can use a coarse ϵ-net to sample overall dataset and a local part of an ϵ-net with smaller value of ϵ to represent a local structure in a selected neighborhood. While a tree built from a global sample will provide a coarse-level representation of the dataset, there is a problem with trees built from a local sample: it may show incorrect relationships for sequences near the boundary of the local neighborhood and fail to reflect the global dataset structure.

Gradual local refinement. This problem is well-known in finite-difference and finite-element computational methods, especially in relation to multigrid approaches [2]. When a grid is refined locally, the transition from coarse- to

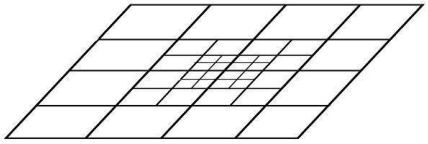

Figure 5. Grid refinement

fine-level descriptions is done gradually, with the local refinement area covered by multiple grids, and coarser grids extending to a wider area (see Figure 5).

We propose to use the *gradual refinement* technique for systematic sampling providing a good sample for building a tree: start with a coarse global sample (ϵ-net) and refine it gradually by adding sequences from ϵ-nets with smaller values of ϵ and in smaller areas around the point of focus.

Let $x \in F$ be a point of focus, R^0_{fine} be the size of the area around x from where all elements are taken, ϵ_{fine} be a resolution of the fine-level ϵ-net, R_{fine} be a a size of the fine-level resolution area around x ($0 \leq R^0_{fine} \leq R_{fine}$). From each of ϵ-nets in the hierarchy, we include an area around x with size $R = R_{fine}(\epsilon/\epsilon_{fine})$. If $\epsilon_k = 2^{N-k}\epsilon_{fine}$, than $R_k = 2^{N-k}R_{fine}$.

Algorithm 3. Locally-refined systematic sampling

Set $\Upsilon = \emptyset$**.**
Include in Υ **all** $\eta \in F$ **such that** $d(\eta, x) \leq R^0_k$**.**
For ($k = 1, \ldots, N$ **)** {
  Calculate $R_k = R_{fine}(\epsilon_k/\epsilon_{fine})$**;**
  Include in Υ **all** $\eta \in M_{\epsilon_k}$ **such that** $d(\eta, x) \leq R_k$**.**
}

As a result, we construct a set of sequences Υ that can serve as an input for a tree algorithm to build a focused tree. Note that while a gradual refinement is used to calculate a correct focused tree, we are not under obligation to show

the whole focused tree including poorly resolved areas. Of course, we can visually represent the part we are focused on and leave poorly resolved areas behind the screen.

4. Discussion

Adaptive aggregative visualization provides the user with a convenient way to view and manipulate the data hierarchy. This approach has been implemented and demonstrated to be efficient and useful. In addition, the progress towards larger datasets requires more advanced computational approaches to avoid excessive computation at the exploratory analysis stage. In section 3 we described our ongoing work on multiresolution tree algorithms facilitating efficient local refinement and allowing efficient facilitation of user requests to focus tree representation on a selected part of the data.

5. Acknowledgements

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

The authors are thankful to David J. Lipman and Stacy Ciufo for productive discussions.

References

[1] J. Baron. *Thinking and Deciding*. Cambridge University Press, 3 edition, December 2000.

[2] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, April 1977.

[3] K. L. Clarkson. Nearest-neighbor searching and metric space dimensions. In G. Shakhnarovich, T. Darell, and P. Indyk, editors, *Nearest-Neighbor Methods in Learning and Vision*, pages 15–59. The MIT Press, 2005.

[4] M. de Berg Mark van Kreveld Mark Overmars Otfried Schwarzkopf. *Computational Geometry. Algorithms and Applications*. Springer-Verlag, 2 edition, 2000.

[5] K. Falconer. *Fractal Geometry*. John Wiley and Sons, 2 edition, 2003.

[6] A. S. Fauci. Race against time. *Nature*, 435(7041):423–424, May 2005.

[7] E. Ghedin, N. A. Sengamalay, M. Shumway, J. Zaborsky, T. Feldblyum, V. Subbu, D. J. Spiro, J. Sitz, H. Koo, P. Bolotov, D. Dernovoy, T. Tatusova, Y. Bao, K. St George, J. Taylor, D. J. Lipman, C. M. Fraser, J. K. Taubenberger, and S. L. Salzberg. Large-scale sequencing of human influenza reveals the dynamic nature of viral genome evolution. *Nature*, 437(7062):1162–1166, October 2005.

[8] L. V. Kantorovich and G. P. Akilov. *Functional Analysis*. Pergamon, 2 edition, 1982.

[9] G. Mather. *Foundations of Perception*. Psychology Press, 1 edition, January 2006.

[10] The National Center for Biotechnology Information(NCBI). The NCBI Influenza Virus Resource. `http://www.ncbi.nlm.nih.gov/genomes/FLU/FLU.html`.

[11] L. Zaslavsky, Y. Bao, and T. A. Tatusova. An adaptive-resolution tree visualization of large influenza virus sequence datasets. In I. Mandaiu and A. Zelikovsky, editors, *Bioinformatics Research and Applications*, volume 4463 of *Lecture Notes in Bioinformatics*, pages 192–202. Springer-Verlag, 2007.