# ANALYSIS OF HOST-PATHOGEN INTERACTIONS VIA CLUSTERING, STATISTICAL ANALYSIS, AND DATA VISUALIZATION

A Dissertation

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

SAMANTHA WARREN

Dr. Dmitry Korkin, Dissertation Supervisor

MAY 2015

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

ANALYSIS OF HOST-PATHOGEN INTERACTIONS VIA CLUSTERING,

STATISTICAL ANALYSIS, AND DATA VISUALIZATION

presented by Samantha Warren, a candidate for the degree of doctor of philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Dmitry Korkin

Professor Gavin Conant

Professor Dong Xu

Professor Jianlin Cheng

I would like to dedicate this dissertation to my wonderful, supportive family: my mother Tracy, father Nick, and sister Sabrina.  I couldn’t have done any of this without them.

# ACKNOWLEDGMENTS

I would like to acknowledge contributions from my peers in the Korkin Lab: Andi Dhroso, Xingyan Kuang, Thanh Thieu, Hongzhu Cui, Nathan Johnson, Nan Zhao, and Xaoimin Wang. I would also like to acknowledge Mike Phinney and Eva Petakovic for their help with developing BacPaC.

Additionally, I would like to thank the following faculty members for their continued support: Dmitry Korkin, Gavin Conant, Melissa Mitchum, and Chi-Ren Shyu.

I would, also, like to acknowledge funding from the DoE GAANN Fellowship (P200A100053) for funding my graduate studies.

# TABLE OF CONTENTS

# LIST OF FIGURES

Figure Page

# LIST OF TABLES

Table Page

# Abstract

Infectious diseases are caused by a variety of agents: viruses, bacteria, parasites, or even proteins. Using existing state-of-the-art methods and tools I developed myself, I studied aspects of infectious agents. To find the most conserved and diverse regions of influenza A proteins, I found clusters of extremely conserved or diverse residues. Because traditional methods of clustering proved ineffective for diverse regions, I developed a Metropolis Criterion Monte Carlo (MMC) clustering algorithm to discover clusters of extremely diverse regions. In addition to viruses, I studied pathogenic bacterial proteins known as effectors. Using an in-house prediction method, Preffector, I generated predicted effectors for 14 bacteria and created a database and webserver to hold relevant information: BacPaC. BacPaC uses intuitive visualizations and script-generated profile pages to display relevant data about the predicted effectors. Finally, I applied structural modeling and docking techniques to soybean proteins that are known to incur resistance to nematodes. For each of these studies, I used clustering, data analysis, and data visualization to better understand infectious agents.

# 1 Introduction

The study of host-pathogen interactions encompasses all aspects of the interactions between pathogens – viruses, bacteria, and parasites – and their respective hosts. Aspects of these interactions include host resistance mechanisms, pathogenic infection mechanisms, and host and pathogen evolution. Pathogens affect all forms of life from single celled bacteria to plants to animals. In agriculture, parasitic infections and viruses cause substantial damage to crops and livestock resulting in lost profits and public health issues. In humans, viral and bacterial infections pose a serious threat to public health and result in substantial loss of life and even economic damage. In the case of the 2009 H1N1 influenza pandemic, the death toll was quite low, but the overall economic damage was substantial due to loss of work hours and cancelled travel plans. Aside from viral infections, bacterial infections result in worldwide devastation, especially in regions of the word where medical care is not readily available. A class of these infections are called neglected tropical diseases (NTD). For a disease to be classified as a NTD it must primarily affect people in undeveloped nations and lack attention from the research community.

The study of host-pathogen interactions using experimental methods is both expensive and time consuming. To alleviate these problems, computational methods have been developed, including: supervised and unsupervised learning, data visualization, sequence analysis, database structures, and structural prediction. These bioinformatics methods are used to do everything from comparative genomics to computational drug design. In my work, I used unsupervised learning, clustering, and structural bioinformatics to study: (i) evolutionary patterns in viral genomes, (ii) determine the key players in bacterial infections, and (iii) determine the functional effects of disease-associated polymorphisms in soybeans.

## *1.1 Pathogenic agents and mechanisms*

### 1.1.1 Influenza virus

The influenza virus is from the family *Orthomyxoviridae* and has been classified into three genera: influenza A, B, and C [1]. In this work, I will be discussing only influenza type A. Influenza A has a negative sense ssRNA segmented genome composed of 8 segments: HA, M, NA, NP, NS, PA, PB1, and PB2. These 8 segments code for 11 proteins: HA, M1, M2, NA, NP, NS1, NS2, PA, PB1, PB1-F2, and PB2 [2, 3]. At the time of my studies, PB1-F2 lacked sufficient structural and sequential data for the analyses performed, so it was not included. HA, NA, and M2 exist in the membranous coat of the virus and function in host attachment, in host entry, and as an ion channel, respectively [4]. PA, PB1, and PB2 function as the viral RNA polymerase and, along with NP proteins, form the viral RNP complex that is necessary for viral replication [5, 6]. NP oligimarizes to form a protein-RNA complex that attaches to the polymerase complex. M1 is the highly conserved matrix protein that is known to interact with the vRNP complex and even the internal portions of HA and NA during viral assembly. NS1 is responsible, primarily, for host cell control.

NS1 inhibits host cell interferon (INF) response and binds to mRNA, preventing host mRNA from leaving the nucleus; thus, increasing the expression of vRNA. Finally, NS2 (NEP) works as a nuclear export protein that helps to bring vRNP to the assembling virons.

Because of influenza A's zoonotic properties and segmented genome, studying its evolution is complicated. Influenza A has two forms of evolution: antigenic drift and antigenic shift [1, 7]. Drift occurs via point mutations that occur during RNA replication. This drift is the mechanism that is most commonly associated with the term evolution. Shift, on the other hand, occurs through the reassortment of the genomes of two or more viruses coinfecting a cell. This second mechanism makes studying influenza A evolution very difficult, because the evolutionary history of one gene may differ from that of a second. It is often difficult to determine phylogenies of strains overall due to influenza's segmented reassortable genome. It is, however, possible to study the phylogenies of individual genes. In order to understand influenza A evolution, one must look at each protein individually and understand how the evolution of the individual proteins affects the interactions between the proteins.

Influenza A, a zoonotic virus, infects a wide variety of vertebrate hosts: humans, dogs, birds, pigs, cats, horses, among others. Human-affecting subtypes also infect birds, a natural reservoir, and often pigs. Typically, when a host shift occurs to humans, it is transferred from livestock. Since influenza is endemic in avian and swine livestock populations, cross-species infections occur frequently. All recorded influenza A pandemics – caused by H1N1, H2N2, H3N2, and H5N1 – resulted from reassorted strains between an endemic livestock strain and a currently circulating human strain [8].

### 1.1.2 Bacterial effectors

In order to attack host cells, Gram-negative bacteria have secretion systems. These secretion systems work by either pushing effector proteins into the extracellular matrix or directly into the host cell like a needle. Secretion systems come in seven varieties, one of which is found exclusively in *Mycobacterium* (Type VII) [9]. The type III secretion (T3SS) is the most studied system because it is the primary secretion system of *E. coli* [10]. Many bacteria have multiple secretion systems. Secretion-related genes tend to be grouped on bacterial genomes in structures called pathogenicity islands: this structure gives potential insights to their evolutionary origins.

The secreted proteins are called effector proteins. The function of these proteins range from essential cellular functions to toxins. The purpose of these proteins is to hijack host cellular function for the good of the infecting bacterium. These types of attacks can cause the host cell to voluntarily die (cellular apoptosis) or even create excessive amounts of a nutrient that the infecting bacterium requires. Because effector function is so diverse, it is difficult to detect them; therefore, very few bacterial effectors have been experimentally characterized. Computational solutions have emerged, but with limitations. Available prediction methods only cover one secretion system – either Type III or Type IV [11-15]. Since (i) there are seven secretion systems and (ii) most Gram-negative bacteria have multiple secretion systems, these methods are inadequate for predicting effectors on a large scale.

## *1.2 Data analysis methods in bioinformatics*

### 1.2.1 Clustering of biological data

Large-scale analyses of biological data are becoming more common as information becomes more available and methods develop to handle the data. Because we typically do not know the expected result of our analysis, we need to use unsupervised learning methods to discover patterns within our data. Depending on the quantity, distribution, and representation of the data, we select one of a variety of clustering algorithms. For large quantities of data (tens of thousands of data points) K-means is an ideal algorithm because of its simplistic implementation and low runtime [16]. For moderate sized datasets it is imperative to understand the underlying distributions within the data. If clusters are spheroid in shape, centroid-based algorithms such as K-means often work well. If, however, clusters are likely to be elongated, C-shaped, or form concentric circles density-based algorithms such as DBSCAN [17] and OPTICS [18] are more effective. For instances where little is known about the underlying data distributions or cluster topology and a distance threshold is unknown, hierarchical clustering may be appropriate [19]. Hierarchical clustering generates a dendrogram representing the clusters that would be generated at several thresholds. Using one of many methods, including knee and elbow methods, a reasonable threshold for that data can be determined.

With any clustering algorithm, the most difficult problem lies with evaluating the final clustering. For distance-based clustering, the Minkowski distance is a good measurement of compactness [20]. This measure fails, however, with elongated, irregular clusters. In addition, there may be a measure related to the biological properties of the data. For instance, when finding clusters of similar genes, distance-based clustering is appropriate, but using a similarity measure of the genes – similarity among Gene Ontology or conservation – may be a valuable way to evaluate the clustering. These intrinsic properties of the data have been exploited for several types of clustering analyses. One example is the FLAME algorithm which was designed for microarray data to determine clusters of similarly expressed genes [21]. For most applications, existing clustering algorithms can be adapted to suit the user's needs; but, in many cases, a new algorithm needs to be developed to capture the appropriate clustering of biological data.

### 1.2.2 Data mining techniques for host-pathogen interactions

Due to the development of high throughput data technics, the size of biological data sets has been growing at an astounding rate. The development of NextGen sequencing has made it possible to sequence DNA in a massively parallel manner with high accuracy [22]: the result has been an explosion in the number of sequences available for not only large genomes but also smaller genomes. For the first time, the full genomes of both hosts and pathogens are available. Additionally, the development of microarray data and ChipSeq approaches have given rise to an entire field of bioinformatics: gene expression analysis. With this massive expansion of available data, valid methods for data analysis have never been more vital. Though many data mining techniques currently exist, many are inadequate to capture the intrinsic information of biological data, especially host-pathogen interaction data. Even determining whether a scholarly article contains information of a host-pathogen protein-protein interaction is a challenge [23]. To further understand host-pathogen protein-protein interactions, groups have attempted to mine relevant sequential and structural data from massive databases such as GenBank and PDB.

# 2 Extreme Conservation of H1N1 Influenza

Influenza A H1N1 caused a pandemic in 2009 due to a strain that was transferred from swine to humans in Mexico. The resulting pandemic, though causing minimal loss of life, caused economic difficulties because of the loss of work hours and cancellation of travel plans. Research into the evolution of H1N1 is generally focused on the antigenic regions of HA and NA. When research is done on the evolution of other, more conserved, proteins, it is focused on specific regions of the protein surface. Additionally, research is done using either structure or sequence, but not integrating the two. To better understand the evolutionary patterns of the H1N1 virus, we determined the extremely conserved residues via a sequence analysis, then mapped those residues to a protein structure. We performed graph-based clustering on these conserved residues to find clusters of sequentially conserved protein surfaces. Finally we analyzed these clusters' relationships with host-pathogen and intra-viral binding sites.

## *2.1 Methods*

### 2.1.1 Data selection and alignment

Our sequence data selection protocol was carried out in three stages. First, a set of 1,100 complete genomes of H1N1 influenza was selected from the NIH Influenza Virus Resource [24]. All 100% identical sequences were pruned to a single example. Because most genomes had only fragments of the PB1-F2 sequence, we chose to use only the other ten proteins. During the second stage, redundant strains were identified: we defined two strains as redundant if the sequence identity for each of the ten pairs of proteins was greater than 95%. Sequence identity was calculated based on sequence alignments computed using MAFFT [25]. Finally, the strains were clustered into redundancy clusters, relative to their redundancy with each other, and a representative was selected for each cluster, resulting in 75 non-redundant strains. Using the remaining 1,025 sequences, we analyzed how the addition of sequences to the non-redundant set of 75 affected site-specific conservation. This analysis was also done using MAFFT [25].

### 2.1.2 Protein structure prediction and surface analysis

The accurate identification of the surface residues for each influenza protein is a critical step in our approach. The ideal method for inferring each surface residue is to compute a homology model for each protein sequence and using the model structure to define the accessible surface residues. However, making such inferences for each sequence is computationally expensive. Therefore, in our protocol, a single target sequence was randomly chosen from the selected strains of each of the ten proteins, and a corresponding protein structure was predicted using the comparative modeling software MODELLER (Table 1) [26]. Next, for each modeled protein, we identified exterior residues using the CalcSurface subroutine. This routine calculates the solvent accessible surface area (SASA) using the MolMol software package [27]. Residues with a SASA greater than 25% were defined as exterior. This threshold has been previously used to identify a protein's surface residues [28]. Finally, the surface residues of the remaining 74 strains were mapped from the modeled strain using the multiple sequence alignment.

| Protein | Strain | Template | Template subtype | Template similarity, % | Residues covered |
|---|---|---|---|---|---|
| **HA** | A/Fort Worth/50 | 1H0A (A) | - | 45 | 19-517 |
| **M1** | A/Iowa/1943 | 1AA7 (A) | - | 97 | 1-158 |
| **M2** | A/Iowa/1943 | 2KIH (A) | H5N1 | 89 | 23-60 |
| **NA** | A/Iowa/1943 | 3B7E (A) | H1N1 | 91 | 83-467 |
| **NP** | A/swine/Alberta/ OTH-33-8/2009 | 2Q06 (A) | H5N1 | 93 | 28-502 |
| **NS1** | A/Fort Worth/50 | 3F5T (A) | H5N1 | 90 | 5-202 |
| **NS2** | A/Iowa/1943 | 1PD3 (A) | H1N1 | 100 | 68-116 |
| **PA** | A/Iowa/1943 | 3HW3 (A) | H5N1 | 96 | 1-193 |
| | | 2ZNL (A) | H1N1 | 96 | 239-699 |
| **PB1** | A/Iowa/1943 | 2ZNL (B) | H1N1 | 100 | 1-15 |
| | | 3A1G (A) | H1N1 | 95 | 686-736 |
| **PB2** | A/Iowa/1943 | 2ZTT (B) | H1N1 | 94 | 1-36 |
| | | 2VQZ (A) | H3N2 | 95 | 318-457 |
| | | 3R2V (A) | H3N2 | 93 | 538-720 |

**Table 1| Coverage and sequential similarity of protein templates.**
To compute the protein models, we first selected a sequence and one or more templates for each protein (many of the proteins needed multiple structures in order to cover most of the sequence). To select the templates for PA, PB1, and PB2 we chose the PDB references with the highest coverage and best resolution. For the others, we used MODWEB, which will automatically pick the best template. We picked the sequence (or strain) based on the sequence alignment. We generally selected either the sequence with the least number of gaps or the smallest number of unique gaps. The sequence similarity between the template and sequence is significantly high due to the high conservation between strains (Table 2).

#### 2.1.2.1 Template-based modeling of viral proteome

All proteins were modeled using MODELLER [26], a software suite that is used to generate, refine, and analyze protein structures. I used the template-based modeling portion of MODELLER to generate models for a representative of each protein of the influenza proteome. MODELLER works by first aligning the target (sequence to be modeled) sequence and the template (known protein structure) sequence. It is essential that the alignment be of the highest quality or the resulting model may be of low quality. Achieving this quality can be difficult because even though the template and target sequences may share an identity much higher than the 30% minimal threshold, the alignment may contain gaps. These gaps will dictate where MODELLER will have to predict *ab initio*, without a template. Once a suitable alignment has been determined, MODELLER maps the bond lengths and dihedral angles (ϕ and φ) from the template structure to the target sequence based on the alignment. Finally, MODELLER refines the homology model based on sampled distributions of dihedral angles for a given residue type in such a way that spatial constraints are not violated.

In order to evaluate a model, I look at the RMSD (root-mean-square distance) between the model and template as well as the DOPE (Discrete Optimized Protein Energy) score, a score used by MODELLER to evaluate the nativity of a model. If concerned about the overall scores, one can perform refinement of the model. This improvement is performed though loop refinement, which performs an *ab initio* folding of a non-structured region of the protein. This approach is often used to more accurately model linker domains of multi-domain proteins and flexible loops within domains.

#### 2.1.2.2 Solvent Accessible Surface Area (SASA) calculation

Solvent Accessible Surface Area (SASA) is used to distinguish between interior and exterior residues. Making this distinction is useful in the determination of protein surfaces and identifying which residues, or even atoms, are on the surface. Generally, SASA algorithms determine the area of an atom or residue that is exposed to a solvent.

This value is the equivalent of asking: what proportion of the atom is on the surface? The calculation of the exposure of a residue as a whole is:

$$Exposure(r) = \frac{\sum_{a \in r} Exposure(a)}{\sum_{a \in r} SurfaceArea(a)}$$

To determine the exposure of each atom, the SASA algorithm used in MolMol uses the ball rolling algorithm from ML Connolly [29]. This algorithm, from a theoretical standpoint, rolls a ball of a given radius (1.4Å default for MolMol) over the surface of a protein to determine the surface. In reality, however, the algorithm generates spheres for each atom and tori for each pair of contacting atoms. Here two atoms *i* and *j* are in contact if $radius(i) + radius(j) + radius(probe) < distance(c(i), c(j))$, where $c(i)$ is the center of atom *i*. This simplification ensures that the ball, or probe, can be rolled around the contract region of the two atoms. The equations describing each of these tori and spheres are described in ML Connolly[29].

After the calculation of each possible sphere and tori, the algorithm determines whether each torus is free, non-free, or buried. A free torus is one that does not intersect with any other torus, where a non-free torus has at least one collision with another torus. A buried torus is one that has collisions at all points, meaning that there is no point of the torus that is on the surface of the protein. For non-free tori, the probe is placed on all points where three atoms contact. Since the probe is spherical, the surface created at the mutual contact of three atoms is a spherical triangle. This is known as a concave surface. Each of these concave surfaces are then connected with their adjoining tori.

Upon determining the connections of the edges of the tori and concave surfaces, we now have all saddle surfaces. To determine the remaining convex regions, the spheres calculated from each atom are tested for intersection with tori and convex regions. Since each partial surface is represented by an equation, the algorithm can calculate surface collisions using basic geometry. These surfaces and collisions are represented by a graph where the surfaces are vertices and the collisions are edges. This graph is then interpreted by a graphical interface to show a surface.

Given this calculated surface, determining surface area is straight-forward. Since the surface is represented by a graph, the surfaces generated from each atom are detected and the edge list for those surfaces is used to calculate the total exposed surface. Each surface was generated using a probe of identical size and expressed with a standard formula of either a sphere or torus, so the surface of these sections is calculated using basic calculus.

### 2.1.3 Inference of patterns of molecular evolution for surface and interior residues

Using the structural information obtained from the comparative modeling, we explored the difference between patterns of sequence evolution of the proteins' interior and surface residues. Specifically, we fit three models of sequence evolution to these data using maximum likelihood, as implemented in the HyPhy software package [30]. The first and most restrictive model $M_{uniform}$ requires that the estimated branch lengths of the surface and interior partitions be identical. Thus, this model allows for no overall difference in the rate of evolution between the surface and interior residues. In the second model, $M_{scaled}$, we relaxed this assumption slightly to allow two partitions to have branch lengths that differ by a scaling constant $\alpha$. Thus, each branch length for the surface partition is multiplied by α (generally <1.0) to give a corresponding length for the interior partition. In the third model, $M_{arbitrary}$, the branch lengths of the two partitions are estimated completely independently. We note that our models do not explicitly take into account rate heterogeneity. Phylogenetic analyses typically treat rate heterogeneity as a poorly understood nuisance parameter [31]. However, as we have previously discussed, a significant contributor to this variation is the variation between surface and interior residue selective constraint [32, 33] that we have accounted for with our structural models.

These three models are nested with respect to each other, with model $M_{uniform}$ being a special case of both model $M_{scaled}$ (when α=1.0) and $M_{arbitrary}$ (when the paired branch lengths for the two partitions are equal). We can thus use a likelihood ratio test [34] to ascertain whether $M_{scaled}$ constitutes a statistically significant improvement over the null model $M_{uniform}$. The likelihood ratio compares the difference in log-likelihood between the two models to a chi-square distribution, where the number of degrees of freedom of that distribution is given by the number of excess parameters in the alternative model. For $M_{scaled}$, the parameter α adds one degree of freedom. Therefore, if the above test shows significant improvement for $M_{scaled}$, one can then explore whether the model may be further improved by allowing each branch to differ between the surface and interior residues (i.e., model $M_{arbitrary}$). We again used the likelihood ratio test: in this case there are 146 extra parameters, corresponding to the 147 extra branch lengths in $M_{arbitrary}$, minus the unnecessary α parameter.

#### 2.1.3.1 Generation of phylogenies using PhyML

PhyML is a program that generates phylogenies using BIONJ and refines them using Maximum Likelihood (ML) estimation – quite common in phylogenies. PhyML performs six basic steps: (i) determine pairwise evolutionary distance, (ii) build a tree using BIONJ algorithm, (iii) determine likelihood for all subtree swappings and alternative branch lengths, (iv) improve likelihood of model by adjusting free parameters, (v) refine the tree by iteratively selecting modifications that improve the overall likelihood, and (vi) output the final tree when no modification can be performed that improve the likelihood of the tree. Each step of this process is essential, but I will be focusing primarily on steps 3 through 6. First, however, I would briefly like to discuss the determination of evolutionary distance and the BIONJ[35] algorithm and its improvement over basic neighbor joining algorithms.

Evolutionary distance between two sequences is determined by the number and types of changes between two sequences. Two sequences will have a small distance if they have few substitutions and/or the substitutions are similar. The effect of an amino acid substitution is typically determined by the BLOSUM62 substitution matrix, which is used by most sequence alignment programs. The algorithm used by PHYML is similar to DNADIST [36] algorithm from the PHYLIP [37] program. The BIONJ [35] algorithm takes the matrix of pairwise evolutionary distances as input and uses a neighbor joining (NJ) algorithm, similar to that of Saitou and Nei [38], to compute a phylogenetic tree. NJ algorithms create trees by taking the pair of taxa with that result in the minimal branch length and joining them into a single node, then replacing those two taxa in the distance matrix with the new node. This is done until the entire matrix has been reduced and all taxa are connected via one tree. BIONJ improves upon the standard NJ approach by incorporating the variances and covariance of the distance matrix to pick which neighbors should be joined at each step. The join that will minimize variance is selected.

Once PHYML has calculated the tree using BIONJ, maximum likelihood (ML) estimates are used to determine if/where improvements can be made to the tree. Two types of adjustments can be made: (i) branch lengths and (ii) subtrees. To begin with, PHYML optimizes the branch length by determining maximizing the likelihood:

$$L = \prod_{i} \sum_{h,h' \in A} \pi_h L(i = h|U) L(i = h'|V) P_{hh'}(l)$$

where *A* is the set of all amino/nucleic acids, $\pi_h$ is the *a priori* probability of *h*, $L(i = h|U)$ is the probability that position *i* is *h* given that the sequence is in set *U*, *U* and *V* are the subtrees on either side of the branch, and $P_{h,h'}(l)$ is the probability of *h* to change to *h'* in the interval *l*. A Newton-Raphson method is used to select an *l* such that *L* is maximized.

The likelihood of a specific configuration of subtrees

$$L(i = h|U) = \left( \sum_{g \in A} L(i = g|W) P_{hg}(l_W) \right) \times \left( \sum_{g \in A} L(i = g|Y) P_{hg}(l_Y) \right)$$

where *W* and *Y* are subtrees on the *U* side of the branch *l*. Combinations of *W*, *X*, *Y*, and *Z* – all subtrees – on the *U* and *V* sides of branch *l* are considered.

Once all possible alterations to the current tree are calculated using the above likelihood estimates, the algorithm accepts a certain number of alterations, which most greatly increase the likelihood of the entire tree, dependent on a λ parameter. If λ = 1, then all alterations will be applied. Conversely, if λ = 0, then none of the alterations will be applied. PHYML begins with λ = 0.75 and decreases λ with each iteration to ensure convergence. Once there are no more subtree swaps or branch length reassignments can be done that increase the likelihood, it is assumed that the tree has reached its maximum likelihood and that tree is returned. It is possible, however, that the tree is not truly the maximum likelihood tree, but rather a local maximum.

### 2.1.4 Automated conservation analysis pipeline

We next developed an automated computational pipeline to determine structurally conserved protein regions and assess their statistical significance. This pipeline was applied to study the extremely conserved regions of the H1N1 proteome (

Figure **1**) and consists of four basic steps. We first determined the conserved residues shared between a set of representative protein sequences. Second, we used the homology models of the H1N1 proteins to filter out the conserved residues in the core of each protein. Third, we clustered the remaining residues that are fully conserved on the surface into regions. Finally, we determined the statistically significant regions by employing a random model that generates surface regions with similar properties. The process is further described below.

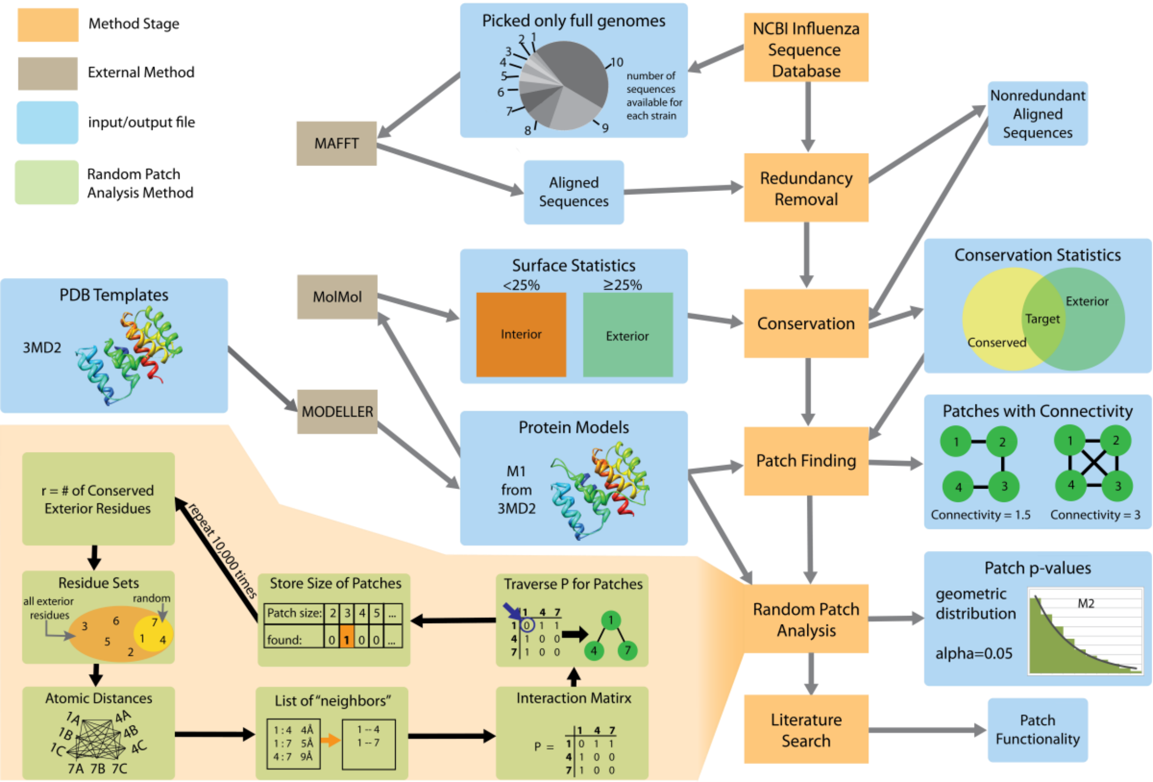


**Figure 1| Analysis pipeline.**
Our conserved patch analysis method consists of six stages (orange boxes): data collection, redundancy removal, conservation detection, patch finding, random patch analysis, and functional annotation of the conserved regions. The method integrates data from multiple sources (blue) and employs four previously developed software packages (grey): MAFFT, MolMol, and MODELLER. The random patch analysis stage is described in more detail below (peach).

To identify regions of extreme conservation, we aligned the set of 75 representative sequences for each of the 10 proteins and determined which of the surface residues were 100% conserved across all 75 sequences. Next, we calculated the Euclidean distance between all pairs of 100% conserved exterior residues. Pairs of conserved residues that were no farther than 6Å apart were defined as structural neighbors. The neighborhood relationship was then summarized as a binary contact matrix of a graph, and the whole set of surface residues were represented as a neighborhood graph with edges designated by the contact matrix. Finally, the surface residues were clustered into regions by defining each connected component of the neighborhood graph to be a cluster. In addition, for each region we calculated its size, contributing surface residues, and residue connectivity. The residue connectivity is defined as an average number of edges per vertex in the neighborhood graph.

To assess if the sizes of the observed regions were larger than expected by chance, we generated a sample of random patches using the corresponding MODELLER subroutine [39]. For each sample, the procedure randomly selects the same number of unique surface residues as conserved surface residues on the protein structure. We then apply the same clustering algorithm as the one discussed above to each of the randomly generated samples, obtaining a patch of neighboring residues and determining the size of the patch. We repeated this procedure 10,000 times (the number is selected as a trade-off between the sample size and the computational time of the random trial procedure), yielding a distribution of patch sizes expected for randomly selected groups of exterior residues. The conserved regions obtained from the real data were compared against this distribution, identifying significant regions. Specifically, we determined the *P*-value for each region size using a geometric distribution with a weighted average:

$$P - value = 1 - \left(1 - \frac{1}{avg(X)}\right)^{n}, \qquad avg(X) = \frac{\sum_{n=1}^{L} n * y_n}{\sum_{n=1}^{L} y_n},$$

where *X* is the set of all random patches and frequencies, *n* is conserved region size, $y_n$ is the frequency of a patch of size *n*, and *L* is the largest possible patch. For this weighted average, we also considered patches of size 1, the residues that were isolated after clustering. The addition of these residues was necessary for understanding the underlying distribution. The distribution appeared exponential, however since the distribution was of discrete values, we decided that a geometric distribution was a better choice. We then defined a region as significant if the *P*-value was less than 0.05.

Each statistically significant region was functionally annotated. Specifically, we mapped intra- and inter-species binding sites of the H1N1 influenza proteins collected from our database of macromolecular interactions DOMMINO [40] and PubMed literature search, and then determined if each of the conserved regions overlap with any of the mapped binding sites (Table 2).

| Protein | $r_e/r_i$ | Exterior / Interior | N of patches | Template coverage | Intra-viral interactions | |
|---|---|---|---|---|---|---|
| | | | | | **Literature** | **Structure** |
| HA | 1.2 | 0.53 | 3 | 88% | [41] | |
| M1 | 2.2 | 0.63 | 1 | 63% | [42, 43] | |
| M2 | 2.3 | 6.60 | 1 | 38% | [44-47] | |
| NA | 1.5 | 0.33 | 1 | 82% | | 3B7E |
| NP | 1.2 | 0.56 | 2 | 94% | [48-50] | |
| NS1 | 1.5 | 0.97 | 1 | 83% | [51] | |
| NS2 | 1.1 | 1.45 | (1) | 40% | [52-54] | |
| PA | 1.9 | 0.65 | 5 | 91% | [6, 55] | 2ZNL |
| PB1 | 1.2 | 1.15 | (3) | 7% | | 3A1G, 2ZNL |
| PB2 | 1.2 | 0.70 | 2 (2) | 47% | [56] | 3A1G |

**Table 2| Protein evolutionary rates and patch information.**
The ratio of protein evolutionary rates for the exterior and interior residues ($r_e/r_i$) was determined using HyPhy. Shown are the ratios for entire proteins. The significant regions are shown in the following column with regions that are biologically significant, but must be explained structurally rather than statistically in parentheses. For some viral proteins the homology models of do not cover the entire sequence due to the limited coverage of their templates. Shown is the percentage of the protein sequence coverage for each structural model. The last column summarizes the evidence for the intra-viral interactions in recent literature and from DOMMINO.

### 2.1.4.1 Graph-based clustering

Pseudocode:

```
Function: find_clusters
C=contact matrix of conserved exterior residues
p=current cluster
for i∊[0,length(C)-1)
  if C[i,0]== 1
       begin cluster
       search_vertical(i)
       Write p
end

Function: search_horizontal(int j)
for i∊[0,length(C)-1)
  if C[i,j]== 1
       C[i,j]=C[j,i]=0
p <- i
search_vertical(i)
end

Function: search_vertical(int i)
for j∊[0,length(C)-1)
  if C[j,i]== 1
     C[j,i]=C[i,j]=0
p <- i
search_horizontal(i)
end
```

The graph-based clustering algorithm described above searches a graph G(V,E), where *V* = {residue r| *exposure(r)* > 25%} and *E* = {1 if *distance*($v_i$, $v_j$) ≥ 6Å, 0 otherwise}, to find connected components. This is a recursive neighborhood searching algorithm. Given G(V,E), represented by the matrix C, the algorithm begins at C[0,0] and iteratively across C[i,0]. With each iteration, the algorithm checks if the $i^{th}$ element will begin a new cluster. If C[i,0] = 1, then the algorithm begins building a cluster *p* by searching down C[i,j] until the next element of the cluster is found – where C[i,j] = 1. Once this element is found

C[k,j] is searched recursively until a *k* is found such that C[k,j]=1. This search will continue recursively until there is no such *m* and *n* such that C[m,n] = 1 and *m* and *n* are reachable from *i*. At this point, the algorithm will move to the next *i* such that C[i,0] = 1. Once we have reached the end of C[i,0] the algorithm will terminate and all clusters – connected components – will have been detected. This algorithm runs in O($n^3$) time.

Given that this algorithm is graph-based, it can naturally be compared to other graph-based clustering algorithms. Similar algorithms do one of two things: (1) determine strongly connected components or (2) gathers clusters hierarchically. Strongly connected component (SCC) algorithms determine SCC, clusters, based on the local connectivity of a subset $H(V_0,E_0) \subseteq G(V,E)$ rather than simply determining sets of mutually reachable vertices. There are benefits to using SCC algorithms, mainly that they generate more compact clusters. At the time of this analysis, I had not yet determined the need for compact clusters. In a later analysis (see section 3), I appreciated the need for more compact clusters and elected to use a Density-Based clustering algorithm. The other subset of graph-based algorithms can be used to determine clusters hierarchically. This determination is performed using a dynamic threshold which refers to an Edge value, *value(E)*. In the context of residue connectivity across a protein surface, *value(E)* would refer to the distance between two vertices. In this case, hierarchical clustering would not be useful as the value of 6Å is a commonly used distance convention. Had we been unsure of an appropriate threshold, hierarchical graph-based clustering would have been considered.

## *2.2 Results*

### 2.2.1 Data collection

The initial set of H1N1 strains included 1,100 unique genomes, each containing ten sequences (2.1). We employed a redundancy filter with a whole-genome sequence identity threshold of 95% which yielded a final set of 75 strains (see 2.1) including 10 avian, 34 human, and 31 swine strains, with all strains dating between 1933 and 2009. The 1918 'Spanish flu' strain was not included in the final set due to several of its proteins having 100% sequence identity with the corresponding proteins in other strains, but was included as a case study. Nevertheless, the conservation of the surface regions between the 1918 and 2009 H1N1 pandemic strains was analyzed in detail (2.2.5). The average sequence identity between the individual proteins in our dataset varied from 88.7% to 96.4%. As expected, there were pairs of strains sharing identical or near-identical proteins, even when other proteins in these strains were less than 95% identical. No strains shared the same proteins with less than 60% protein sequence identity (Table 3).

| | HA | M1 | M2 | NA | NP | NS1 | NS2 | PA | PB1 | PB2 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Average** | 89 | 97 | 89 | 89 | 95 | 87 | 93 | 96 | 96 | 96 |
| **Minimum** | 77 | 92 | 71 | 77 | 86 | 60 | 76 | 88 | 86 | 90 |
| **Maximum** | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

**Table 3| Strain conservation across the ten proteins**.
The proteins vary in their conservation. When removing redundant sequences, we first calculated the pairwise conservation percentage for each individual protein. From this we calculated the average pairwise conservation. We also determined the minimum and maximum conservation.

### 2.2.2 Phylogenetic analysis of structure-based evolution

For each of the ten proteins, we computed the maximum likelihood phylogeny using PhyML [57]. We then fitted several models of evolution to these alignments (2.1). The most basic, $M_{uniform}$, requires all nucleotides in the sequence to evolve at the same rate. We compared that model to $M_{scaled}$, where the evolution rate of positions corresponding to surface residues was allowed to be more dissimilar than for interior residues. As expected, all ten proteins showed higher rates of substitution for surface positions (the rate ratio, $r_e/r_i$ was greater than 1.0; Table 3, likelihood ratio test, see section 2.1. We then investigated whether this pattern was the result of differing surface to interior constraints on the various branches of Figure 1, but found no such pattern. Similarly, $r_e/r_i$ varied only slightly for seven of the proteins: 1.1 for NS2; 1.2 for HA, NP, PB1, and PB2; 1.5 for NA and NS1. The ratio was considerable higher for the other three proteins: 1.9 for PA, 2.2 for M1, and 2.3 for M2.

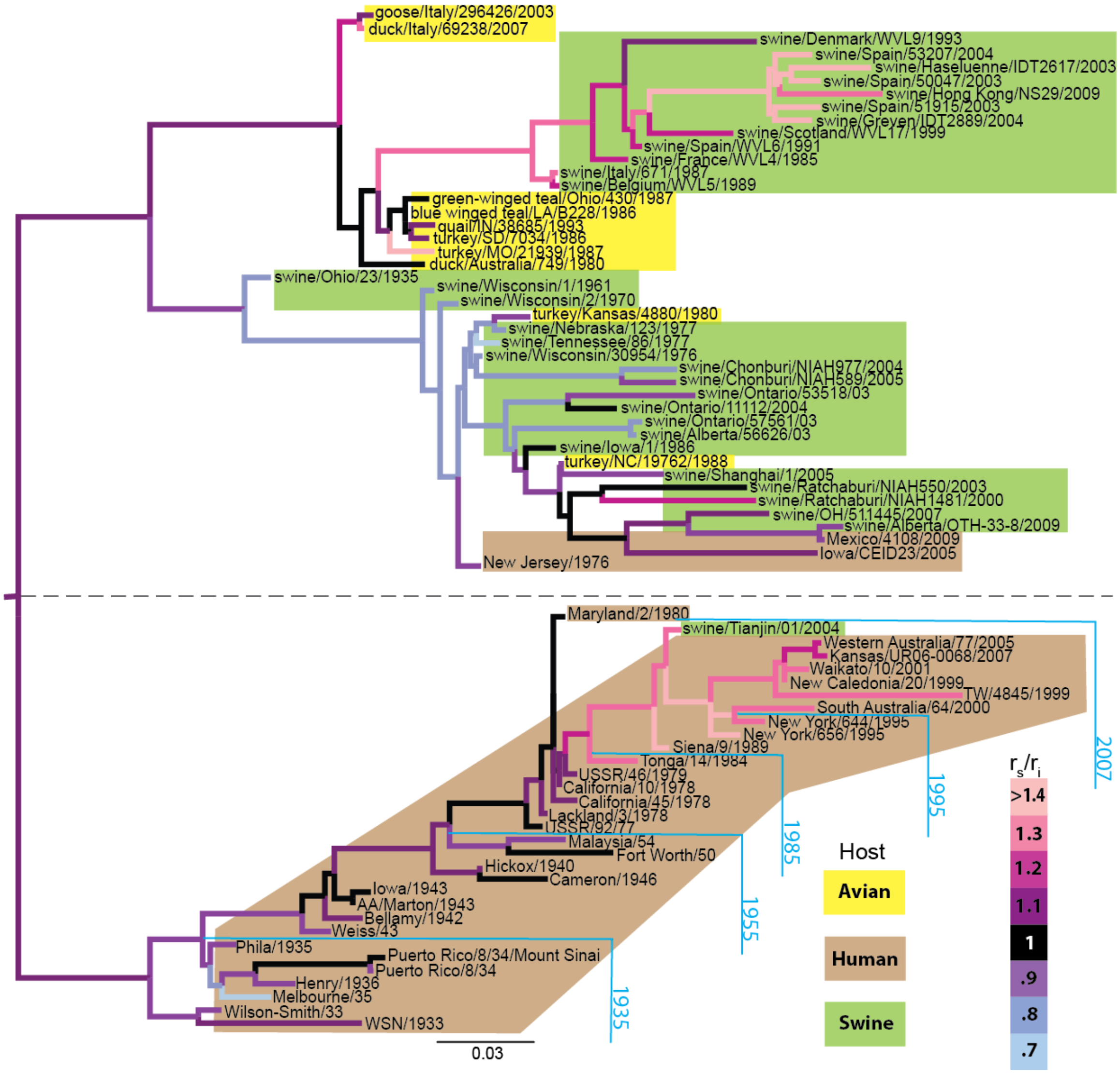


**Figure 2| Phylogenetic relationships, species derivation and relative evolutionary rates for 75 accessions of H1N1 influenza.**

Shown is the topology inferred for the HA protein (see subsection *Inference of patterns of molecular evolution for surface and interior residues* in *Methods*); other proteins show somewhat differing relationships (Supplemental online data). We also show the ratio of surface-to-interior amino acid substitutions ($r_e/r_i$), calculated as the difference between the branch lengths estimated from the exterior and interior residues. Variation in $r_e/r_i$ is illustrated from low to high with colors from blue to pink. Each colored box represents the organism of origin: Avian (yellow), Human (beige), and Swine (green). We note that the lower clade (separated by a dashed line) is composed almost entirely of human-derived strains, with the exception of one swine accession (Tianjin/01/2004). This clade also shows a fairly clear timeline (cyan). The upper clade, however, does not give such clear indications of timing.

The phylogenetic trees inferred were clearly separated into the host-specific lineages with the occasional inclusion of strains from other species (Figure 2). The human lineage in both HA and NA trees exhibits a strong ‘trunk-like’ temporal pattern that has been previously observed in the phylogenetic trees generated from whole-protein sequence alignments [58, 59] (Figure 2). In the case of PA, this pattern is less evident. A few human strains were found as a part of the swine clade, and a recent swine strain was found as a part of the human clade across all three analyzed proteins, indicating the bi-dimensional transmission of influenza A viruses between the animal and human interface. Interestingly, we found that after 1984, the surface-to-core ratio of human HA and NA proteins, but not PA proteins, becomes significantly higher. This observation suggests an increasing selective pressure on the surface residues of the former two proteins due to the widespread use of seasonal vaccination.

Unlike the human lineage, the swine and avian lineages of HA and NA trees did not exhibit the trunk-like pattern. Instead, the swine lineage was divided into two clades, one comprised primarily of North American strains and another comprised of Eurasian strains. Moreover, while Eurasian swine strains had a surface-to-core ratio that was generally higher than in North American strains, we did not observe the same sudden increase in the ratio values as a function of time, as we did in the human lineages. Finally, several human strains, namely Mexico/2009, Iowa/2005, and New Jersey/1976 were included in swine lineages of HA and NA proteins (Figure 2), representing spillover cases of H1N1 virus from swine to human. This situation was not necessarily the case for other influenza proteins, which may have originated in different hosts.

### 2.2.3 Homology modeling of the individual influenza proteins

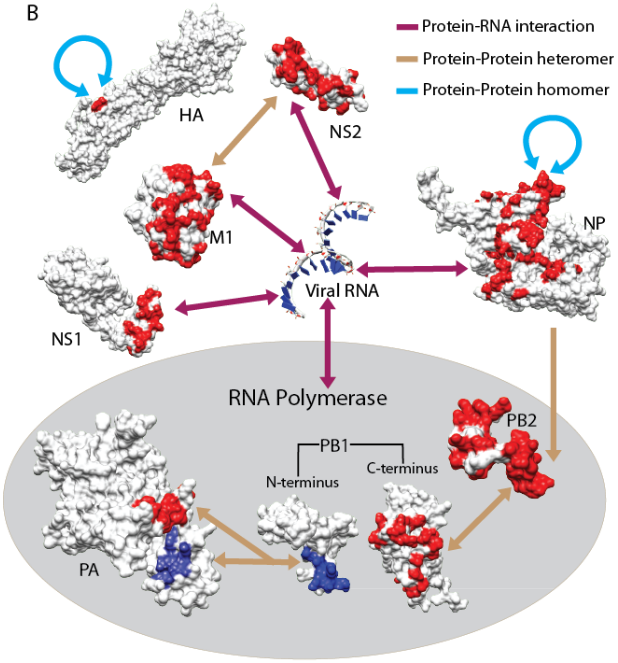
The structural analysis of H1N1 protein surfaces using homology modeling is challenging due to the limited structural template coverage of some influenza proteins. Three-dimensional structures of several influenza A proteins have been modeled before and used for functional and evolutionary studies [60-64]. Unfortunately, for some influenza proteins (M2, NS2, PB1, PB2) the templates cover only a small portion of the target sequence, while for other influenza proteins the entire sequence is covered by a single template or a number of templates with a little or no structural overlap (HA, M1, NA, NP, NS1, PA). Therefore, we used a single template as the basis for our models for seven proteins and a multiple-template strategy for the remaining three (Table 3). As a result, we obtained models covering almost entire sequences of 6 H1N1 proteins, with the exception of small N-terminal and C-terminal regions. Sequences of 3 proteins were partially covered by two or more fragments (PA, PB1, and PB2). Only one protein (M2) did not have a significant portion of its sequence (residues 23-60) covered by any structural template (Table 3); these regions were not modeled structurally. The average target-template sequence identity was 91% (minimal sequence identity was 45%). This high sequence identity, thus, allowed for an accurate determination of surface and core residues of H1N1 proteins based on the homology models.

### 2.2.4 Conserved regions on H1N1 proteins surface are associated exclusively with intra-viral interactions

Each H1N1 protein was found to have at least one evolutionary conserved region that was statistically significant (Figure 3 A, C). A literature search combined with a search of the DOMMINO database of macromolecular interactions [40] resulted in 8 proteins with regions that had been previously functionally described in the literature (17 papers in total) and 4 proteins that contained regions characterized by structural data (5 PDB structures in total) (Table 2). Even though each protein contained a significant region, some proteins had regions that required structural explanation, such as NS2, PB1, and PB2. The distributions of random patch sizes obtained for these proteins did not fit well using an exponential distribution. Specifically, the distribution of random patch sizes for NS2 closely resembled a linear stepwise function, and for structurally modeled fragments of PB1 and PB2 the underlying distributions favored the regions of maximum size. This observation can be explained by the large percentage of surface residues that are classified as conserved. Indeed, since a large number of surface residues are conserved, it is difficult to create several isolated regions of small size; thus, the typical regions are large. For M1, we also obtained a random patch size distribution, which appeared almost exponential with the exception of an additional peak. Finally, the M2 protein, with a similarly high substitution rate, had a significant region on its surface. However, the small size of the M2 structural model covers only part of its sequence, possibly giving rise to a spurious patch. The location of the modeled structure in the transmembrane region increases the likelihood of existence of such a patch.

A

| Protein | Interaction type | Patch Size | En(x) | Connectivity |
|---|---|---|---|---|
| HA | Homotetramer | 2 | 5.497 | 1.0 |
| M1 | Hetrodimer and RNA binding | 32 | 0.015 | 2.8 |
| NP | Homodimer, Heterodimer, and RNA binding | 50 | 0.010 | 3.0 |
| NS1 | RNA binding | 14 | 0.002 | 3.7 |
| NS2 | Heterodimer and RNA binding | 20 | 0.073 | 3.8 |
| PA | Heterotrimer/ RNA Polymerase formation | 6 | 1.090 | 3.3 |
| | | 8 | 0.675 | 3.3 |
| PB1 | Heterotrimer/ RNA Polymerase formation | 16 | 0.052 | 2.8 |
| | | 7 | 0.318 | 3.7 |
| PB2 | Heterotrimer/ RNA Polymerase formation | 22 | ~0.012 | 5.0 |



C

| Protein patches | HA | M1 | M2 | NA | NP | NS1 | NS2 | PA | PB1 | PB2 |
|---|---|---|---|---|---|---|---|---|---|---|
| Significant | 0 | 2 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| Marginal | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| Insignificant | 12 | 0 | 0 | 7 | 11 | 3 | 0 | 11 | 2 | 10 |

**Figure 3| Conserved regions are exclusively associated with known intra-viral interaction positions.**

A) Eight of the ten viral proteins have regions that are involved in known intra-viral interactions. For each interaction, we list the type of interaction, the size of the patch, $E_n(x)$, and the patch connectivity. We determine $E_n(x)$ as the expected number of randomly generated regions of a given size. We calculate the connectivity of the regions as the average number of neighbors each residue has in the patch. The color of the three right-most columns match to the color of the regions in panel B. B) Each of the eight proteins forms a unique interaction with (i) a copy of itself (indicated by a blue arrow), (ii) viral RNA (purple arrow), or (iii) another viral protein (tan arrow). Some conserved regions participate in more than one interaction. A uni-directional arrow indicates an interaction occurring between two proteins, but is not necessarily characterized by conserved regions on both proteins. The three proteins of RNA polymerase, PA, PB1, and PB2, are grouped by a grey oval. Shown is the interaction between the polymerase complex and the viral RNAs. C) The distribution of significant ($E_n(x) \leq 0.05$), marginal ($0.05 < E_n(x) \leq 0.1$), and insignificant ($E_n(x) > 0.1$) regions across all ten proteins.

Intriguingly, the functional annotations of the significant regions reveal that all the regions are exclusively associated with the intra-viral protein-protein and protein-RNA interactions (Figure 3B), with the exception of a single residue from a region on NP (Table 2). The protein-protein interactions include both homomers (self-interactions of proteins M1 [43], M2 [45, 46], NP [50], and NS1 [51]) and heteromers (interactions mediated by proteins M1 [42], NP [48], NS2 [52, 53], PA [6, 55], PB1 (PDB: 2ZNL, 3A1G), and PB2 (PDB: 3A1G)). Several of these proteins, including M1 [42], NP [49], NS1 [51], NS2 [54], PB2 [56], also had significant surface regions associated with protein-RNA interactions. The conserved regions share several interesting properties. First, we found that all interactions involved in the assembly of the RNA-polymerase complex included at least one region of extreme conservation. Second, while regions usually occurred on only one binding site of the interaction interface, we also found protein-protein interactions with the regions included in both binding sites (interactions between proteins PB1 and PA [6] (PDB: 2ZNL), PB2 and PB1 (PDB: 3A1G), and M1 and NS2 [42, 52, 53]). Finally, we found that NS2 had a conserved region annotated with multiple functions: the region from residues 65-72 is involved in both viral RNA and M1 binding, while residues 74-79 are involved exclusively in M1 binding (Table 2).

The inferred regions were only slightly affected when we additional sequences were introduced to the original non-redundant set of 75. Specifically, it took between 900 and 1000 sequences to introduce even a single non-synonymous polymorphism into a single influenza protein. Most strains experienced exactly one such mutation across their entire proteome. A particular set of outlying strains caused at most 5 polymorphisms and affected at most 3 different proteins. This set contains 14 proteomes that can be grouped by geographic location and close years, and within these groups, the sequence identity ranges from 97% to 100%. This indicates that these grouped sequences are in the same redundancy cluster, during the redundancy removal procedure and thus could have only a minimal effect, if any, on the analysis.

### 2.2.5 Regions of extreme conservation in 1918 and 2009 pandemics

Following the findings by Xu *et al* [65], which identified nearly identical functional sites shared between HA proteins of the 1918 and 2009 H1N1 pandemics, we compared our identified regions of extreme conservation across strains from both pandemics. Notably, all identified regions across all proteins were identical between the 1918 and 2009 strains. This finding is in agreement with the fact that the 2009 swine origin pandemic influenza A virus is thought to originate from a recent inter-species reassortment from swine to human, and another observation that same extreme regions were found not only between human H1N1 strains but also across swine and avian strains.

We finally sought to understand the relationship between the identified regions of extreme conservation and the evolutionary dynamics of the virus when treated with antiviral drugs. Specifically, we used recently reported viral population data obtained from an immunosuppressed patient infected with 3 variants of H1N1/2009 influenza and treated with neuraminidase inhibitors [66]. The data included a set of ten mutation sites from four proteins obtained using a deep sequencing approach: HA ($Val_{6}$, $Asn_{55}$, $Val_{125}$, $Thr_{220}$), NA ($Ile_{106}$, $Asp_{199}$, $Asp_{248}$, $His_{275}$), NP ($Ile_{100}$), and NS1 ($Ile_{123}$). These sites were mapped onto the homology models of the proteins and compared to locations of conserved regions (Figure 4). We found that none of the ten mutation sites belonged to any of the conserved regions.  Interestingly, NA's mutation site $\text{Ile}_{106}$ was in close proximity to residues 107 and 108, which belonged to a conserved region.  However, the mutation reported at this position (I106V) [66] is unlikely to cause any changes in the function associated with the conserved region due to similar properties of the residues.

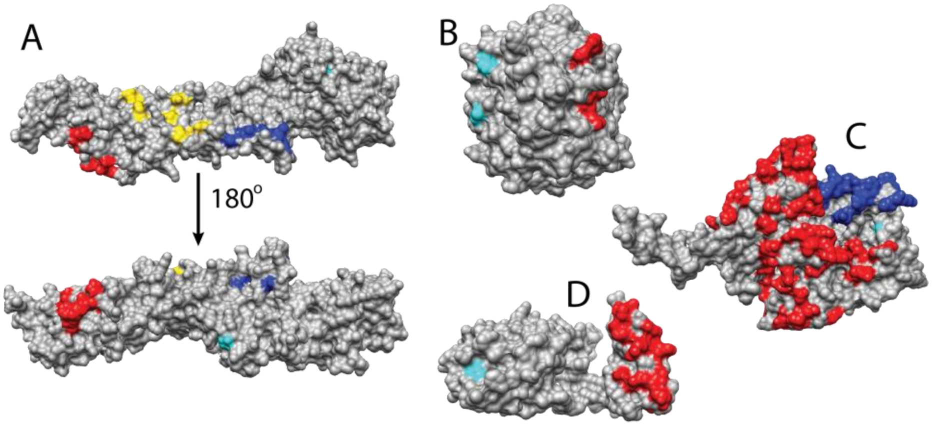


**Figure 4| Genetic variation of the viral population data obtained from a patient does not affect regions of extreme conservation.**
Shown are ten mutation sites (cyan) from four proteins, HA, NA, NS1, and NP, obtained using a deep sequencing approach. The mutation sites were mapped onto the structural models and their locations were compared to the conserved regions. Individual regions of extreme conservation were colored red, blue, and yellow.

## *2.3 Discussion*

### 2.3.1 Overview of results

The conservation of functionally important residues on protein surfaces has been well documented [67, 68]. In particular, several studies, both general and targeting specific protein families, determined the sequence and structure conservation of residues in the protein binding sites mediating intra-species protein-protein interactions [67, 69, 70].

However, the impact of the purifying selection on the protein binding sites in viral proteins is not clear, due to the intrinsic relationship between the intra-viral and viral-host protein-protein interactions. Unexpectedly, we gained new insight into the evolution of viral binding sites while addressing more general questions related to influenza protein evolution. The first question is whether the surface residues of the proteins evolve faster than the core residues, and whether this pattern is seen equally across all influenza proteins. The second question is whether, in spite of the rapid evolution of surface residues in influenza proteins, there are any "extreme" protein regions that are fully conserved. To answer these questions, our approach integrated the data from evolutionary genomics, structural bioinformatics, and deep sequencing. The automatic pipeline we developed (

Figure **1**) has allowed for the first time to detect statistically significantly conserved regions in the entire influenza proteome that are structurally connected but may not necessarily be sequentially contiguous. The pipeline is readily available to study proteomes of other viral families.

### 2.3.2 Evolutionary dynamics of H1N1 and our hypothesis

It was recently shown that reassortment with swine strains resulted in nearly identical regions of conserved antigenic residues in HA protein of the 1918 and 2009 H1N1 strains [65, 71]. However, that conservation is in striking contrast to the 50% sequence divergence between strains from 2007 and the 1940's [65] and appears the result of the replacement of H1N1 genes from the human strains with those from swine strains, which are much more slowly evolving in their protein sequences [8]. This combination of rapid evolution and reassortment is the principal reason for the lack of conserved regions around the HA antigenic sites, when considering H1N1 strains of different years. The result points to a more general conclusion: the evolutionary conserved surface regions, should any exist, are unlikely to occur in the regions mediating the viral-host interactions, for which the host proteins may be subject to selection against viral replication. Indeed, host-viral interactions may give rise to Red-Queen/arms-race type dynamics [72].

### 2.3.3 Insights to obtained exterior-to-interior evolutionary rates across different proteins

In addition to confirming a higher rate of evolution on the surface of viral proteins when compared to the interior, our phylogenetic study revealed signals of viral reassortment in influenza strains from other hosts [4, 73];. As a result, each protein has a unique gene-tree topology (although we did not assess the phylogenetic uncertainty inherent in these trees, since the tree inference was not a primary goal of our study). The source of the variation in exterior-to-interior residue rate ratios ($r_e/r_i$) is less straight-forward to explain. While most values were between 1.1 and 1.5, PA (1.9), M1 (2.2), and M2 (2.3) were significantly higher.  One possible reason is that PA and M2 were both incomplete structures, thus residues that are buried in the full structure could be assigned as "exterior" residues.  Thus, structural data for M2 was limited to the helix-linker-helix structural fragment of the transmembrane region, resulting in 33 “exterior” residues and only 5 “interior” residues, even though all of these residues would be buried in a membrane *in vivo*.

### 2.3.4 Structure-based phylogenetic analysis provides insights into the multi-species evolution of H1N1 virus

Using structure-driven phylogenetic analysis, we found that the human lineage of HA and NA phylogenetic trees of the H1N1 virus had a trunk-like structure while swine and avian lineages did not, indicating that the topological diversities of phylogenetic trees for H1N1 viral proteins can reflect the difference of selective pressures in human and animals. Indeed, due to a longer life span and fewer limitations on geographical barriers, the human influenza virus can be further exposed to herd immunity. As a result, one strain can be easily circulated globally. On the other hand, multiple sublineages of influenza viruses can be co-circulating in different and geographically separated animal populations. In contrast to the surface proteins, the human lineage of internal H1N1 proteins, *e.g.* PA, do not have trunk-like structures. This is likely due to the frequent reassortments [58, 59], and these proteins can have different animal origins and evolutionary histories.

The fact that there are viruses from multiple hosts located at the same lineage indicates frequent bi-dimensional transmission of influenza A viruses at human-animal, and animal-animal interfaces. For example, Mexico/2009, Iowa/2005, and New Jersey/1976 are three well-documented swine-origin influenza A viruses [74-76]. Nevertheless, the comparative analysis of the structural patterns in the phylogenetic trees of individual proteins suggests that these reassortments were different in their nature: for HA, all three strains are clustered together within North American swine lineage; for NA, Iowa/2005 and New Jersey/1976 strains are clustered with North American, while Mexico/2009 is clustered with European clade; finally for PA, Iowa/2005 and Mexico/2009 are clustered with a larger clade that includes avian and European swine lineages, while New Jersey/1976 is clustered together with other human strains.

An interesting feature of the human lineage is that the surface-to-core ratios of HA and NA proteins have increased significantly since 1984 (Figure 2). Such increase could be due to H1N1-specific herd immunity from accumulating infections of H1N1 since 1977 as well as vaccine-derived immunity, as the first nation-wide vaccination was introduced in the U.S. at the end of 1976 [77, 78]. This behavior was observed only among the surface proteins HA and NA and not internal proteins, presumably because HA and NA are the primary target of the immunological system.

Finally, when comparing the surface-to-core rates between Eurasian and North American swine lineages, two differences were noticed. The first difference, the fact that Eurasian swine lineage is clustered together with the avian lineage, while North American swine lineage is not, can be explained by the well-documented multiple transmission events of the avian H1N1 virus to pigs in continental Europe and later in Asia [79-81]. The second difference, the consistently higher surface-to-core ratios in Eurasian swine lineage, compared to the North American lineage, has not been previously reported. One explanation may be that unlike the classical swine flu in North American lineage, the swine influenza virus from the European lineage, once transmitted from the avian host, required fast adaption to the swine host. In addition, the rate difference may be associated with the suggested difference in epizootiology between the U.S. and European swine influenza, since in Europe herds may harbor the virus while showing no clinical symptoms [82, 83]. A further analysis with a more detailed reassortment history between the avian and swine lineages may be required to confirm this hypothesis.

We note that sampling bias of the strains could also be a factor influencing in our analysis, since it is one of the most common problems in influenza sequence analysis in general. For instance, the most diverse of large clusters of similar strains defined during the redundancy removal is likely to have more random mutations than those of small clusters. Thus, the higher $r_e/r_i$ ratio of the Eurasian swine lineage compared with the North American lineage could be a byproduct of sampling bias. Unfortunately, sampling bias is difficult to avoid, so one should be cautious not to over-interpret the changes of $r_e/r_i$ ratio over time in such cases. To handle sampling bias, several approaches could be explored in the future. For instance, one could look at the correlation of the $r_e/r_i$ ratios of strains with the number of redundant strains they represent or at the average values of $r_e/r_i$ ratio per year versus number of samples in the dataset before and after redundancy removal for the same year.

### 2.3.5 Effects of positive and negative selection on the protein surface in H1N1 proteins

While all of the virus proteins are subject to evolutionary change, the extent to which each protein allows certain changes depends on several factors such as location of the protein in the virion, the protein's function, and the fact that some genes are encoded on the same genomic segment. For instance, HA is expressed on the surface of the virion, is involved in host binding, and is located on its own gene segment [4]. Thus, HA is subject to a stronger selective pressure compared to the internal proteins, such as M1, which serves a structural purpose as well as RNA binding, and shares a coding region with M2 [4]. Because of this shared coding region, each mutation risks causing a detrimental change in the other gene. There is also variation within a given protein: HA's antigenic sites are subject to positive selection due to host immune pressure, yet the stem region is subject to purifying selection due to its role in trimer formation. This mixed selection is seen in essentially all of the proteins: there exist regions that are subject to positive selection due to their role in viral-host interactions and there exist regions that are subject to negative selection due to their role in intra-viral interactions.

### 2.3.6 High conservation of H1N1 functional regions have been previously reported

There have been several studies that have found high but not necessarily 100% conserved regions on the surfaces of the influenza proteins. For instance, it has been found that the dsRNA binding track of NS1 consists of conserved binding residues [84]. Additionally, the conservation of the surface regions has been determined near the stem region of HA protein, [85]. Since HA evolves considerably faster than NS1, it is of note that both of these structures are known to have conserved binding regions. The regions found in the present study overlapped with regions identified, experimentally, to be conserved but did not overlap with them entirely.

### 2.3.7 Analysis of extremely conserved regions

In concert with the above findings, we found that all of the conserved regions detected were associated with the intra-viral macromolecular complexes, including protein homomers, heteromers, or protein-viral RNA interactions. Interestingly, each region covered a part of, but never the entire, binding site. This type of co-localization suggests that though most of an intra-viral binding site is conserved, variable residues exist perhaps under weaker selective pressure than their conserved neighbors. In the case of M1, NP, and NS2, the conserved regions are co-localized with multiple binding sites. Note that each of these interactions buries the exposed residues of conserved regions in the interaction interface, effectively making them the interior residues. However, while some interactions are more long-term than others, none remain bound for the entire viral life cycle. In contrast to the situation with host-viral interactions, natural selection is expected to stabilize intra-viral interactions [86], which accounts for their conservation. Alternatively, there could be co-evolution between the interacting residues, such as found in some host-viral interactions [87, 88]. While each significant region has been associated with at least one known functional region, there are portions of each region that do not overlap with any functional sites. Those regions may be involved in undiscovered intra-viral interactions.  This hypothesis is plausible, given that very few known interactions have been comprehensively characterized on the residue level. The geographic scale and time scale, together with the degree of observed extreme conservation in the influenza proteins allows one to suspect that these conserved regions would also occur across viral strains in any given year. Consistent with this hypothesis, our mapping of genetic variation obtained from an individual carrying three genetic variants from two distinct phylogenetic clades did not find a single mutation in any of the conserved regions. However, further studies involving multiple subjects and larger viral populations are necessary to provide a stronger linkage between the temporal and population-wise conservation of the functional regions in influenza proteins.

### 2.3.8 Our findings may provide insights into new influenza drug targets

Attaining total protection against influenza A virus through the development of universal antivirals and vaccines has been a challenging task due to the increasing resistance to the treatments of new viral strains as well as the enormous diversity of the viral population. Recently, a number of promising approaches have been identified, including human monoclonal antibodies and antivirals inhibiting the activity of influenza proteins. Both vaccines and antivirals are capable of neutralizing a wide range of influenza A and often B strains [89-95], but they have been focused thus far on only a few protein targets: the vaccines for HA and antivirals for M2 and NA. Moving beyond these targets, the design of new protein inhibitors of influenza polymerase has been recently suggested as a potential direction in the development of new antivirals due to its high conservation and significance to viral function [96]. Our study may provide further insight towards identifying new protein targets for influenza antivirals or antibodies, pinpointing the key binding regions that are conserved across a wide range of current and past influenza strains and thus likely to be preserved in future strains. One example from our data is the PB1 to PB2 interaction, which, if disrupted, could result in the loss of viral RNA replication function [97]. One of the main challenges in targeting the regions of extreme conservation, however, comes from their intrinsic property: the regions become inaccessible upon intra-viral macromolecular interactions. Understanding the dynamics of such interactions may provide further insight into this challenge as well as the evolutionary mechanisms behind the extreme conservation.

# 3 Evolutionary Patterns of Pandemic Influenza

After the 2009 H1N1 influenza pandemic, a large increase in research concerning pandemic influenza and potentially pandemic subtypes began. As a result, there became a need for an analysis of the overall pattern of evolution across all pandemic subtypes: H1N1, H2N2, H3N2, and H5N1. Influenza researchers tend to focus on only one subtype and rarely perform inter-subtype analyses. Additionally, research is generally done on only conservation or diversity and a full-scale analysis of influenza evolution has yet to be performed. To understand the evolutionary pattern of the protein surfaces of pandemic influenza, I computed the extremely conserved and diverse residues for each protein and subtype via sequence alignment. As in section 2, I mapped these residues to the protein surfaces. Unlike my previous analysis, however, I used DBSCAN clustering rather than graph-based clustering (to improve intra-cluster connectivity) for conserved residues and developed a Metropolis Criterion Monte Carlo Clustering (MMC) method to cluster the diverse residues. Thus providing an overall pattern of evolution across the protein surfaces for each subtype, enabling an analysis across all pandemic subtypes.

## *3.1 Methods*

### 3.1.1 Data sources and preprocessing

Influenza sequences were gathered from the NCBI Influenza sequence database. All incomplete and highly similar sequences from H1N1, H2N2, H3N2, and H5N1 were then filtered out to remove redundancy. Two sequences are considered redundant if they (i) are more than 95% identical, (ii) are from the same year, and (iii) have the same host organism. Protein models were constructed for each of the ten proteins of each of the subtypes. These models were created using MODELLER with templates listed in Table 9.

### 3.1.2 Definition of conservation and diversity

Before we can assign a conservation or diversity measure to a given residue position, we had to first align the sequences with MUSCLE [98]. We defined conservation in a binary fashion: either a residue is 100% conserved or it is not. Diversity is not this straight-forward and we tested several diversity measures. First we considered percent identity, but this value failed to capture the effect of different mutation types. To solve this we tried using Shannon Entropy, which takes different mutation types into account, but fails to recognize the relationships between the mutation types. For example, a change from leucine to isoleucine is minimal while a change from glycine to tryptophan is quite substantial. We then decided to use a measure referred to as $C_{trident}$ as described by Valdar [99].

$$C_{trident} = (1 - r(x))^{\alpha}(1 - t(x))^{\beta}(1 - g(x))^{\gamma}$$

Where *t(x)* is a scaled Shannon Entropy, *r(x)* corresponds to the types of residue changes, and *g(x)* is the percentage of gaps. The parameters α, β, and γ can be used to weight one portion more heavily. For this analysis, we set each parameter to one. Each of these functions (displayed in Figure 6) is scaled by a separate factor, λ, to make the value fall in the range [0, 1].

### 3.1.3 Clustering of conserved residues

We began by using MolMol[27] to determine which residues are on the surface. We defined an external residue as any residue such that a single atom was more than 5% exposed to match the definition used by MODELLER to generate random regions (see section 3.1.4). From this list of exterior residues, we selected only those that were 100% conserved ($C_{Trident}$ = 1) across all sequences of that subtype. We then used Xwalk [100] to determine the solvent accessible surface distance (SASD) between each residue in order to make a distance matrix of all conserved exterior residues for each protein model of each subtype. We chose to determine SASD rather than using only Euclidian distance, as we had before, because SASD will detect residues that may be spatially close, but are on opposite sides of the protein (as on M2) or on distinctly different regions of the protein (as on the flexible loop of NP). This procedure resulted in a more realistic representation of the protein surfaces.

Now that we had pairwise distance matrixes, we needed to cluster them (Figure 5). Our previous approach was too slow (O($n^3$) complexity) and failed to take density of patches into account. When we did not take density into account, we ended up with patches that were somewhat *stringy*, meaning that they did not look compact in a way that you would expect a binding site to look. By taking density into account, we would be able to take a long *snake-like* patch and break it into several smaller dense patches. To do this we used DBSCAN (Density Based Spatial Clustering of Application with Noise) [17]. DBSCAN is a density-based clustering algorithm that can take inputs of several forms (see section 3.1.4.1.2). We chose to use our previously computed distance matrices as input. DBSCAN, unlike k-means, does not require that the number of clusters be specified in advance and does not require a vector representation for each data point, both features of which are better fits to our data. In other words, if we only represented the residues as a vector of the x,y,z coordinates, then we would fail to take the SASD into account.

Since DBSCAN is nondeterministic, we ran it 100 times for each distance matrix. We found that 100 iterations was sufficiently large to cover data space without being computationally wasteful. These runs naturally resulted in 100 different clusterings. We decided to use the Minkowski distance as the criterion for the best clustering.

$$M = \left( \sum_{j=1}^{N-1} \sum_{i=j}^{N} d_{ij}^{4} \right)^{1/4}$$

We chose this measurement because it is commonly used in clustering. We determined the Minkowski distance for each patch and then found the average for that clustering. The clustering with the smallest Minkowski distance was assumed to be the best. Even with the requirements listed above, this clustering protocol ran far faster than our previous naive clustering.

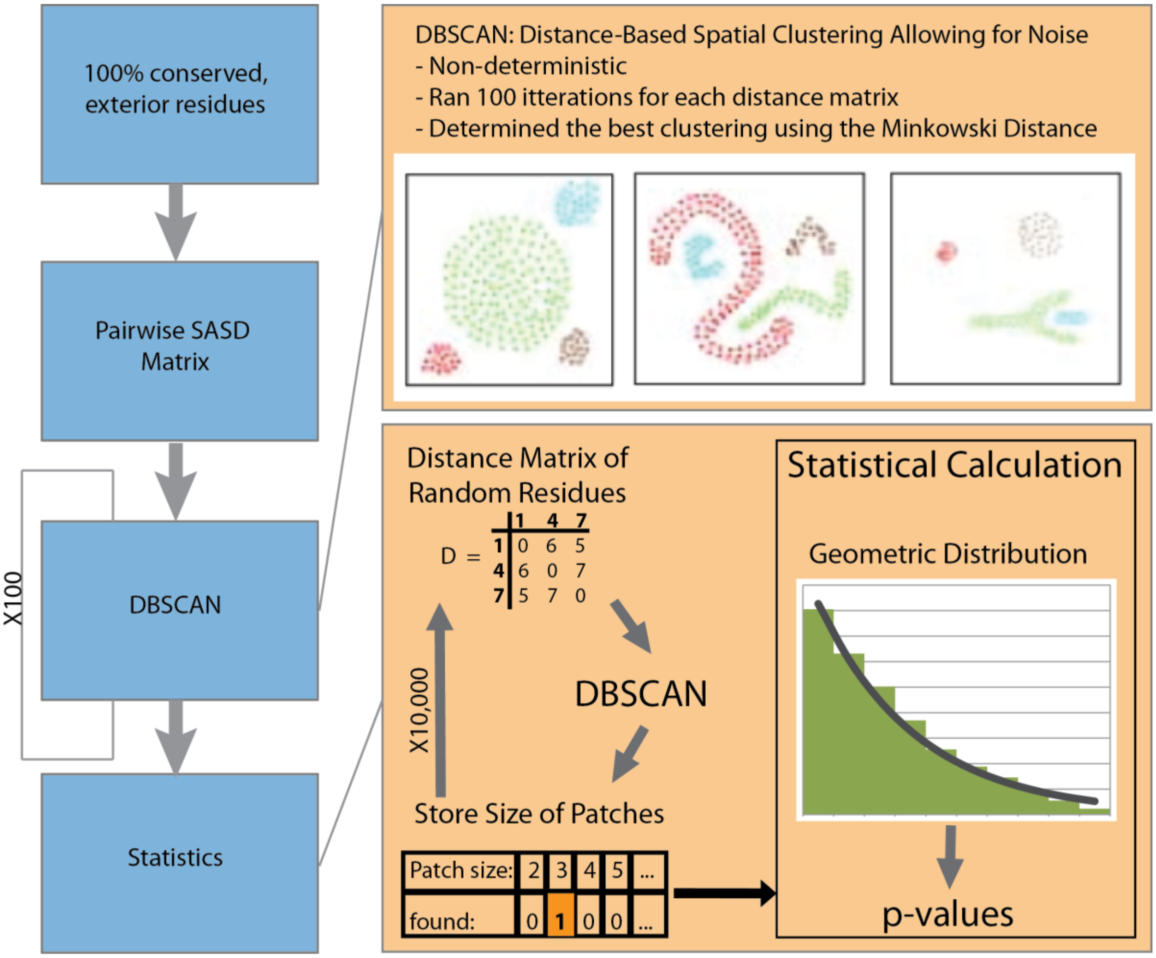

100% conserved, exterior residues
Pairwise SASD Matrix
X100
DBSCAN
Statistics
DBSCAN: Distance-Based Spatial Clustering Allowing for Noise
- Non-deterministic
- Ran 100 itterations for each distance matrix
- Determined the best clustering using the Minkowski Distance
Distance Matrix of Random Residues
D =
1 4 7
1 0 6 5
4 6 0 7
7 5 7 0
X10,000
DBSCAN
Store Size of Patches
Patch size: 2 3 4 5 ...
found: 0 1 0 0 ...
Statistical Calculation
Geometric Distribution
p-values

**Figure 5|Clustering pipeline of conserved residues.**
Here we outline the steps used in determining extremely conserved residues. We begin by giving examples of the usage of DBSCAN and continue by explaining the method by which we determine the statistical significance of each cluster.

Using the Minkowski distance to choose the best possible clustering does not ensure that our clusters were statistically significant. Instead we required a statistical method make this determination. We began by creating a random distribution of patches. We did so by creating distance matrices for each protein structure that was of the same size as the original matrix (i.e., the same number of random residues as there were conserved exterior residues), then performing our DBSCAN clustering protocol on the distance matrix of random residues. We performed 10,000 iterations of this randomization procedure to create a distribution of patches of various sizes. As we previously discovered, these patches fall into a discrete geometric distribution. Using this distribution we determined the p-value for each patch size.

### 3.1.4 Clustering of diverse residues

Clustering diverse residues (Figure 6) was a far more difficult task than clustering conserved residues. Conservation is a binary state: either something is or is not 100% conserved. Diversity has no such binary property. Even with a good measure of diversity ($C_{Trident}$), there is a varying level of diversity ranging from 0 to 1. It would be inaccurate to set a diversity threshold T and assume that any residue with $C_{Trident} < T$ could be defined as diverse and then incorporated into a distance matrix as above. The results of setting such a threshold are shown in section 3.1.4.1 using a variety of state-of-the-art clustering methods.

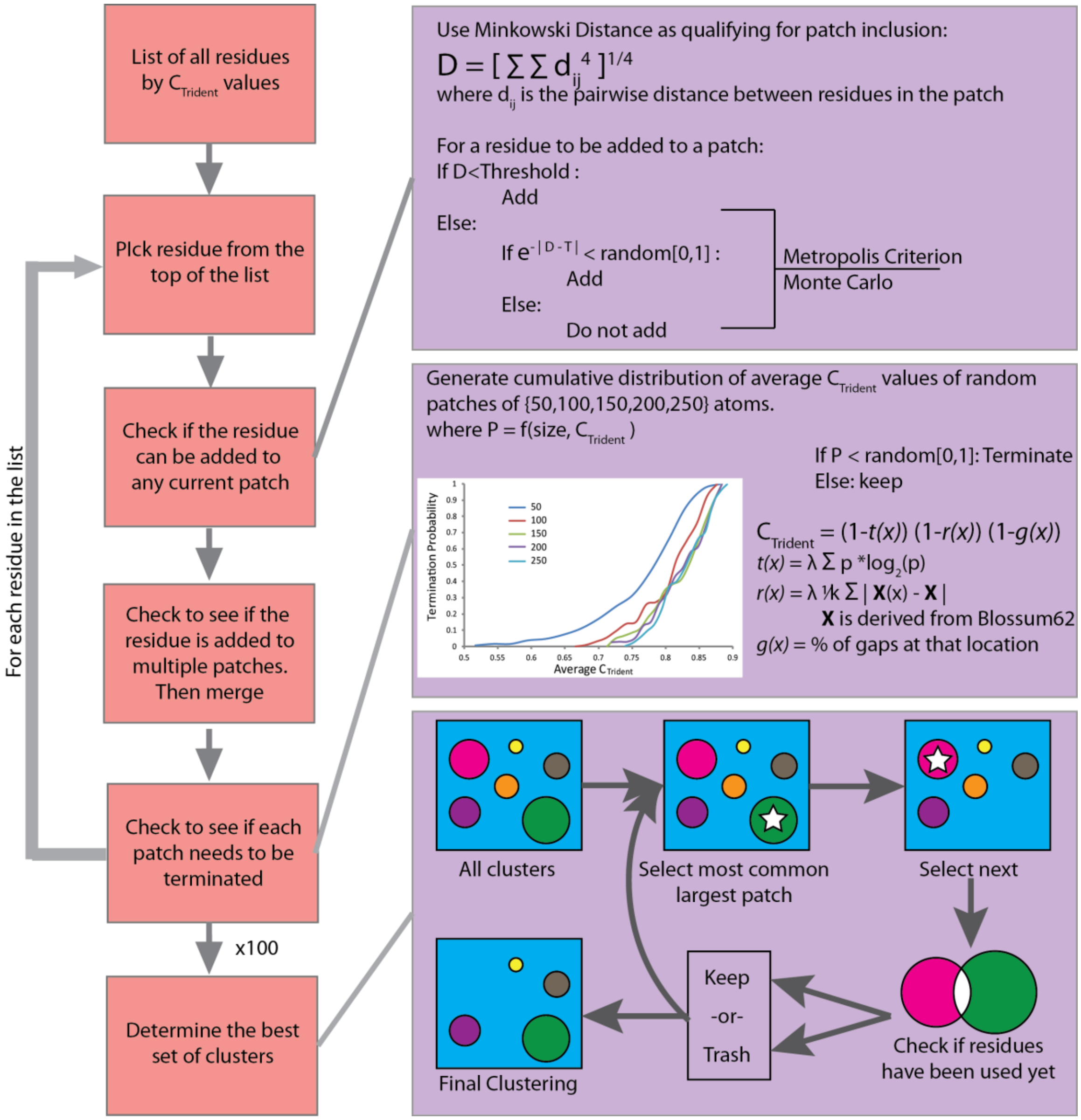

List of all residues by $C_{Trident}$ values
PIck residue from the top of the list
Check if the residue can be added to any current patch
Check to see if the residue is added to multiple patches. Then merge
Check to see if each patch needs to be terminated
x100
Determine the best set of clusters
For each residue in the list
Use Minkowski Distance as qualifying for patch inclusion:
$D = [\sum\sum d_{ij}^{4}]^{1/4}$
where $d_{ij}$ is the pairwise distance between residues in the patch
For a residue to be added to a patch:
If D<Threshold :
Add
Else:
If $e^{-|D-T|}$ < random[0,1] :
Add
Else:
Do not add
Metropolis Criterion
Monte Carlo
Generate cumulative distribution of average $C_{Trident}$ values of random patches of {50,100,150,200,250} atoms.
where $P = f(size, C_{Trident})$
If P < random[0,1]: Terminate
Else: keep
Termination Probability
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
Average $C_{Trident}$
50
100
150
200
250
$C_{Trident} = (1-t(x)) (1-r(x)) (1-g(x))$
$t(x) = \lambda \sum p * \log_2(p)$
$r(x) = \lambda \frac{1}{k} \sum | \mathbf{X}(x) - \mathbf{X} |$
**X** is derived from Blossum62
$g(x)$ = % of gaps at that location
All clusters
Select most common largest patch
Select next
Check if residues have been used yet
Keep -or- Trash
Final Clustering

**Figure 6|Clustering of Diverse Residues.**
Here we highlight the steps of our MMC method: from the labeling of residue diversity, to statistical analysis, to determining the ideal set of clusters.

We decided, instead, to create an ascending order sorted list of all residues by their $C_{Trident}$ values, then iterate through that list form most diverse, to least (0 to 1). We began by taking the most diverse residue and creating a patch from it. We then took the next in the list and determined if it could be added in a patch to the existing residue, or if it needed to be added to its own patch. We then did this for each surface residue on the list.

To determine if a residue could be added to a patch we used a Metropolis Criterion Monte Carlo (MMC) method (detailed information in section 3.1.4.1.4). We used Minkowski distance (see section 3.1.3) as our criterion for addition. We determine Minkowski distance D of the patch with the residue added. If D is less than a threshold T, then the residue is added to that patch. If not, then we would determine the probability that the residue should be added using the Metropolis Criterion:

$$p = e^{-(|D-T|)}$$

Using this probability we used a Monte Carlo simulation to determine if the residue could be added. We selected a random number R between 0 and 1. If R<p, then we would add the residue to that patch.

Once we had finished attempting to add a residue to the existing patches we would determine if the patch had been added to multiple patches. If so, then we would attempt to merge those patches. To merge two patches, we would determine the Minkowski distance of the merged patches and accept the merge based on the MMC method described above. If the patches were not merged, then one would be terminated, namely the one with the highest Minkowski distance. When a patch is terminated, it simply means that no more residues can be added to it. It is considered a *mature* patch.

After a residue had been checked against all patches and any merging operations had been executed, we determined which patches, if any, should be terminated. This computation was done using a Monte Carlo method with termination probabilities based on cumulative distributions of random surface patches (described later). If a patch's termination probability was greater than a given random number, then that patch was terminated.

To determine these termination probabilities we created random surface patches of with {50, 100, 150, 200, 250} atoms using MODELLER. This method generated patches that were very compact, which is what we were expecting from our final patches. We then determined the average $C_{Trident}$ value of all residues involved with these constructed patches. We kept the $C_{Trident}$ values of each of these patches and generated a distribution for each patch size. These distributions did not appear to follow a standard distribution: they were most similar to multimodal Gaussian distributions. We chose, instead, to generate cumulative distributions. We then scaled the y-axis of these distributions to a maximum value of 1. This procedure resulted in our termination probabilities for each patch size.

This diversity clustering method is nondeterministic, so it had to be run 100 times for each protein model. This approach resulted in several clusterings. As before, we had to determine which clusters to keep and which to discard. We did so by making a list of all possible clusters ordered by size and frequency, in that order of importance. We began by taking the largest most common cluster. We then took the next in the list and kept it only if none of the residues had been used in previous patches. This approach proved to be an effective method for determining the ideal cluster set for each protein model.

#### 3.1.4.1 Analysis of clustering algorithms for diverse residues

To benchmark my Metropolis Criterion Monte Carlo clustering method (detailed in section 3.1.4.1.4), I compared it to five other clustering algorithms: K-means, DBSCAN (used for conserved residues), and Graph-based Connectivity (detailed in section 2.1.4.1). I selected these algorithms because they are the current state-of-the-art methods that are most appropriate for my data. The difficulty of this analysis came from determining a threshold for diversity. The $C_{trident}$ measure scales from 0 (most diverse) to 1 (100% conserved), but there is no defined cutoff to define conservation versus diversity. Additionally, each protein has varied level of diversity: HA is the most diverse (highest $C_{trident}$ average) and M1 is the least diverse (lowest $C_{trident}$ average). $C_{trident}$ values vary depending on subtype.

To analyze the effectiveness of each clustering algorithm I performed clustering of the diverse residues of HA and M1. These two proteins were chosen because they yielded the most extreme cases of diversity and conservation, respectively. Since it is not possible to pick a specific threshold, I made four datasets, each with their own threshold. The threshold was decided as $T(p) = (Max - Min)p + Min$, where $p = \{0.2, 0.3, 0.4, 0.5\}$ and *Max* and *Min* refer to the maximum and minimum values of surface residue $C_{trident}$ values for that protein/strain combination. Because *Max* = 1 for all combinations, *T(p)* can be simplified to $T(p) = p + (1 - p)Min$ .

In essence, *T(p)* determines a threshold which covers 20%, 30%, 40%, or 50% of the gap between the most diverse and least diverse surface residue. Because the $C_{trident}$ values are not uniformly distributed, covering, for example, 10% of the difference between the

*Min* and *Max* covers less than 10% of the total residues. Information about the datasets can be found in Table 4.

| Protein | Subtype | # surface resides | *p* | Threshold | # residues |
|---|---|---|---|---|---|
| HA | H1N1 | 399 | 0.2 | 0.4184 | 11 |
| | | | 0.3 | 0.4911 | 28 |
| | | | 0.4 | 0.5638 | 60 |
| | | | 0.5 | 0.6365 | 85 |
| | H2N2 | 399 | 0.2 | 0.5265 | 8 |
| | | | 0.3 | 0.5857 | 13 |
| | | | 0.4 | 0.6449 | 25 |
| | | | 0.5 | 0.7041 | 37 |
| | H3N2 | 406 | 0.2 | 0.4066 | 5 |
| | | | 0.3 | 0.4808 | 15 |
| | | | 0.4 | 0.5549 | 38 |
| | | | 0.5 | 0.6291 | 65 |
| | H5N1 | 403 | 0.2 | 0.4460 | 4 |
| | | | 0.3 | 0.5152 | 9 |
| | | | 0.4 | 0.5845 | 22 |
| | | | 0.5 | 0.6537 | 59 |
| M1 | H1N1 | 129 | 0.2 | 0.6814 | 10 |
| | | | 0.3 | 0.7212 | 12 |
| | | | 0.4 | 0.7611 | 13 |
| | | | 0.5 | 0.8009 | 15 |
| | H2N2 | | 0.2 | 0.6063 | 3 |
| | | | 0.3 | 0.6555 | 3 |
| | | | 0.4 | 0.7047 | 6 |
| | | | 0.5 | 0.7539 | 8 |
| | H3N2 | 124 | 0.2 | 0.6718 | 4 |
| | | | 0.3 | 0.7128 | 8 |
| | | | 0.4 | 0.7538 | 12 |
| | | | 0.5 | 0.7949 | 15 |
| | H5N1 | 126 | 0.2 | 0.6820 | 6 |
| | | | 0.3 | 0.7218 | 10 |
| | | | 0.4 | 0.7615 | 12 |
| | | | 0.5 | 0.8012 | 17 |

**Table 4| Benchmarking datasets.**
Each dataset was generated from $C_{trident}$ values of surface residues. Each clustering algorithm was then performed on these datasets.

### *3.1.4.1.1 K-means*

K-means clustering is one of the most commonly used clustering algorithms due to the algorithm's simplicity and scalability. K-means can easily handle thousands of data points. The algorithm has been implemented in nearly every language: python (by scikit-learn), MATLAB, Java, and C. K-means takes two inputs: (i) data as n-dimensional vectors and (ii) k – the number of clusters. This algorithm is appropriate if you know the number of clusters in your data, you have high dimensional data, you have a very large number of clusters, and/or your clusters are likely to be compact and spherical. K-means tends to perform poorly for elongated or non-spherical shaped clusters and lacks the sensitivity for a small number of data points.

The K-means algorithm works by first assigning *k* centers that will serve as the initial mean values μ. These centers are determined in a few different ways depending on implementation. The simplest assignment is to pick *k* random data points to serve as your initial centers. This approach, however could result in centers that belong to the same cluster. To reduce the chance of this problem occurring, centers can be selected not from the data, but rather from evenly distributed portions of the data space. This method tends to a popular way to pick data, though it requires more initial computation.

Once the *k* centers are defined, each data point is initially added to the *k* closest to it. Once all of the data points have been added assigned to cluster, the μ for each cluster is then updated. At this point, data points are again assigned to the nearest center. These two steps – assignment and evaluation – are repeated until the assignments become stable. The primary downfall to this process is the possibility of reaching a sub-optimal solution. Running the algorithm multiple times with different initial centers greatly minimizes this risk, but results in a much greater runtime.

To use this algorithm on my data the algorithm had to be altered, a common practice in clustering. Since my data is represented as a distance matrix and not as vectors, the mean calculation and reassignment paradigm didn’t work. I, instead, ran the non-deterministic version of the K-means algorithm which works by generating many different sets of centers and assigning data points this way. The difficulty here is determining the best clustering – a general problem in clustering. To assess each possible clustering, I calculated the inertia for each cluster:

$$I(C_i) = \sum_{x \in C_i} dist(x, k_i)^2$$

Where $k_i$ is the chosen center of $C_i$. The clustering with the minimum average inertia was selected as the final clustering.

Additionally, for K-means to work, the user must input the desired number of clusters. Since my clustering was discovery-driven, I did not know how many clusters to expect. This meant that I needed to try several values of *k*. I selected values of *k* such that $2 \leq k \leq \left\lfloor {}^{datasize}/_{2} \right\rfloor$ and perform clustering for each *k*. From there, I selected the value of *k* that resulted in the lowest average inertia. This rule favored larger values of *k*, but despite this, many of the clusters are considered outliers – clusters containing only one item – meaning that the number of meaningful clusters was typically less than the value of *k*. Overall results of K-means clustering are shown in Table 5.

| protein | p | num_clusters | C_trident | Minkowski | outliers |
|---|---|---|---|---|---|
| H1N1_HA | 0.2 | 5 | 0.36 | 14.31 | 1 |
| H1N1_HA | 0.3 | 10 | 0.42 | 5.89 | 3 |
| H1N1_HA | 0.4 | 17 | 0.46 | 4.46 | 4 |
| H1N1_HA | 0.5 | 20 | 0.50 | 4.78 | 2 |
| H1N1_M1 | 0.2 | 5 | 0.65 | 20.00 | 4 |
| H1N1_M1 | 0.3 | 6 | 0.67 | 16.67 | 5 |
| H1N1_M1 | 0.4 | 6 | 0.68 | 16.67 | 5 |
| H1N1_M1 | 0.5 | 7 | 0.68 | 14.29 | 6 |
| H2N2_HA | 0.2 | 4 | 0.46 | 8.18 | 1 |
| H2N2_HA | 0.3 | 6 | 0.49 | 4.79 | 3 |
| H2N2_HA | 0.4 | 9 | 0.53 | 7.35 | 2 |
| H2N2_HA | 0.5 | 11 | 0.58 | 6.09 | 2 |
| H2N2_M1 | 0.2 | 1 | 0.53 | 100.00 | 0 |
| H2N2_M1 | 0.3 | 1 | 0.53 | 100.00 | 0 |
| H2N2_M1 | 0.4 | 3 | 0.60 | 19.60 | 2 |
| H2N2_M1 | 0.5 | 4 | 0.63 | 7.18 | 2 |
| H3N2_HA | 0.2 | 2 | 0.35 | 8.14 | 1 |
| H3N2_HA | 0.3 | 6 | 0.40 | 5.88 | 1 |
| H3N2_HA | 0.4 | 10 | 0.48 | 6.63 | 0 |
| H3N2_HA | 0.5 | 15 | 0.50 | 5.85 | 3 |
| H3N2_M1 | 0.2 | 2 | 0.64 | 8.10 | 1 |
| H3N2_M1 | 0.3 | 4 | 0.66 | 7.79 | 1 |
| H3N2_M1 | 0.4 | 6 | 0.69 | 6.38 | 2 |
| H3N2_M1 | 0.5 | 7 | 0.70 | 9.40 | 1 |
| H5N1_HA | 0.2 | 2 | 0.40 | 37.68 | 1 |
| H5N1_HA | 0.3 | 4 | 0.44 | 9.14 | 1 |
| H5N1_HA | 0.4 | 9 | 0.49 | 8.94 | 1 |
| H5N1_HA | 0.5 | 17 | 0.58 | 7.23 | 1 |
| H5N1_M1 | 0.2 | 3 | 0.65 | 7.48 | 1 |
| H5N1_M1 | 0.3 | 4 | 0.67 | 8.34 | 0 |
| H5N1_M1 | 0.4 | 5 | 0.67 | 3.05 | 2 |
| H5N1_M1 | 0.5 | 6 | 0.70 | 3.95 | 1 |

**Table 5|K-means Clustering.**
Above are the results of K-means clustering including $C_{trident}$ value, Minkowski distance, and cluster size. The value of *p* that offers the minimum Minkowski distance is selected as the ideal threshold and is in red.

### *3.1.4.1.2 DBSCAN*

DBSCAN (Density Based Spatial Clustering of Applications with Noise) is a more advanced algorithm that generates clusters based on density and not just distance. DBSCAN was the second (published only months after OPTICS) of the density based clustering algorithms that account for noise and remains popular today due to its low run time and sensitivity for smaller datasets. In my analysis of extremely conserved regions (see section 3.1.3), I chose DBSCAN for those reasons, also that DBSCAN is implemented in python by scikit-learn. For input DBSCAN takes an association matrix, a minimum points, and a maximum distance. Using these parameters, DBSCAN can easily be tuned to fit most needs. Because it is a density-based method, DBSCAN is excellent at finding irregularly shaped clusters and can distinguish between clusters with poor boundaries. The primary difficulty of DBSCAN is that it is non-deterministic, so it must be run several times (I selected 100 iterations). This fact means that a user needs to decide how to evaluate the clustering in order to determine the ideal clustering. For my data, the Minkowski distance was appropriate – it is also a standard for evaluating clustering. Also, determining the threshold for minimum points and maximum distance can be difficult depending on the type of data.

DBSCAN begins by generating a random seed within the data space and finding all points that are *density reachable*. For point *p* to be density reachable from *q*, we must be able to draw a series of circles of radius *max_distance* which contain at least *min_points* such that each subsequent circle is drawn with the center as a point within the previous circle until a circle contains *q*. Just because *q* is density reachable from *p* does not mean that *p* is density reachable from *q*. Any given cluster will contain points that are density reachable from the seed. Seeds are generated until all data points have been reached. If a point is not density reachable from any other point, then that point will be assigned as noise. Also, if a point is density reachable from multiple clusters, but does not bridge the two clusters, then that point is also assigned as noise.

| protein | p | num_clusters | C_trid | Minkowski | outliers |
|---|---|---|---|---|---|
| H1N1_HA | 0.2 | 1 | 0.28 | 2.40 | 0 |
| H1N1_HA | 0.3 | 4 | 0.43 | 3.53 | 0 |
| H1N1_HA | 0.4 | 4 | 0.47 | 3.76 | 0 |
| H1N1_HA | 0.5 | 10 | 0.47 | 5.24 | 0 |
| H1N1_M1 | 0.2 | 1 | 0.65 | 100.00 | 0 |
| H1N1_M1 | 0.3 | 1 | 0.65 | 100.00 | 0 |
| H1N1_M1 | 0.4 | 0 | 0.00 | 0.00 | 0 |
| H1N1_M1 | 0.5 | 0 | 0.00 | 0.00 | 0 |
| H2N2_HA | 0.2 | 1 | 0.45 | 4.10 | 0 |
| H2N2_HA | 0.3 | 2 | 0.50 | 6.43 | 0 |
| H2N2_HA | 0.4 | 5 | 0.53 | 4.77 | 0 |
| H2N2_HA | 0.5 | 6 | 0.57 | 5.26 | 0 |
| H2N2_M1 | 0.2 | 1 | 0.51 | 100.00 | 0 |
| H2N2_M1 | 0.3 | 1 | 0.51 | 100.00 | 0 |
| H2N2_M1 | 0.4 | 1 | 0.58 | 75.28 | 0 |
| H2N2_M1 | 0.5 | 0 | 0.00 | 0.00 | 0 |
| H3N2_HA | 0.2 | 0 | 0.00 | 0.00 | 0 |
| H3N2_HA | 0.3 | 2 | 0.37 | 6.22 | 0 |
| H3N2_HA | 0.4 | 3 | 0.43 | 5.67 | 0 |
| H3N2_HA | 0.5 | 11 | 0.48 | 4.54 | 0 |
| H3N2_M1 | 0.2 | 0 | 0.00 | 0.00 | 0 |
| H3N2_M1 | 0.3 | 1 | 0.66 | 22.36 | 0 |
| H3N2_M1 | 0.4 | 0 | 0.00 | 0.00 | 0 |
| H3N2_M1 | 0.5 | 1 | 0.70 | 3.10 | 0 |
| H5N1_HA | 0.2 | 0 | 0.00 | 0.00 | 0 |
| H5N1_HA | 0.3 | 2 | 0.40 | 3.40 | 0 |
| H5N1_HA | 0.4 | 2 | 0.49 | 2.88 | 0 |
| H5N1_HA | 0.5 | 6 | 0.46 | 5.53 | 0 |
| H5N1_M1 | 0.2 | 1 | 0.64 | 34.63 | 0 |
| H5N1_M1 | 0.3 | 1 | 0.67 | 4.43 | 0 |
| H5N1_M1 | 0.4 | 1 | 0.68 | 8.08 | 0 |
| H5N1_M1 | 0.5 | 1 | 0.69 | 6.56 | 0 |

**Table 6|DBSCAN Clustering.**

Above are the results of DBSCAN clustering including $C_{trident}$ value, Minkowski distance, and cluster size. The value of *p* that offers the minimum Minkowski distance is selected as the ideal threshold and is in red. Here, outliers are automatically determined to be noise, thus no outliers are detected.

To cluster my data I used *min_points* = 2 and *max_distance* = 6, the same values that were used for clustering conserved residues. As with the conserved residue protocol, I repeated the clustering 100 times using the Minkowski distance as the criterion to choose the best clustering. The results of DBSCAN clustering are presented in Table 6.

#### *3.1.4.1.3 Graph-based connectivity*

Graph-based Connectivity is the Graph-based algorithm initially used to cluster extremely conserved residues (see section 2.1.4.1). Results of the clustering are listed in Table 7. It is important to notice that Graph-based clustering results in more outlier clusters than true clusters. In some cases there are no non-trivial clusters. Graph-based clustering uses a strict distance threshold as the clustering criterion, meaning that for two points to be in the same cluster they absolutely must have a distance less than the threshold. For the threshold of 6Å and very low $C_{trident}$ thresholds, there were very few, if any, residue pairs.

| protein | p | num_clusters | C_trid | Minkowski | outliers |
|---|---|---|---|---|---|
| H1N1_HA | 0.2 | 10 | 0.37 | 0.24 | 9 |
| H1N1_HA | 0.3 | 15 | 0.42 | 2.21 | 10 |
| H1N1_HA | 0.4 | 22 | 0.50 | 3.24 | 14 |
| H1N1_HA | 0.5 | 21 | 0.53 | 4.12 | 11 |
| H1N1_M1 | 0.2 | 10 | 0.65 | 0.00 | 10 |
| H1N1_M1 | 0.3 | 12 | 0.66 | 0.00 | 12 |
| H1N1_M1 | 0.4 | 13 | 0.66 | 0.00 | 13 |
| H1N1_M1 | 0.5 | 15 | 0.68 | 0.00 | 15 |
| H2N2_HA | 0.2 | 6 | 0.47 | 1.23 | 5 |
| H2N2_HA | 0.3 | 8 | 0.50 | 1.91 | 5 |
| H2N2_HA | 0.4 | 16 | 0.56 | 1.65 | 11 |
| H2N2_HA | 0.5 | 20 | 0.61 | 2.34 | 13 |
| H2N2_M1 | 0.2 | 3 | 0.53 | 0.00 | 3 |
| H2N2_M1 | 0.3 | 3 | 0.53 | 0.00 | 3 |
| H2N2_M1 | 0.4 | 6 | 0.60 | 0.00 | 6 |
| H2N2_M1 | 0.5 | 8 | 0.63 | 0.00 | 8 |
| H3N2_HA | 0.2 | 3 | 0.36 | 2.69 | 2 |
| H3N2_HA | 0.3 | 9 | 0.42 | 1.99 | 5 |
| H3N2_HA | 0.4 | 18 | 0.48 | 1.50 | 13 |
| H3N2_HA | 0.5 | 23 | 0.54 | 3.28 | 10 |
| H3N2_M1 | 0.2 | 4 | 0.64 | 0.00 | 4 |
| H3N2_M1 | 0.3 | 8 | 0.67 | 0.00 | 8 |
| H3N2_M1 | 0.4 | 11 | 0.69 | 0.28 | 10 |
| H3N2_M1 | 0.5 | 12 | 0.71 | 1.12 | 9 |
| H5N1_HA | 0.2 | 4 | 0.39 | 0.00 | 4 |
| H5N1_HA | 0.3 | 7 | 0.45 | 0.97 | 5 |
| H5N1_HA | 0.4 | 15 | 0.51 | 0.85 | 12 |
| H5N1_HA | 0.5 | 24 | 0.59 | 2.25 | 17 |
| H5N1_M1 | 0.2 | 5 | 0.64 | 1.12 | 4 |
| H5N1_M1 | 0.3 | 5 | 0.65 | 2.54 | 3 |
| H5N1_M1 | 0.4 | 5 | 0.66 | 2.15 | 3 |
| H5N1_M1 | 0.5 | 7 | 0.69 | 3.05 | 5 |

**Table 7|Graph-Based Clustering.**

Above are the results of Graph-Based clustering including $C_{trident}$ value, Minkowski distance, and cluster size. The value of *p* that offers the minimum Minkowski distance is selected as the ideal threshold and is in red. M1 for H1N1 and H2N2 have no threshold selected because there is no clustering with non-outlier clusters.

### *3.1.4.1.4 Metropolis-Criterion Monte Carlo Clustering (MMC)*

Pseudo code:

```
list <- sorted list of residues by C_trident
for i in list {
     Add(list[i]) {
          while curr !=null
               //determine if residue should be added
               Distance(list[i],curr)
          //check if any clusters need to be merged
          Merge(p1, p2)
          while curr !=null
               //check if each cluster should be terminated
               Terminate(curr)
     }
}

boolean Distance(point r, cluster p) {
     calculate Minkowski distance where
                         $mink = \sum_{i,j \in P} d_{ij}$
     accept with probability $e^{-|mink-Threshold|}$
}

String Merge(cluster p1, cluster p2) {
     while !finished
          while curr!=null
               if two clusters have the same residue
                    calculate Minkowski distance between
                    all residues in the two clusters
                         $mink = \sum_{i,j \in P} d_{ij}$
                    merge with probability $e^{-|mink-Threshold|}$
                    if not merged
                         Terminate(p1, p2)
}

String Terminate(cluster c) {
     Read in stats from file (cumulative distributions)
     //C_trident values for clusters sized 50, 100,..., 350
     terminate with probability from file
}

String Terminate(cluster p1, cluster p2)
```

Because diversity is not a binary state, unlike conservation, a threshold for $C_{trident}$ would need to be defined. Since there is no suggested or intuitive method for setting a threshold, it was most logical to generate a probabilistic method that can determine clusters based on statistical data. Thus, I developed the Metropolis Criterion Monte Carlo Clustering (MMC) algorithm (pseudo code above, outline in Figure 6). There are three main parts of MMC clustering: (i) determining termination statistics from random data, (ii) cluster data with multiple iterations, and (iii) determine best set of clusters. MMC differs from Markov Chain Monte Carlo (MCMC) as I do not use Markov Chains and there is no concept of states. Additionally, I did not use *a priori* distributions for the addition and merge criterion, these require the Metropolis Criterion for the acceptance and denial condition.

To determine the termination statistics, random surface regions of various size (in this case 50, 100, 150, …, 350 atoms) were determined by the MODELLER suite [39] method *make_region*. I then calculated the average $C_{trident}$ for that region. This computation was repeated 10,000 times for each region size. Once all $C_{trident}$ values were determined, I generated a cumulative distribution of the values for each region size and normalized the distribution by dividing each frequency by 10,000. I originally attempted to fit the initial distribution, but the closest known distribution was a multi-modal Gaussian mixture model, and even this distribution resulted in a poorly fit model. Given these cumulative distributions, termination probabilities were determined based on cluster size, in atoms, and $C_{trident}$ value.

Once all clusters are determined (using the algorithm described above) for 100 iterations, I needed to choose the best set of clusters, because MMC is non-deterministic. While most clustering algorithms choose the best iteration of clustering, I chose to select from all clusterings. I determined how many times each cluster was generated, then sorted the clusters by, and in the order of: size (in residues), frequency, and $C_{trident}$ value. The largest, most frequent cluster is selected first, then I continued down the list adding each cluster that does not contain residues that are already in a selected cluster.

The MMC method is truly set apart by its reverse approach to statistical data. Typically, one performs clustering then determines the p-values for each cluster through random analysis. MMC allows the user to do all of the statistical analysis first, then the parameters can be refined at the user's discretion. Also, MMC selects the best clustering from an entire set of possible clusterings by *mixing-and-matching* all clusterings.

The clustering of HA and M1 did not require the use of the distance matrix or $C_{trident}$ threshold (results in section 3.2.3), making the overall set up MMC different from the other algorithm previously described. This difference also makes MMC somewhat application specific. MMC can, obviously, be used for diverse regions of proteins with other diversity measures, but it could also be employed for evolutionary studies more generally. MMC could be applied to study clusters of similar genes across multiple genomes where the diversity measure will be similarity across genomes, clusters will be based in gene distance, and initial statistics would be determined by random clusters of genes.

#### *3.1.4.1.5 Comparison of algorithms*

Although all of the clustering algorithms described above are state-of-the-art methods, they were not well suited for my data. The comparative view of clustering results is displayed in Table 8. The Graph-based clustering algorithm found primarily outlier clusters and failed to detect significantly sized clusters. However, Graph-based clustering tended to find the clusters with the smallest $C_{trident}$ values due to the fact that the most diverse residues of HA happen to be directly next to each other. DBSCAN tended to follow the same pattern. Neither of these algorithms were sensitive enough to detect the diverse regions of M1. The K-means clustering did quite well. The clusters for HA were very diverse, but had a high Minkowski distance, meaning that they will be more spread out. This situation was one that I was trying to avoid. Finally, my MMC detected, by far, the largest number of clusters and the largest clusters. The clusters are also more compact (lower Minkowski distance), but sometimes have higher $C_{trident}$ values. In several cases, I listed the largest cluster, even though a smaller cluster was a better representation of the data. This approach was necessary for the purpose of comparison. My goal of creating the MMC method was to detect regions of various diversities, in this way MMC is ideal. If one was interested only in grouping the most extremely diverse regions without concern of cluster shape, K-means would be adequate. Since I was interested in generate binding site-like clusters, I chose to put more emphasis on maintaining a minimal Minkowski distance. As with any clustering problem, the choice of algorithm depends on the type of data and desired result. Because I could not easily, or meaningfully, determine a $C_{trident}$ threshold and I had precomputed statistical data, my MMC is the most appropriate.

| protein | | K-means | DBSCAN | Graph-Based | MMC |
|---|---|---|---|---|---|
| H1N1_HA | #clusters | 13 | 1 | 1 | 39 |
| | largest cluster | 10 | 2 | 2 | 12 |
| | C_trident | 0.49 | 0.28 | 0.28 | 0.64 |
| | Minkowski | 24.40 | 2.4 | 2.40 | |
| H2N2_HA | #clusters | 3 | 1 | 1 | 18 |
| | largest cluster | 4 | 2 | 3 | 10 |
| | C_trident | 0.48 | 0.45 | 0.47 | 0.72 |
| | Minkowski | 13.4 | 4.1 | 7.37 | |
| H3N2_HA | #clusters | 12 | 11 | 5 | 48 |
| | largest cluster | 12 | 11 | 10 | 13 |
| | C_trident | 0.55 | 0.48 | 0.49 | 0.97 |
| | Minkowski | 38.39 | 10.46 | 6.46 | |
| H5N1_HA | #clusters | 16 | 2 | 3 | 38 |
| | largest cluster | 10 | 6 | 5 | 12 |
| | C_trident | 0.56 | 0.51 | 0.49 | 0.69 |
| | Minkowski | 14.25 | 6.19 | 6.95 | |
| H1N1_M1 | #clusters | 1 | 1 | 0 | 10 |
| | largest cluster | 7 | 9 | | 11 |
| | C_trident | 0.64 | 0.65 | | 0.98 |
| | Minkowski | 100.00 | 100.00 | | |
| H2N2_M1 | #clusters | 2.00 | 1.00 | 0 | 5 |
| | largest cluster | 4 | 5 | | 11 |
| | C_trident | 0.66 | 0.58 | | 0.91 |
| | Minkowski | 15.82 | 75.28 | | |
| H3N2_M1 | #clusters | 4 | 1 | 1 | 5 |
| | largest cluster | 3 | 2 | 2 | 10 |
| | C_trident | 0.70 | 0.70 | 0.70 | 0.89 |
| | Minkowski | 7.63 | 3.10 | 3.10 | |
| H5N1_M1 | #clusters | 3 | 1 | 1 | 10 |
| | largest cluster | 4 | 3 | 2 | 13 |
| | C_trident | 0.70 | 0.67 | 0.66 | 0.99 |
| | Minkowski | 4.17 | 4.43 | 5.60 | |

**Table 8|Comparison of all clustering algorithms.**
Each algorithm is compared based on the number of non-outlier clusters found, the largest cluster and its $C_{trident}$ value and Minkowski distance. The largest cluster is the one listed, not necessarily the best cluster.

## *3.2 Results*

### 3.2.1 Data distributions

After the removal of redundancy, we had a significant reduction in data, but not as substantial as with my H1N1 analysis. Most strains came from avian hosts, which is likely because birds are the natural reservoir for influenza. Conservation across subtypes is quite high. This becomes especially when you consider generating homology models. We were able to cover a majority of each protein structure (Table 9) with the exception of M2, which is only partially covered. Because M2 is an ion channel, it is difficult to structurally characterize. Fortunately, PDB recently added PDBID 4WSB which is the full structural complex of the viral polymerase – PA, PB1, and PB2.

| Protein | PDBID | % identity | Coverage |
|---|---|---|---|
| HA | 1H0A (A) | 89.4 | 88 – 501 |
| M1 | 1AA7 (A) | 90.2 | 1 – 132 |
| M2 | 2KIH (A) | 78.9 | 18 – 44 |
| NA | 3B7E (A) | 48.5 | 1 – 452 |
| NP | 2Q06 (A) | 97.0 | 22 – 747 |
| NS1 | 3F5T (A) | 88.3 | 82 – 187 |
| NS2 | 1PD3 (A) | 98.1 | 59 – 92 |
| PA | 4WSB (A) | 70.5 | 1 – 716 |
| PB1 | 4WSB (B) | 79.1 | 1 – 757 |
| PB2 | 4WSB (C) | 67.6 | 1 – 759 |

**Table 9|Protein Structural Models.**
Each protein was able to be mostly covered by only one template with very high sequence similarity. Data for % identity is shown for H1N1.

Finally, the $C_{trident}$ value for each residue of each protein was calculated. However, it appears that the $C_{trident}$ values are partially dependent on the number of sequences that are available (Figure 7). This dependence is due to the fact that $C_{trident}$ calculation is sensitive to the number of sequences available. If there are fewer sequences, it is less likely that diversity will be introduced whereas larger numbers of sequences tend to introduce more diversity. Though a correlation between the number of sequences and $C_{trident}$ values appears to exist, we lack significant data to prove the correlation statistically or correct for the data size bias. We have to account for the possibility that some subtypes could be more conserved than others.

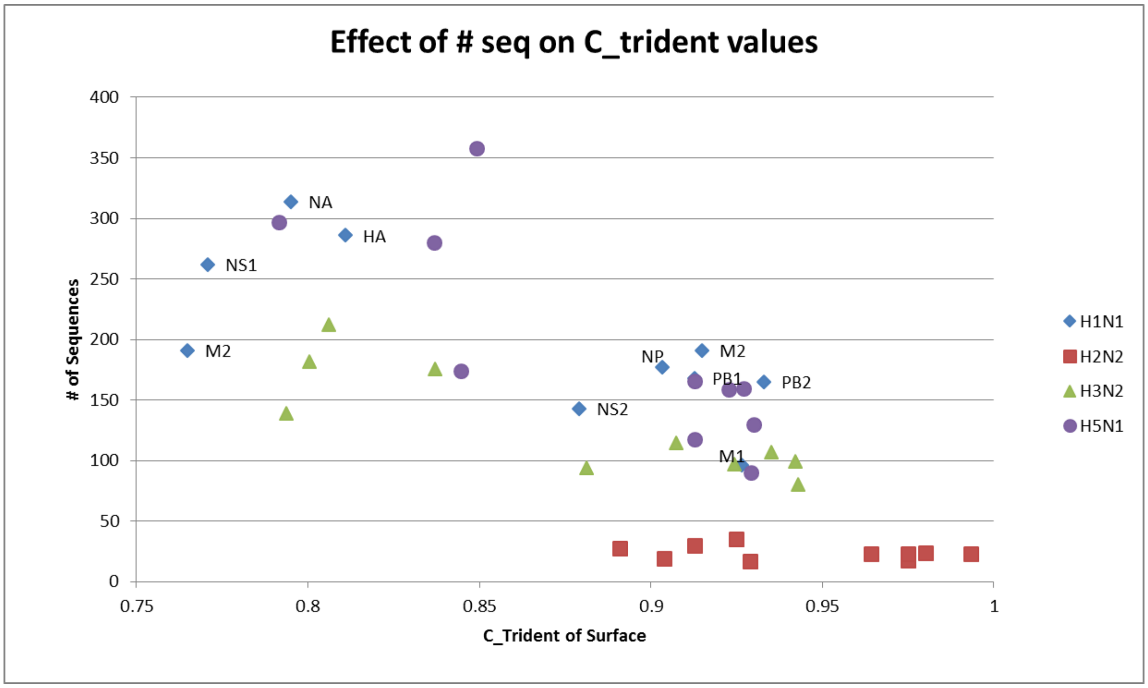


**Figure 7| $C_{trident}$ values possible dependency on number of sequences.**
Larger number of sequences generally results in lower values of $C_{trident}$ values when compared across the same protein of different subtypes.

### 3.2.2 Conserved viral-viral interaction sites

After DBSCAN clustering and random cluster analysis, I found statistically significant clusters on the surface of M1, M2, NP, NS1, NS2, PA, PB1, and PB2. There were even small patches on the stem region of HA, but they had high p-values due to the small number of conserved residues. DBSCAN assigned most of the random clusterings as all noise, which made statistical analysis difficult. For the other proteins, however, similar, if not identical, clusters were found across all four subtypes. This similarity was especially clear for NP, PA, PB1, and PB2: the proteins responsible for the viral RNP (vRNP) complex. The vRNP complex is responsible for packaging the viral RNA along with the polymerase to allow for replication within the host cell. I had previously found evidence of this extreme conservation of the vRNP complex in H1N1 (section 2.3.7).

### 3.2.3 Diverse regions on host-pathogen binding sites

Using MMC clustering, I found several significantly large and diverse clusters on HA and NA. Several other proteins, even M1, had very small diverse clusters that did not fall on any known intra-viral binding sites. Unlike the conserved residues, diverse clusters are not found on intra-viral binding sites, but rather on host-pathogen interaction sites. For HA, diverse clusters were found on the head of the protein for each subtype, but not on the same portion. Though the clusters didn’t significantly overlap, each cluster did overlap with one of the antigenic sites of HA: Sa, Sb, Ca, and Cb (Figure 8). Additionally, NA has a diverse cluster near the NC41 epitope that is required for host recognition.

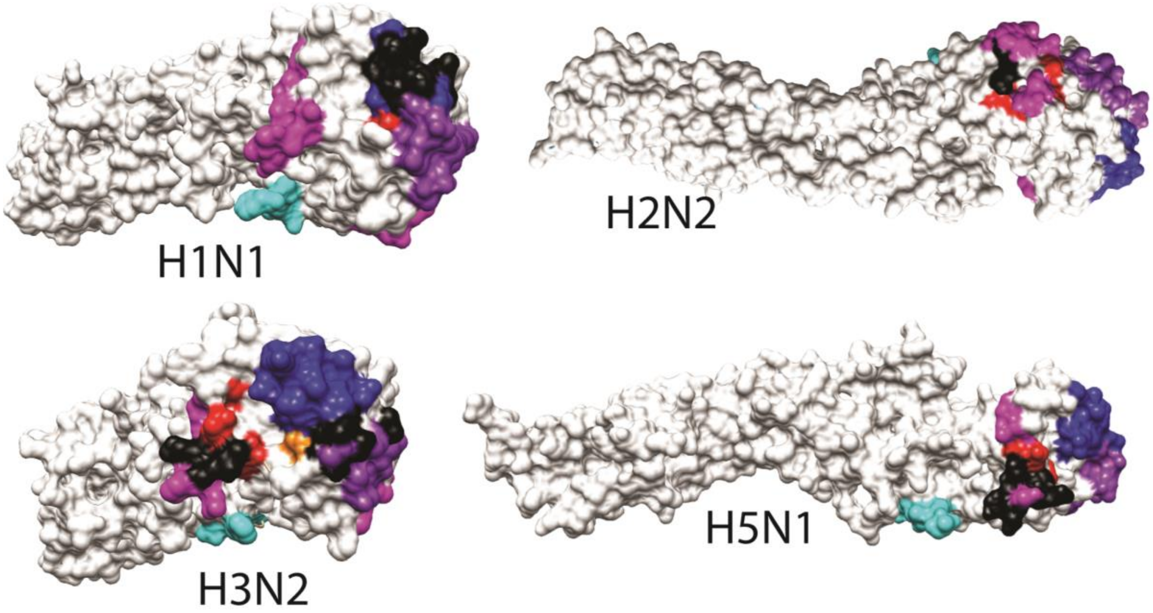


**Figure 8|Diverse Regions of HA.**
Sa in purple, Sb in blue, Ca in magenta, Cb in cyan, and overlap in black. Each subtype has overlap with an antigenic site, but not necessarily with each other.

### 3.2.4 Structural and temporal patterns of influenza evolution

When trying to understand the overall evolution of the influenza subtypes, it is useful to take temporal patterns into account. By comparing the host-pathogen interaction sites of HA across strains from various years beginning with 1918, we find that years with high similarity to the 1918 pandemic flu at the antigenic sites are also times when there was either a pandemic or narrowly avoided pandemic. Though these events are likely related to reassortment events, antigenic drift still occurs; so, to better understand how this occurs, the diversity of the antigenic regions needs to be taken into account. Since I discovered that the antigenic regions overlap with diverse clusters it is likely that the high level of diversity is caused by strong selective pressure.

## *3.3 Discussion*

### 3.3.1 Extreme conservation of vRNP complex

The vRNP complex is essential for viral replication. The vRNP complex is made up of a viral RNA segment, NP dimers form a helical complex with PA, PB1, and PB2 attached at the head. Each of the components must fit together for this complex to form. Additionally, if the vRNP complex is improperly assembly either the vRNA will not reach the viron or the viron will not be able to replicate the vRNA within the host cell, depending on what type of error has occurred. If NP cannot dimerize, then the vRNA cannot wrap around the NP-NP complex meaning that there will be free-floating vRNA in the host cell, which will be destroyed by the host cell defenses. On the other hand, if the polymerase does not properly form or does not attach to the vRNP complex, then the virus cannot copy its genome and becomes inert. Since I found that the intra-viral binding sites for the vRNP complex overlap with extremely conserved clusters, these regions would be excellent drug targets.

### 3.3.2 Improvement of diversity analysis with MMC

As will be described later (section 3.1.4.1.5), the MMC clustering algorithm works very well for diverse residue data. When compared to other clustering algorithm, MMC is more flexible due to the pre-computation of statistical data, and ability to avoid setting a diversity threshold. In fact, with MMC, all thresholds are involved in a Monte Carlo step so all thresholds are elastic. The clusters determined by MMC are small, compact, and highly diverse while other algorithms generated artificially large clusters, or no clusters at all.

### 3.3.3 Results shared across all four pandemic subtypes

Interestingly, the clustering results are similar, sometimes identical, across all four pandemic subtypes for both extremely conserved and diverse regions. The similarity across conserved regions is not a particularly surprising result as the functional regions that they cover are involved in interactions with other portions of the virus. It is unlikely for a mutation to be detected in these regions. The more interesting observation is the similar pattern seen in diverse regions. It is expected that diverse regions would co-localize with host-pathogen binding sites, but it is unexpected that they would overlap with different sites in different subtypes. This overlap is likely due to evolutionary pressure that is placed by different hosts. The immune systems of swine, avian, and human are different and respond to different antigens. It is likely that the difference in diverse regions is due to the profile of hosts.

### 3.3.4 Insights into new drug targets and surveillance methods

As previously discussed, extremely conserved regions are found exclusively on intra-viral binding sites. Many of these regions are shared across subtypes either entirely or partially. Because of this, these regions are ideal targets for universal drug therapies. Currently, drugs work on a subset of subtypes and many strains are now drug resistant. Using these regions, which are under strong negative selection, as drug targets as it is far less likely that the virus will be able to quickly and easily mutate to evade the drug.

Additionally, since we have found that the antigenic regions of HA can give indications of pandemic potential, it is useful to understand that these highly diverse regions could be used to monitor influenza populations.  Surveillance of currently circulating strains is equally as important as drug development, primarily because predicting which strains will be most prevalent is the first step to determining the makeup of that season's vaccine.

# 4 Prediction and Storage of Bacterial Effectors

Computational prediction of bacterial effectors – bacterial proteins secreted via a secretion system to attack the host cell – is a difficult task that had previously only been possible on two of the seven known secretion systems (see section 1.1.2). With the development of Preffector, a tool for genome-wide prediction of bacterial effectors of any secretion type, we are able to generate prediction data quickly; thus, we decided to create a database to hold the results. This database, BacPaC (Bacterial effectors: Predicted and Curated), contains predictions of 14 bacteria from all seven secretion types accompanied by data pertaining to the proteins. Each predicted effector has a unique profile page containing the relevant data represented by intuitive visualizations and easy to interpret menus and lists.

## *4.1 Clustering of effectors predicted by Preffector*

### 4.1.1 Preffector tool

Preffector uses a Support Vector Machine (SVM) approach, a common supervised learning technique, to predict bacterial effectors based on a training set of known effectors and non-effectors. Each protein is represented by a feature vector containing data about the signal regions of the protein, the length of the protein, its secondary structure, its solvent accessibility, its physio-chemical properties, and its dipeptide composition. Signal regions are based on known signal sequences for different secretion systems. Preffector accepts protein sequences as input and predicts which proteins are likely to be effectors.

## 4.1.2 Single-link clustering of predicted effectors

### 4.1.2.1 Hierarchical clustering with unknown threshold

To find possible pathogenicity islands – compact regions in the bacterial genome containing pathogenic genes – we performed a clustering analysis of the effectors in the genome. We began by creating a distance matrix corresponding to the number of genes on the genome between each pair of predicted effectors. Adjacent effectors had a distance of 0. Using this distance matrix, we performed single-link clustering using MATLAB.

Single-linkage clustering is a hierarchical clustering method that creates a dendrogram based on clusters that can be made at any given threshold. The method of single-link clustering is similar, conceptually, to neighbor joining. Beginning with the lowest threshold, $T=0$, all points that are within $T$ of each other are joined together into clusters. Then by incrementing $T$ we begin joining clusters such that clusters $C_i$ and $C_j$ can be joined if there exists elements $k_i \in C_i$ and $k_j \in C_j$ such that $d(k_i, k_j) \leq T$. This procedure is repeated until all clusters are joined into one cluster. Given that we have this dendrogram, post clustering, we can then easily determine the number of clusters for any given threshold. Hierarchical clustering methods are truly ideal for situations where the threshold or number of clusters are unknown as they allow us to do a general clustering and then analyze for different thresholds in minimal time. Since we did not know our number of clusters or a threshold, hierarchical clustering was the most appropriate method. Additionally, the threshold for one bacterial genome would not necessarily be the same for another, mainly because the size of the genomes varied from 900 to 4,000 genes. Of the hierarchical clustering methods, I chose single-linkage clustering because of the inherent structure of our data. Bacterial genomes are circular and each gene has two nearest neighbors, meaning that there is a single linkage between one gene and another. Methods such as density-based clustering would miss the linear distribution of the data.

#### 4.1.2.2 Setting a distance threshold using the knee method

In order to determine an appropriate distance threshold for each bacterial genome, I had to try several and choose the most suitable. I chose the threshold, *T*, using the knee method, which finds the longest plateau and selects a member of that plateau as the threshold, thus selecting a threshold that resulted in stable results. For each genome I ran the hierarchical clustering algorithm using thresholds that vary from 1 to 50 genes. Thus, an effector can be added to a cluster if it is at most *T* genes away from another effector in that cluster. After performing the clustering for each threshold, I plotted them as a stepwise function (Figure 9) of the number of clusters, to help determine the ideal threshold. Once they had been plotted, it was simple to detect the longest plateau, which corresponds to the largest run of thresholds yielding the same number of clusters. The threshold is then set to $T = (t_2 + t_1)/2$, where $t_1$ and $t_2$ are the beginning and end test threshold values of the plateau. If there are multiple candidate thresholds – multiple plateaus of the same size – I chose the smaller threshold. Additionally, we did not accept thresholds that resulted in less than three clusters.

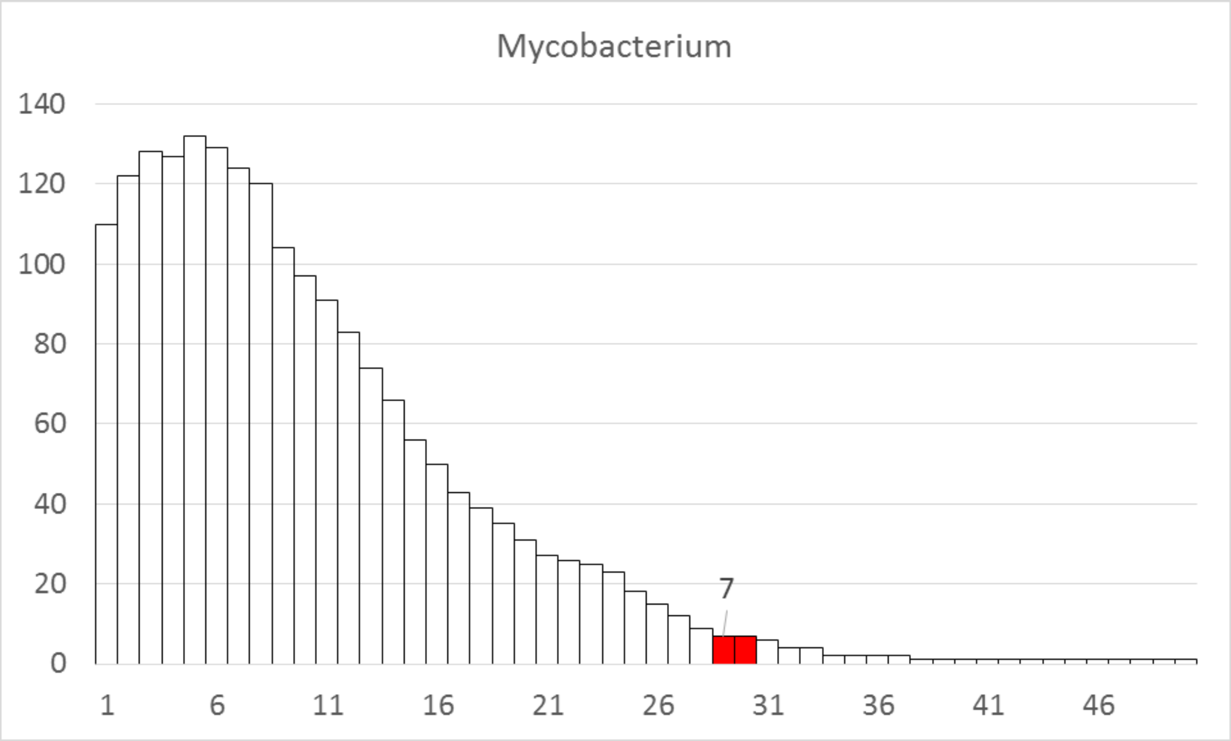


**Figure 9| Threshold plot for clustering of predicted effectors.**
To determine an appropriate threshold for single-link clustering, I generated step-wise plots of the resulting number of clusters to help find the longest plateau.

### *4.1.3 Gene Ontology enrichment among effectors*

We used GO annotations to find the possible functions of the effectors. To map GO terms to our effectors we used Goanna [101], developed by AgBase [102], which maps GO terms based on sequence similarity to annotated sequences. This tool returns 0, 1, or many GO annotations for each protein, also whether the protein function is classified as a biological process (P), molecular function (F), or cellular component (C).

Since the GO is hierarchical in nature, we were able to map our terms, or Slim, to the second highest level in the hierarchy. This reduction was necessary because there was a very large number of GO terms assigned to a limited number of proteins; thus, enrichment studies would provide substantially noisy results. To properly slim our terms, we used CateGOrizer[103], a GO term slimmer which allows you to input your own slimming criterion and maps your current list of GO terms to those in the slim list. This approach left us with 60 unique GO terms.

To determine which functions were enriched or depleted, I used a two-sided p-value test described by Rivals *et al.* [104]. All functions were mapped onto a matrix as follows:

| | Effector | Non-Effector |
|---|---|---|
| GO ID | $n_{11}$ | $n_{21}$ |
| Not GO ID | $n_{12}$ | $n_{22}$ |

Given this we determined the p-value for a given *x* as (a hypergeometric representation of Fisher's exact test):

$$P(N_{11} = x) = \frac{\binom{n_{+1}}{x}\binom{n_{+2}}{n_{12}}}{\binom{n}{n_{1+}}}$$

Where $n_{+1} = n_{11} + n_{21}$. Also,

$$P(N_{11} > n_{11}) = \sum_{i=0}^{i<n_{11}} P(N_{11} = i)$$

The same concept holds for $P(N_{11} > n_{11})$. Using these formulas we determine the p-value for a given $n_{11}$ as

$$p(n_{11}) = 2 \times min\left[P(N_{11} > n_{11}) + \frac{1}{2}P(N_{11} = n_{11}), P(N_{11} < n_{11}) + \frac{1}{2}P(N_{11} = n_{11})\right]$$

The first entry of the *min* function is the p-value for enrichment and the second entry depletion. Using this, we determined the enriched and depleted functions for each bacterial genome (Table 10).

| | **Acientobacter** | | **Chlamydia** | | **Helicobacter** | | **Legionella** | | **Mycobacterium** | |
|---|---|---|---|---|---|---|---|---|---|---|
| **GO TERMS** | p-value | EN/DE | p-value | EN/DE | p-value | EN/DE | p-value | EN/DE | p-value | EN/DE |
| GO:0000003 | 0.719 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 0.868 | 0 |
| GO:0000988 | 0.866 | 0 | 0.659 | 0 | 0.481 | 0 | 0.150 | 0 | 0.624 | 0 |
| GO:0001071 | 0.003 | -1 | 0.512 | 0 | 1.000 | 0 | 0.308 | 0 | 0.004 | -1 |
| GO:0001906 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0002376 | 0.562 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 0.868 | 0 |
| GO:0003824 | 0.000 | -1 | 0.000 | -1 | 0.000 | -1 | 0.000 | -1 | 0.000 | -1 |
| GO:0004872 | 0.000 | 1 | 0.179 | 0 | 0.268 | 0 | 0.638 | 0 | 0.095 | 0 |
| GO:0005198 | 0.027 | 1 | 0.256 | 0 | 0.000 | 1 | 0.961 | 0 | 0.215 | 0 |
| GO:0005215 | 0.002 | -1 | 0.169 | 0 | 0.001 | -1 | 0.000 | -1 | 0.332 | 0 |
| GO:0005488 | 0.002 | -1 | 0.244 | 0 | 0.017 | -1 | 0.001 | -1 | 0.435 | 0 |
| GO:0005576 | 0.375 | 0 | 0.489 | 0 | 0.053 | 1 | 0.005 | 1 | 0.172 | 0 |
| GO:0005623 | 0.038 | -1 | 0.007 | -1 | 0.000 | -1 | 0.000 | -1 | 0.254 | 0 |
| GO:0008152 | 0.008 | 1 | 0.920 | 0 | 0.095 | 0 | 0.065 | 0 | 0.570 | 0 |
| GO:0009055 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0009295 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 0.493 | 0 |
| GO:0009987 | 0.000 | -1 | 0.028 | -1 | 0.019 | -1 | 0.000 | -1 | 0.000 | -1 |
| GO:0016015 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0016020 | 0.000 | -1 | 0.026 | -1 | 0.007 | -1 | 0.000 | -1 | 0.001 | -1 |
| GO:0016209 | 0.281 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 0.132 | 0 |
| GO:0016247 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 0.868 | 0 |
| GO:0016530 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0019012 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0022414 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0022610 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0023052 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0030054 | 0.600 | 0 | 0.179 | 0 | 0.094 | 0 | 0.840 | 0 | 0.997 | 0 |
| GO:0030234 | 0.886 | 0 | 0.869 | 0 | 0.588 | 0 | 0.473 | 0 | 0.414 | 0 |
| GO:0030545 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0031012 | 0.281 | 0 | 1.000 | 0 | 0.253 | 0 | 0.131 | 0 | 1.000 | 0 |
| GO:0031386 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0031974 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0032501 | 0.192 | 0 | 1.000 | 0 | 0.694 | 0 | 0.362 | 0 | 0.372 | 0 |
| GO:0032502 | 0.600 | 0 | 0.411 | 0 | 0.392 | 0 | 0.961 | 0 | 0.002 | 1 |
| GO:0032991 | 0.064 | 0 | 0.228 | 0 | 0.007 | -1 | 0.385 | 0 | 0.889 | 0 |
| GO:0036370 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0040007 | 0.000 | -1 | 0.019 | -1 | 0.105 | 0 | 0.006 | -1 | 0.004 | -1 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| GO:0040011 | 0.866 | 0 | 0.489 | 0 | 0.871 | 0 | 1.000 | 0 | 0.856 | 0 |
| GO:0042056 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0043226 | 0.759 | 0 | 1.000 | 0 | 0.694 | 0 | 1.000 | 0 | 0.754 | 0 |
| GO:0044420 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0044421 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0044422 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0044423 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0044425 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0044456 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0044464 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0044699 | 0.000 | -1 | 0.390 | 0 | 0.209 | 0 | 0.173 | 0 | 0.554 | 0 |
| GO:0045182 | 1.000 | 0 | 1.000 | 0 | 0.694 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0045202 | 0.719 | 0 | 1.000 | 0 | 0.029 | 1 | 0.508 | 0 | 0.655 | 0 |
| GO:0045499 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0045735 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0048511 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0050896 | 0.082 | 0 | 0.150 | 0 | 0.325 | 0 | 0.051 | -1 | 0.000 | 1 |
| GO:0051179 | 0.849 | 0 | 0.804 | 0 | 0.668 | 0 | 0.862 | 0 | 0.509 | 0 |
| GO:0051234 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0051704 | 0.017 | 1 | 0.877 | 0 | 0.923 | 0 | 0.378 | 0 | 0.000 | 1 |
| GO:0055044 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |
| GO:0060089 | 0.026 | -1 | 0.751 | 0 | 0.862 | 0 | 0.040 | -1 | 0.798 | 0 |
| GO:0065007 | 0.446 | 0 | 0.991 | 0 | 0.436 | 0 | 0.787 | 0 | 0.000 | 1 |
| GO:0097423 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 | 1.000 | 0 |

**Table 10|Enriched or and Depleted GO functions.**
Listed are all 60 slimmed GO terms and the p-value for enrichment/depletion. Enrichment is denoted as 1 where depletion is denoted as -1 and are only given if p < 0.06. Enriched functions are highlighted in green while depleted functions are highlighted in red.

## 4.2 *BacPaC: Bacterial Effectors, Predicted and Curated*

### *4.2.1 Methods and Materials*

#### *4.2.1.1Data Curation*

BacPaC integrates data from common and specialized databases and web servers. We began by predicting effectors on a full genome scale for each bacterial genome. Following the prediction we determined functional, structural, interaction, and homology data for the whole genome: effector and non-effector (Figure 10). Function data includes Gene Ontology, Enzyme Commission, as well as host and bacterial subcellular localization. Structural information includes structural templates and their associated PDB [105] IDs, which can be used for homology modeling, and the domain architecture annotated using PFAM [106] and SCOP [107] domain definitions. Interaction information includes literature-based host-pathogen interactions obtained from the HPIDB [108] database, structurally characterized homologous interactions obtained from the DOMMINO [40] database, and PFAM domain-domain interactions obtained from the iPFAM [109] database. Finally, homology information includes homologous proteins that are annotated as effectors or non-effectors.

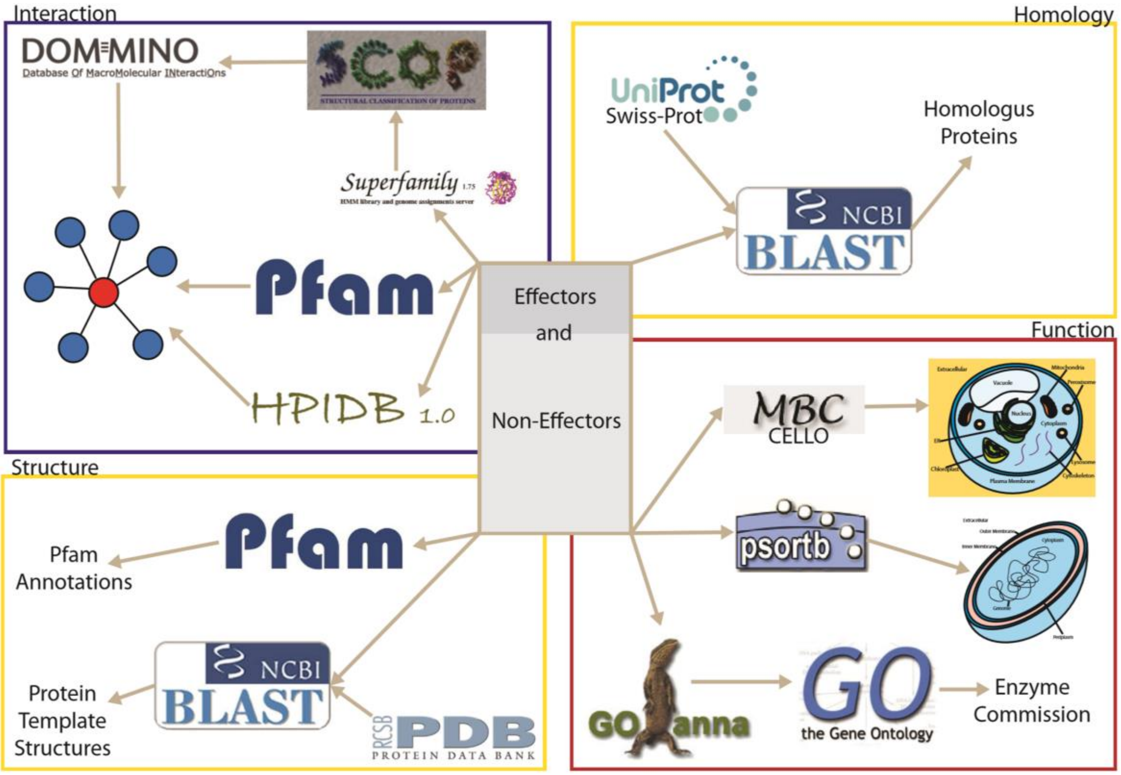


**Figure 10| Data sources and curation.**
All data can be broken in to one of four categories: (a) functional, (b) structural, (c) homology, and (d) interaction.

#### 4.2.1.1.1 Selection of Bacterial species

All 14 bacterial species selected for BacPaC are causative agents of tropical diseases. Tropical diseases occur primarily in underdeveloped and developing nations and are not broadly researched. Most of these conditions either do not exist in developed nations, or they are rare or easily treated. Many of these bacteria, *Yersinia pesits* for example, will occasionally affect people in the United States, but the outbreaks are easily controlled and patients receive the proper treatment. In underdeveloped nations, many of these treatments, or even properly trained medical staff, are unavailable.

The bacteria that we chose (Table 11) are the causative agents of diseases designated tropical or neglected tropical diseases by the WHO. Additionally, we could only select bacteria that utilize secretion systems, namely: Gram-negative bacteria and members of the genus *Mycobacterium*. Interestingly, we found that we could cover all seven secretion types using only causative agents of tropical diseases.

| Bacteria | Disease | # affected/killed | Region |
|---|---|---|---|
| *Acinetobacter baumannii* | Opportunistic infection<br>Affects soldiers injured by IEDs | Unknown number of infections | Middle east<br>Hospitals world-wide |
| *Bordatella pertussis* | Whooping cough | 16 million cases and 195,000 deaths each year | World-wide<br>95% of cases in developing countries |
| *Burkholderia pseudomallei* | Melioidosis | 20-50% mortality rate with treatment | Southeast Asia and Central and South America |
| *Chlamydia trachomatis* | Chlamydial infection<br>Blindness (trachoma) | 499 million/year | World-wide<br>Deaths more prevalent in Africa |
| *Helicobacter pylori* | Gastric ulcers | ~ ½ of the world's population | Worse in developing nations |
| *Legionella pneumophila* | Legionnaires' disease<br>Pneumonia | 4% of all pneumonia cases<br>28% fatality rate | World-wide<br>Under diagnosed in developing nations |
| *Mycobacterium laprae* | Leprosy | 219,000/year | Asia and Africa |
| *Mycobacterium ulcerans* | Buruli ulcer | 5000-6000 reported/year<br>48% are children | Africa and South America (tropical regions) |
| *Neisseria meningitidis* | Meningitis | 88,199 cases<br>5352 deaths | Sub-Saharan Africa |
| *Salmonella enterica* | Typhoid fever | 42,500/year | Developing nations |
| *Shigella dysenteriae* | Dysentery | 120 million/year | Poor Tropical regions |
| *Treponema pallidum* | Yaws, Endemic syphilis | ~75,000/year | Poor Tropical regions |
| *Vibrio cholerae* | Cholera | 3-5 million/year<br>100,000-120,000 deaths/year | Poor Tropical regions |
| *Yersinia pestis* | Bubonic plague | 50%-60% mortality rate<br>Unknown number of infections | Poor Tropical regions |

**Table 11| Causative agents and Effects of 14 Tropical diseases**.
Each of our 14 bacterial genomes corresponds to a tropical disease. Here we describe the disease caused, the impact of the infection, and the regions that are most affected.

#### 4.2.1.1.2 Effector Prediction

| Bacteria | TSS | Genome Length (bp) | N of Effectors | N of Non-Effectors | Total genes |
|---|---|---|---|---|---|
| *Acinetobacter baumannii* | VI | 3,940,614 | 1099 | 2725 | 3824 |
| *Bordatella pertussis* | V | 4,124,236 | 268 | 3188 | 3456 |
| *Burkholderia pseudomallei* | V, VI | 7,247,547 | 683 | 5047 | 5730 |
| *Chlamydia trachomatis* | III | 1,043,025 | 237 | 689 | 926 |
| *Helicobacter pylori* | IV | 1,643,831 | 456 | 1039 | 1495 |
| *Legionella pneumophila* | II, IV | 3,503,610 | 1147 | 2019 | 3166 |
| *Mycobacterium laprae* | VII | 3,268,203 | 274 | 1333 | 1607 |
| *Mycobacterium ulcerans* | VII | 5,631,606 | 567 | 3674 | 4241 |
| *Neisseria meningitidis* | I | 2,272,360 | 489 | 1578 | 2067 |
| *Salmonella enterica* | VI | 4,791,961 | 896 | 3477 | 4373 |
| *Shigella dysenteriae* | II | 4,369,232 | 969 | 3532 | 4501 |
| *Treponema pallidum* | III | 1,138,011 | 154 | 882 | 1036 |
| *Vibrio cholerae* | VI, III | 4,033,464 | 743 | 2818 | 3561 |
| *Yersinia pestis* | III | 4,600,755 | 1083 | 3093 | 4176 |

**Table 12| Secretion systems and predicted effectors.**
For each of the 14 bacterial species, we give the type(s) of secretion systems used by that bacteria. All seven secretion types are represented. Additionally, we show the number of predicted effectors and total number of genes for each genome.

Bacterial effectors were predicted using Preffector (described in section 4.1.1). Because Preffector requires protein sequences as input, we pulled the complete proteome for each bacterial species from the GenBank database[110]. For each species we selected the reference genome and downloaded the proteome and genome. The resulting number of predicted effectors from Preffector are shown in Table 12.  Preffector seems to over-predict for several of the bacterial species.

#### 4.2.1.1.3 General gene information and genomic location

Gene and protein data were obtained from the GenBank database [110].  The genomic localization was extracted from the FASTA file and used to find and map each predicted effector to the bacterial chromosome or plasmid.  Then the mapped positions of the effectors are then visualized using CGview tools [111]. In addition, we mapped the location of effectors known from literature and their homologues predicted by sequence similarity so the user can visualize how these predicted effectors cluster together on the sequence. Additionally, on the inner circle we display all predicted effectors from the genome; and, within the inner most circle we show GC content.

#### 4.2.1.1.4 Structural and homology annotation

Each predicted effector was structurally characterized by determining its sequence- and structure-based domain architectures and providing information on structural templates; thus, enabling users to build a comparative structural model. The sequence-based domain architecture was obtained using Pfam domain annotation[106] of each protein sequence. The structure-based domain architecture was determined based on the SCOP domain definition derived using the SUPERFAMILY tool [112]. Finally, all structural templates of proteins homologous to the proteome are derived from PDB[105]  using the psiBLAST search tool [113] with an e-value cutoff of 0.001 and a percent sequence identity cutoff of 30% - typically used for homology detection. Besides homologues with resolved structures, we also determined the sequential homologues across all kingdoms, using identical parameters for psiBLAST to search the SwissProt sequence repository[114].

##### 4.2.1.1.5 Functional information

The functional information on each protein was derived from Gene Ontology (GO), Enzyme Commission (EC), and host and bacterial subcellular localization data. The goANNA tool [101], a part of the AgBase suit [102], was used to determine the GO IDs for each protein. EC IDs were then derived from those GO IDs. Bacterial and host subcellular localizations were predicted using PSORTb [115] and CELLO [116] (selecting Eukaryotic as the prediction option), respectively. These programs were chosen based on their high accuracy ability to perform batch jobs.

##### 4.2.1.1.6 Interaction information

Interaction information was obtained from three databases: iPfam [109], DOMMINO [40], and HPIDB. Each database contains interaction data obtained using different types of evidence. We obtained possible interactions based on sequence-based structural motifs, Pfam IDs, using iPfam [109]. Similarly, structurally characterized protein interactions between the SCOP domains were extracted from DOMMINO [40]. In addition, the interaction partners obtained are labeled by their superkingdoms that were extracted from the PDB annotation of the interaction structure. Finally, to find data on host-pathogen interactions known from literature and mediated by our bacterial proteins, we searched HPIDB [108]. Because most of our effectors were not previously described in literature, we expanded our search by querying HPIDB for homologous interaction partners. Each of these interaction types is represented with a network visualization tool, developed using the visualization suite D3 [117].

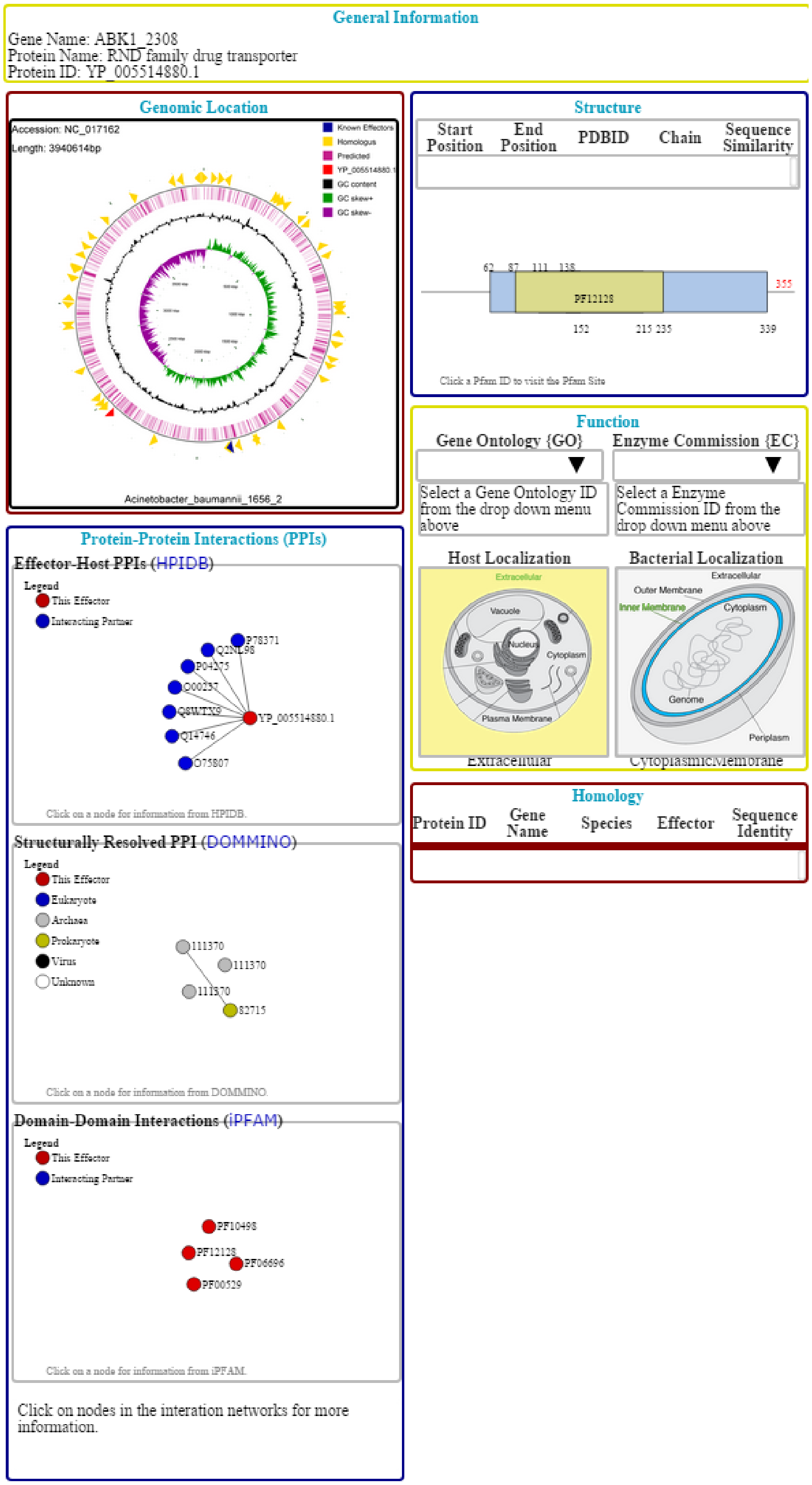
### *4.2.1.2User Interface*

The web-based BacPaC interface supports a comprehensive, effector-centric approach to describing bacterial genomes. Users can access effector data in four different ways: (i) using the basic or advanced keyword search, (ii) conducting a BLAST search, (iii) browsing all effectors, or (iv) downloading comma-separated files containing effector information. Each effector profile contains data pertaining to structure, function, homology, genomic location, and interactions. All data is intuitively displayed using modern network and genome visualization techniques such as D3 [117] and CGView [111], respectively. To ensure that all data is available to the user, much of the information will link to other pages such as PDB, Uniprot, HPIDB, and Pfam.

#### 4.2.1.2.1 **Effector profile page**

A profile page is available for each effector, detailing its function, genomic location, protein-protein interactions, structure, and homology. As shown in Figure 11, profile pages include a general information section regarding the identified effector such as protein name, NCBI accession number, gene name, and organism name. The genomic location section displays a visualization of the bacterial genome with the location of effectors that are predicted, experimentally validated, and homologous to experimentally known effectors. Additional genomic information includes GC content. The structure section lists information about homologous structural templates and the region of the effector sequence that they cover. Each PDBID listed links to PDB. In addition, the domain architecture is shown as a visualization displaying Pfam domain boundaries and identifiers. The function section includes the list of GO Identifiers, EC terms, and a visualization of the effectors' predicted location in host and bacterial cells. GO and EC are listed in dropdown menus, and the corresponding name of the GO or EC will be displayed after the identifier is selected. The interaction section provides three different types of protein-protein interactions (PPIs): (i) host-pathogen PPIs extracted from HPIDB, (ii) structurally resolved interactions (based on SCOP family) extracted from DOMMINO, and (iii) domain-domain interactions shared between Pfam domains extracted from iPfam. Each type of interaction is represented with an interactive network visualization generated using D3 [117]. After clicking a node, the user is redirected to the corresponding external database site for more detailed information on the selected interaction. The homology section contains information about homologues with 30% sequence identity or higher, their annotations as effectors or non-effectors, and source organism.

These profile pages are static pages generated at the time that new effectors are incorporated into the database.  These static pages reduce load time by precomputing information from the database so that the user does not need to wait for database queries.  Additionally, all visualizations are precomputed.



General Information

Gene Name: ABK1_2308
Protein Name: RND family drug transporter
Protein ID: YP_005514880.1

Genomic Location

Accession: NC_017162
Length: 3940614bp

- ■ Known Effectors
- ■ Homologus
- ■ Predicted
- ■ YP_005514880.1
- ■ GC content
- ■ GC skew+
- ■ GC skew-

Acinetobacter_baumannii_1656_2

Structure

| Start Position | End Position | PDBID | Chain | Sequence Similarity |
|---|---|---|---|---|

Click a Pfam ID to visit the Pfam Site

Function

Gene Ontology {GO}
▼
Select a Gene Ontology ID from the drop down menu above

Enzyme Commission {EC}
▼
Select a Enzyme Commission ID from the drop down menu above

Host Localization

Extracellular

Bacterial Localization

CytoplasmicMembrane

Homology

| Protein ID | Gene Name | Species | Effector | Sequence Identity |
|---|---|---|---|---|

Protein-Protein Interactions (PPIs)

Effector-Host PPIs (HPIDB)

Legend
- ● This Effector
- ● Interacting Partner

Click on a node for information from HPIDB.

Structurally Resolved PPI (DOMMINO)

Legend
- ● This Effector
- ● Eukaryote
- ● Archaea
- ● Prokaryote
- ● Virus
- ○ Unknown

Click on a node for information from DOMMINO.

Domain-Domain Interactions (iPFAM)

Legend
- ● This Effector
- ● Interacting Partner

Click on a node for information from iPFAM.

Click on nodes in the interation networks for more information.

**Figure 11| Effector profile pages.**

Each effector is represented by a precomputed profile page. These pages have data broken into sections the same as in Figure 10. An example page is given here.

#### 4.2.1.2.2 Search query and list of results

When using the basic or advanced search, the user may search by: NCBI accession number, protein name, gene identifier, gene name, taxonomy identifier, organism name, bacterial subcellular localization, host subcellular localization, EC number, EC name, GO Identifier, GO name, homology % cutoff, PDB identifier, and Pfam identifier. The basic search allows the user to explore effectors relating to one of these search criteria, while the advanced search option enables the user to narrow their query results using multiple search criteria to filter results. Query results contain the NCBI accession number and protein name for each effector. However, the user has the option to display additional information such as: gene id, GO identifier, EC number, Pfam identifier, structural information, or homologous proteins with 90% sequence identity. The BLAST search option allows the user to query either by entering a protein sequence or by uploading a FASTA file containing at most 50 sequences. The results are displayed as the accession numbers of similar effectors, the percent identity, and e-value. The browse and download functions are organized by organism. When browsing, clicking on the folder of a given organism will expand the list of all associated effectors. A downloadable .zip file with information pertaining to each organism is available in the form of comma-separated files.

### *4.2.2 Results*

#### *4.2.2.1Database content and maintenance*

The critical feature of BacPaC is that effector information is manually curated upon integration. While the manual protocol ensures the annotation accuracy, it requires both computational and human intervention. Some steps of our data mining protocol are done using our in-house databases and tools while other steps require external web-server-based processing. The latter steps include GO annotation and host subcellular localization prediction. Many databases, such as PDB and DOMMINO, are periodically updated; thus, our database will be manually updated on a regular basis. In addition, we update the database content with the improvement of the prediction software, including PREFFECTOR (section 4.1.1). BacPaC is organized as a relational database which is normalized and optimized for scalability with respect to the set of queries defined by the implemented advanced search function (Figure 12).

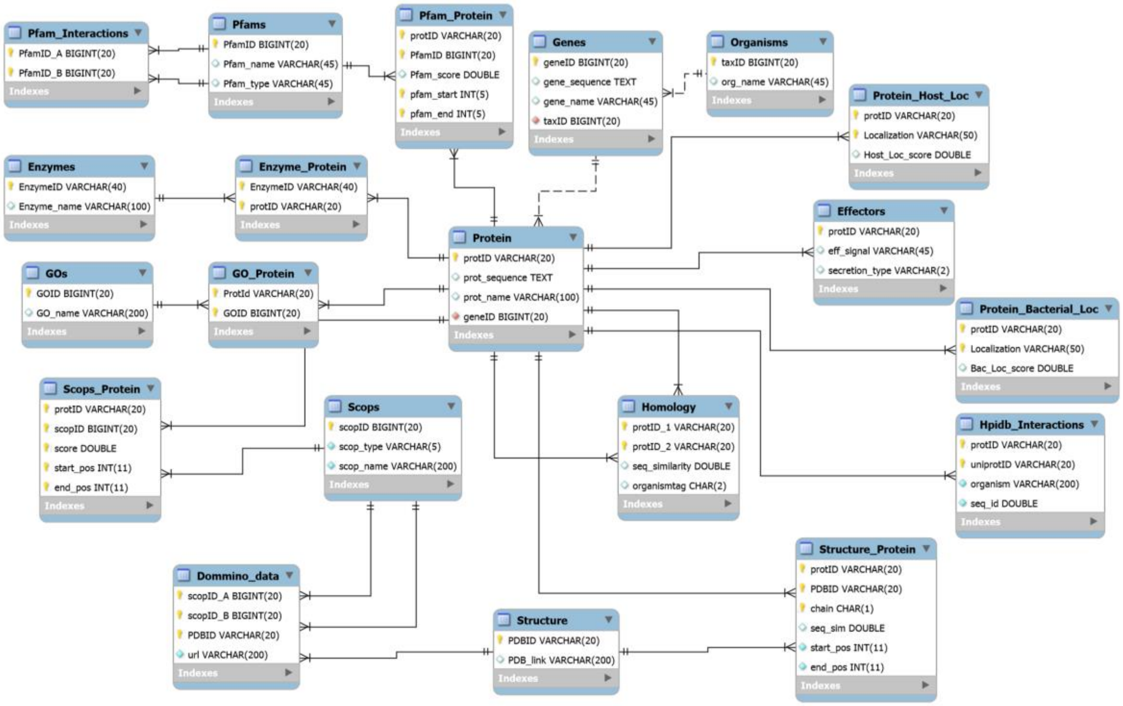


**Figure 12|BacPaC ERD.**
BacPaC is organized around the proteins, not the effectors. This way, data about all bacterial, and homologous, proteins can be stored and used for analysis.

### *4.2.2.2Statistical analysis of effector data*

Currently, comparative genomics studies involving effectors have been nearly impossible due to the lack of centralized data available. Because BacPaC holds data from numerous locations (GO, EC, Pfam, SCOP, etc.), comparative studies are not only possible, but simple. We allow users to download information about any stored bacteria in the form of .zip files that contain comma separated (csv) files of data about that bacteria. These data include GO, EC, Pfam, and SCOP identifiers for all proteins. Since these data are available for all proteins enrichment studies are straightforward, just as is done with Preffector (section 4.1.1).

In addition to enrichment studies, comparative studies across several bacteria is quite easy. Questions such as “Do effectors across all bacteria share the same set of GO functions?”, “Does the type of SCOP classifications change depending on the type of secretion system?”, and “How much structural coverage does PDB have of effectors?” General statistics are also available under the *Statistics* tab on the website.

#### *4.2.2.3Effector profiles allow for fast access and easy understanding*

Profile pages contain all relevant information about an effector in order to offer users a complete understanding of the protein. Because these pages require visualizations and data from each table of the database, it is simpler and faster to generate HTML pages beforehand. All images and drop-down menus are precomputed to avoid pulling data from the database and rendering images at access time. The images in particular need to be precomputed as the genomic location itself typically takes nearly 10 seconds to generate alone. Though this adds an additional step to the update process, it greatly improves the user experience.

#### *4.2.2.4Visualization of genomic location, host/pathogen localization, and interaction networks give users intuitive understanding at a glance*

When creating the web interface for BacPaC, we kept in mind that our main users would be biologists; so, we displayed all data in an intuitively rather than as a spreadsheets or bulleted points. When discussing with our sample group of potential users, they specifically requested that data not be simply listed, but also explained. Though the segmented profile pages break information up clearly, we chose to use several visualizations to help users get immediate knowledge and understanding.

We began by generating genomic atlases for each effector which contains information on proteins homologous to effectors, other predicted effectors, and other known effectors. These images allow users to view a prediction in a genomic context. To better understand where the effector will localize in both the pathogen and host cells, we drew – in Adobe Illustrator – a representation of a bacterial and host cell and set the general coloring to greyscale. When a protein is predicted to fall in a given region, or organelle, that region is colored. This allows the user, at a quick glance, to understand the protein's final destination within the host cell and where it, if not secreted, will localize within the pathogen cell. Finally, we give the user interaction networks based on three criterion: PDB interactions, literature, and Pfam ID interactions. For the PDB-based interactions, mined from DOMMINO [40], we display the information about the source organism via color, thus giving the user the ability to understand what type of interaction is occurring: inter- or intra-species.

These three visualizations give the user different levels of understanding of the predicted effector. The genomic level gives context as compared to other proteins within the same organism. Then the localization level gives context as to where the protein will end up. Finally, the interaction data gives context to the function of the protein once it has entered the host cell. These three pieces of information together give a full context, from beginning to end, of a predicted effector.

### *4.2.3 Discussion*

In this work we introduced BacPaC, a new database of bacterial effectors that covers genomes of the major causative agents of major tropical diseases, spanning all seven secretion systems. Our resource effectively integrates the information about structure, function, genomic location, and interaction of each effector and summarizes the resulting information in a visually intuitive effector profile page. The database includes experimentally, computationally, and homology derived effectors and employs keyword- and BLAST-based search functions that can be combined to create powerful queries. We plan to expand this database by adding genomes of other important pathogens including those affecting animals and plants. In the future, the database will incorporate more systems-level features such as a network view of effectors and their homologues. We intend BacPaC to be useful to a wide range of researchers specializing in fields from computational genomics to experimental microbiology.

# 5 Applying Bioinformatics Tools to Soybean Resistance to Soybean Cyst Nematode

The soybean cyst nematode is the most damaging pathogen of agricultural soybeans, resulting in a severely reduced yield each growing season. Currently, a naturally occurring resistant soybean strain, Forrest, exists but the mechanisms of its resistance is not understood. To find the genes related to resistance *Liu et al* [118] generated several mutant lines, retaining those which displayed complete or partial resistance. They found that the resistant lines had polymorphisms in the *Rhg4* gene that encodes the protein SHMT (serine hydroxymethyltransferase). SHMT converts serine to glycine and, simultaneously, tetrahydrofolate to 5,10-methylenetetrahydrofolate. Thus far, seven mutant lines have been detected, each with unique polymorphisms in this gene.

In addition to *Rhg4*, polymorphisms were also detected in the *Rhg1* gene that encodes SNAP (Soluble NSF Attachment Protein). SNAP is involved in intracellular membrane trafficking including endocytosis and exocytosis. Given these amino-acid changing mutations in both proteins, I generated structural models for each protein to better understand the location of the mutations with respect to the protein surface and potential binding sites.

## *5.1 Mutations of RHG4*

### 5.1.1 Structural modeling of SHMT

Using MODELLER (1EJI as a template), I generated structural models for 8 soybean lines: Essex (wild-type), Forrest (P130R and N358Y), F6266 (E61K), F6756 (M125I), F427-2 (G71D), F1336-1 (L299F), F891-1 (A302V), F1460-2 (G32E), and (Q226*). All mutants have the Forrest mutations in addition to their identifying mutation. It was determined that the mutation M125I of F6756 is on the interior of the protein and is likely to result in an improperly folded, or non-folded, protein. The mutation L299E is also on the interior and could likely cause a similar problem. Q226* (a nonsense mutation) results in the loss of nearly half of the residues of the protein. Given the large section of the protein that is missing, it is very difficult to accurately predict the structure of this mutant, but it is likely the case that this mutant protein is entirely nonfunctional. Even if the protein has a folding conformation, it would be missing the binding pocket that is required for the main function of SHMT occurs. Also, two of the three glycine binding sites are partially, or entirely, missing in this mutant. Omitting this mutant leaves us with Essex, Forrest, F427-2, and F6266 SHMTs that are likely to fold correctly.

### 5.1.2 Mapping of mutations to find proximity to functional regions

SHMT has three major binding functions: three glycine binding sites, one folate binding site, and a dimerization site. To understand how each mutation affects the overall function of SHMT to help better understand the mechanism that incurs resistance, I needed to determine the proximity of the mutations to the functional regions on the structure. Figure 13 shows the location of mutations with respect to each of these binding regions. Both of the Forrest mutations are found in close proximity to the folate and one of the glycine binding sites. Additionally, F6266 and F427-2 mutations are located adjacent to the other portion of the folate binding region and to the dimerization site. These four mutations are the primary focus for the analysis a possible interaction of SHMT and SNAP (section 5.3).

| | **Forrest** | | **F6266** | **F427-2** | **F6756** | **F1336-1** | **F891-1** | **F1460-2** |
|---|---|---|---|---|---|---|---|---|
| **Mutation** | P130R | N358Y | E61K | G71D | M125I | L299F | A302V | G326E |
| **Int/Ext** | **E** | **E** | **E** | **E** | I | I | I | I |
| **Near folate** | x | x | x | x | | | | |
| **Near glycine** | x | x | | | | | | |

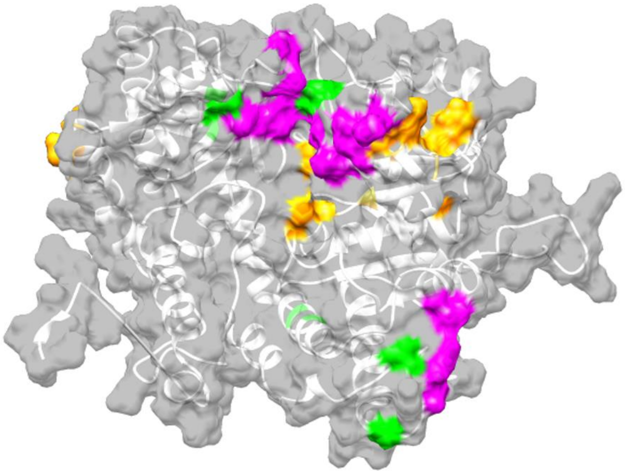


**Figure 13|Structural proximity of SHMT mutations to functional regions.**
Mutations are shown in green, folate binding site in purple, and glycine binding in orange.

## *5.2 Mutations of RHG1*

### 5.2.1 Structural modeling of SNAP

Using MODELLER I created a structural model for Essex, Forrest, and PI88788. There was not just one suitable model for SNAP (unlike SHMT), so I used two templates: 1QQE (31%)

and 2IFU (25%). Since 30% is usually the minimum acceptable threshold for homology modeling, it was necessary to use multiple models and loop refinement to get the best possible model.

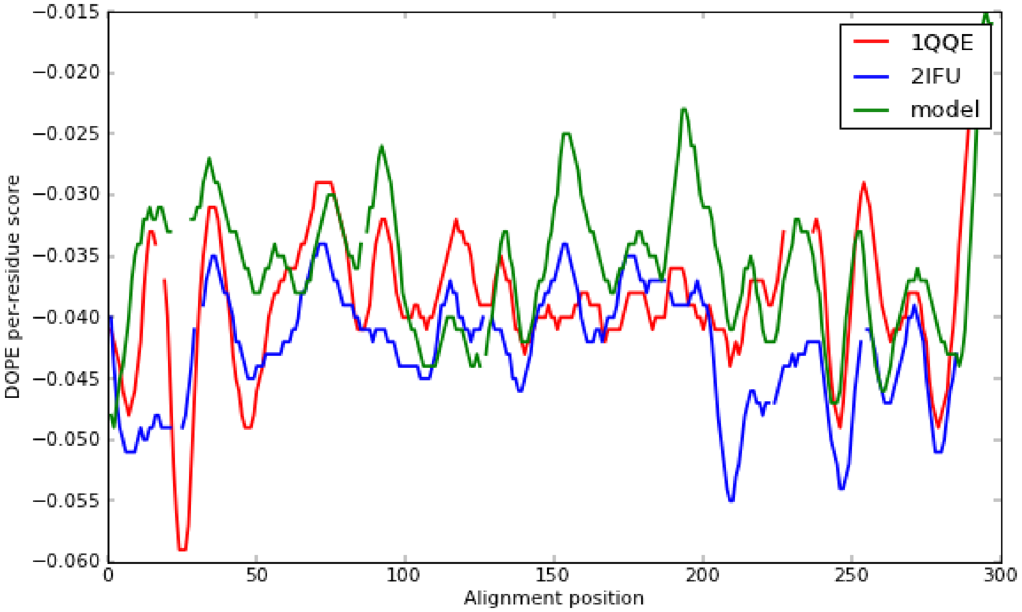


**Figure 14|Structural modeling of SNAP.**
MODELLER uses DOPE score to predict the nativity of a residue. This can be used to compare a generated model with its templates to help determine how similar to each model at each site. The portions where the model is not similar to either template make good candidates for loop refinement.

Figure 14 shows a comparison of the model with each of the two templates. We found regions where the model did not match well with either template as an ideal region to perform loop refinement. The final regions selected for refinement were as follows: 19-25 and 222-228.

Mutations occur at the following sites: 203, 208, 285, 286, 287, 288, and 289. Since the mutation at 288 is an insertion, the mutation 289 maps to 288 on Essex. Mutations 285-289 are located on the C-terminus and have an ill-defined secondary structure, which is common for C-termini. Mutations 203 and 208 are close to each other structurally. 203 is on a helix and 208 is on the adjoining turn (linker). All mutations have exposed regions of the residue; however, with a threshold of 25% exposure, 203 and 208 are not technically exposed.

### 5.2.2 Mapping of mutations to find proximity to functional regions

Because SNAP had far fewer templates than SHMT, mapping of functional regions was very difficult. In order to find possible binding regions, I had to search DOMMINO for remotely homologous structures. Remote homology means that the sequential similarity is less than the standard 30% threshold, but the two proteins have the same SCOP id. In this case, I searched for SCOP ID 48452: TPR-like domain. TPR-like domain is an ancient domain of stacked helixes, so all structures with this domain will be structurally similar. In total, I found 3 PDB IDs (3KD7, 2FBN, and 3R9A) that contained this domain and had structural overlap with the mutations. 3KD7 contained three interactions containing the target SCOP domain, all of which were homomers. 2FBN also contained a homomer interaction. 3R9A, on the other hand, contained a heteromer action (see section 5.3).

To determine the relevant PDB IDs from DOMMINO I needed to perform a structural alignment against all PDBs in DOMMINO with SCOP ID 48452. This alignment was performed using MODELLER. Although better methods exist for structural alignment, MODELLER offered an efficient method for alignment that can be run from a script and requires no human interaction. Once the alignment was completed, I computationally compared them to find if the mutated residues are part of the overlap between the model and the PDB structure. Each overlapping structure was outputted along with the residues involved in the overlap. Since the mutations exist on the unstructured C-terminus, as long as that region overlapped, even slightly, with the structure then it would be identified as a match. Once all matches were identified, I visually verified each match using Chimera.

## *5.3 Possible interaction of SHMT and SNAP*

### 5.3.1 Discovery of a possible interaction

During the prediction of SNAP binding domains I found one PDB structure, 3R9A, which contained a heteromer of human proteins. Where chain B contains a TRP-like domain (SCOPID 48452) that structurally matches with SNAP such that the binding interface of the protein contains mutated residue 203. Chain A binds to chain B and contains the PLP-dependent aminotransferase domain (SCOPID 53383). This SCOPID is the same as that of SHMT. I was not initially looking for any interaction between SHMT and SNAP, but with a thorough evaluation of remote homologs I found that this interaction is possible. Currently, experimental biologists are working to prove that this interaction actually occurs. If this interaction occurs natively in soybeans, then it could be an entirely new mechanism of resistance to soybean cyst nematode and it will have been predicted computationally.

### 5.3.2 Docking of SHMT-SNAP complex

To further understand the binding that may be occurring, I mapped our models for Essex SHMT and SNAP onto the 3R9A structure (Figure 15). The complex resulted in some steric clashing. When working with remote homology, such clashes are often predicted. The interface is actually quite close to that of the 3R9A model. I then used RosettaDock's [119] online tool Rosie to refine the docking, resulting in models that did not have the steric clashes of the original model (Figure 15): the mutations in the C-terminus of SNAP and the two Forrest mutations of SHMT were found to be in the predicted binding regions. This fact means that it is likely that the mutations of SHMT and SNAP work in tandem to incur resistance.

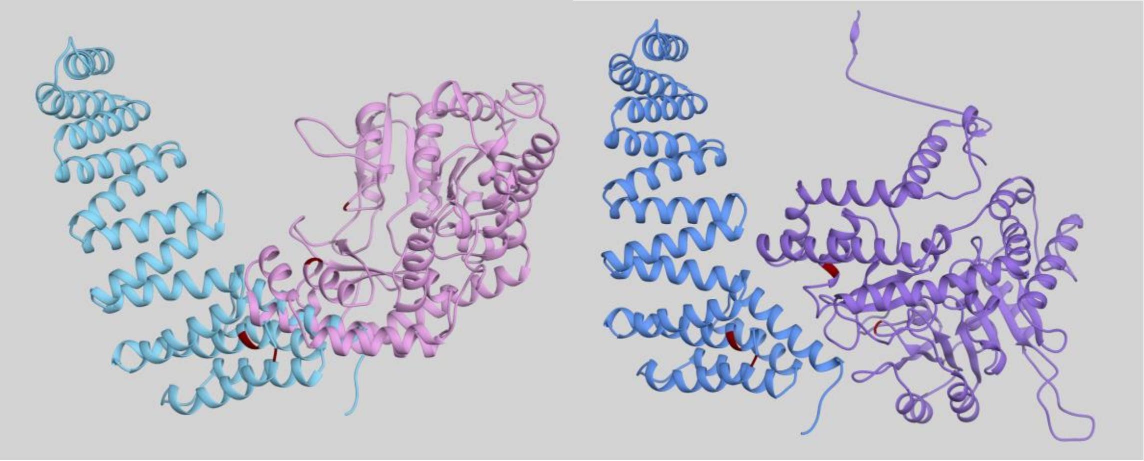


**Figure 15|SHMT-SNAP interface refinement with Rosie.**
Left: the initial overlap of Essex SHMT (pink/purple) and SNAP (blue) with PDB ID 3R9A. Right: complex after refinement with Rosie. After refinement, no steric clashes occur and the mutated regions of SHMT and SNAP are within the binding region. Mutations are shown in red.

### *5.3.2.1Rosetta Dock*

RosettaDock is one of the best macromolecular docking tools available. RosettaDock works by taking an initial complexed model and refining it. By doing this, RosettaDock reduces the search space of interaction dramatically making it faster and more accurate than other tools such as PatchDock. Furthermore, RosettaDock offers an online version of the software called Rosie which will has many of the capabilities of RosettaDock but does not involve installing the software and a user submits a job to their server and awaits a response via email. To refine the docking between SHMT and SNAP I chose the Docking2 protocol that requires only a PDB file of the complex and the chain IDs to be docked. Rosie returns the results in an easy to understand format and allows the user to download the top 10 dockings for comparison. In the case of SHMT-SNAP, each of the top ten dockings had the mutations in the binding regions, giving further evidence to the likelihood that these proteins not only interact, but that the interaction interface contains the mutated regions.

## *5.4 Discussion*

The possible interaction of SHMT and SNAP could indicate a new mechanism for resistance in soybeans. Both proteins perform basic cellular functions and have many interaction partners as a result, meaning that other genes could be involved. At this time, however, we have only detected mutations in SHMT and SNAP in the resistant plant lines. Further work will need to be done to determine whether mutations also occur in other interaction partners of SHMT and SNAP, most importantly, proteins that form larger complexes involving both partners.

# 6 Conclusion

The study of host-pathogen interactions involves the analysis of both the host and the pathogen. I generally focused on the pathogenic side of infection. Using bioinformatics methods, I was able to describe important evolutionary and biological mechanisms for both viruses and bacteria. Using the same methods, I was able to analyze mechanisms of resistance to parasitic infection.

In addition to using current state-of-the-art methods, I developed several methods to improve my analysis, including: Metropolis Criterion Monte Carlo Clustering (MMC), graph-based connectivity clustering, visualization of genomes and phylogenetic trees, and pipeline of analysis to detect remotely homologous interaction pairs. In each case, these computer science and bioinformatics methods are applied to biological data pertaining to host-pathogen interactions to discover patterns of evolution and biological mechanisms of infection.

## *6.1 Clustering as a method of information discovery*

When analyzing protein surfaces and genomes, we rarely know what form our data will take. For my analysis of conserved regions of influenza, it was imperative that I selected my clusters unbiasedly. With the use of clustering methods – Graph-based and DBSCAN – I was able to discover clusters of extremely conserved residues and later determined that these clusters had the common theme of intra-viral interactions. Clustering allowed for naïve discovery of these regions, giving rise to unbiased results.

For the understanding of the diverse regions, however, I had some idea of the type of interactions with which the clusters would be involved. Thus making unbiased information discovery even more important when generating the MMC clustering algorithm. It would have been tempting to simply determine the diversity of the host-pathogen interaction sites and used those as the diverse clusters, but that would not truly have been a new finding. To be sure that I did not bias my results, I designed MMC to find clusters based on robust statistical data and Minkowski distance – a common measure of compactness.

In the analysis of predicted effectors, single-link clustering allowed me to analyze the classification and discover possible pathogenicity islands on bacterial genomes. Discovery of pathogenicity islands is essential in understanding the evolutionary origins of bacterial pathogenesis and possible treatment of infection.

## *6.2 Usage of data visualization to improve understanding*

Data visualization can be used for two major reasons: interpret results and explaining results to others. In order to understand conserved and diverse regions of influenza protein surfaces, it was imperative to visualize the protein structures to see how the locations of these clusters related to known binding sites. Structural visualization became especially important when performing the mutational analysis of SHMT and SNAP and even let to the discovery that the two proteins may interact. Something as simple as structural superposition can be a powerful tool when used in a visualization context. In addition to protein structures, visualization of phylogenetic trees led to a better understanding of the evolutionary dynamics of influenza H1N1. We were able to discover an increased evolutionary rate of HA in a human clade for strains after the introduction of the seasonal vaccine. This visualization, unlike the protein structure visualization, required several layers of information in the form of colors. Though the generation of this information density is computationally costly and time consuming, it allowed me to find results that would have otherwise gone unnoticed.

Though visualization can help interpret results, it can also be used to give quick and intuitive understanding to a variety of audiences. When developing BacPaC to house predicted effectors, we were sure to include many intuitive visualizations on the profile pages to ensure that biologists would be able to quickly gather the information that they need. This data included genomic location, host and bacterial subcellular localization, and potential protein-protein interact partners. Had this data been listed in a table, many non-computational users would have struggled to find the desired information quickly.

## *6.3 Applications of computer science techniques to biological data*

In my analyses, I took many computer science techniques – clustering, data visualization, and database design – and applied them to biological data. Each of these analyses involved determining novel patterns within the data and making that information easily accessible to a broad audience. Whether that was for the determination of the evolutionary dynamics of influenza, prediction and storage of predicted effectors, or determination of possible mechanisms for plant resistance to pathogens, I developed and applied computer science techniques to biological data.

# 7 References

1. Nelson, M.I. and E.C. Holmes, *The evolution of epidemic influenza.* Nature Reviews Genetics, 2007. **8**(3): p. 196-205.

2. Tsai, K.N. and G.W. Chen, *Influenza genome diversity and evolution.* Microbes Infect, 2011. **13**(5): p. 479-88.

3. Guilligay, D., et al., *The structural basis for cap binding by influenza virus polymerase subunit PB2.* Nat Struct Mol Biol, 2008. **15**(5): p. 500-6.

4. Brown, E.G., *Influenza virus genetics.* Biomed Pharmacother, 2000. **54**(4): p. 196-209.

5. Biswas, S.K. and D.P. Nayak, *Influenza virus polymerase basic protein 1 interacts with influenza virus polymerase basic protein 2 at multiple sites.* J Virol, 1996. **70**(10): p. 6716-22.

6. Zurcher, T., et al., *Mutational analysis of the influenza virus A/Victoria/3/75 PA protein: studies of interaction with PB1 protein and identification of a dominant negative mutant.* J Gen Virol, 1996. **77 ( Pt 8)**: p. 1745-9.

7. Rambaut, A., et al., *The genomic and epidemiological dynamics of human influenza A virus.* Nature, 2008. **453**(7195): p. 615-9.

8. Garten, R.J., et al., *Antigenic and genetic characteristics of swine-origin 2009 A(H1N1) influenza viruses circulating in humans.* Science, 2009. **325**(5937): p. 197-201.

9. Abdallah, A.M., et al., *Type VII secretion--mycobacteria show the way.* Nature reviews. Microbiology, 2007. **5**(11): p. 883-91.

10. Buttner, D. and U. Bonas, *Port of entry--the type III secretion translocon.* Trends in microbiology, 2002. **10**(4): p. 186-92.

11. Arnold, R., et al., *Sequence-based prediction of type III secreted proteins.* PLoS Pathog, 2009. **5**(4): p. e1000376.

12. Burstein, D., et al., *Genome-scale identification of Legionella pneumophila effectors using a machine learning approach.* PLoS Pathog, 2009. **5**(7): p. e1000508.

13. McDermott, J.E., et al., *Computational prediction of type III and IV secreted effectors in gram-negative bacteria.* Infect Immun, 2011. **79**(1): p. 23-32.

14. Samudrala, R., F. Heffron, and J.E. McDermott, *Accurate prediction of secreted substrates and identification of a conserved putative secretion signal for type III secretion systems.* PLoS Pathog, 2009. **5**(4): p. e1000375.

15. Yang, Y., et al., *Computational prediction of type III secreted proteins from gram-negative bacteria.* BMC Bioinformatics, 2010. **11 Suppl 1**: p. S47.

16. Hartigan, J.A. and M.A. Wong, *Algorithm AS 136: A k-means clustering algorithm.* Applied statistics, 1979: p. 100-108.

17. Ester, M., et al. *A density-based algorithm for discovering clusters in large spatial databases with noise*. in *Kdd*. 1996.

18. Ankerst, M., et al. *OPTICS: ordering points to identify the clustering structure*. in *ACM Sigmod Record*. 1999. ACM.

19. Johnson, S.C., *Hierarchical clustering schemes.* Psychometrika, 1967. **32**(3): p. 241-254.

20. Davies, D.L. and D.W. Bouldin, *A cluster separation measure.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1979(2): p. 224-227.

21. Fu, L. and E. Medico, *FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data.* BMC bioinformatics, 2007. **8**(1): p. 3.

22. Schuster, S.C., *Next-generation sequencing transforms today's biology.* Nature methods, 2008. **5**(1): p. 16-18.

23. Thieu, T., et al., *Literature mining of host–pathogen interactions: comparing feature-based supervised learning and language-based approaches.* Bioinformatics, 2012. **28**(6): p. 867-875.

24. Bao, Y., et al., *The influenza virus resource at the National Center for Biotechnology Information.* Journal of virology, 2008. **82**(2): p. 596-601.

25. Katoh, K., et al., *MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform.* Nucleic Acids Res, 2002. **30**(14): p. 3059-66.

26. Sali, A. and T.L. Blundell, *Comparative protein modelling by satisfaction of spatial restraints.* J Mol Biol, 1993. **234**(3): p. 779-815.

27. Koradi, R., M. Billeter, and K. Wuthrich, *MOLMOL: a program for display and analysis of macromolecular structures.* J Mol Graph, 1996. **14**(1): p. 51-5, 29-32.

28. Valkenburg, S.A., et al., *Protective efficacy of cross-reactive CD8+ T cells recognising mutant viral epitopes depends on peptide-MHC-I structural interactions and T cell activation threshold.* PLoS Pathog, 2010. **6**(8): p. e1001039.

29. Connolly, M.L., *Analytical molecular surface calculation.* Journal of Applied Crystallography, 1983. **16**(5): p. 548-558.

30. Pond, S.L., S.D. Frost, and S.V. Muse, *HyPhy: hypothesis testing using phylogenies.* Bioinformatics, 2005. **21**(5): p. 676-9.

31. Yang, Z., *Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites.* Molecular Biology and Evolution, 1993. **10**(6): p. 1396-1401.

32. Conant, G.C. and P.F. Stadler, *Solvent exposure imparts similar selective pressures across a range of yeast proteins.* Molecular Biology and Evolution, 2009. **26**(5): p. 1155-1161.

33. Conant, G.C., *Neutral evolution on mammalian protein surfaces.* Trends in Genetics, 2009. **25**(9): p. 377-381.

34. Sokal, R. and F. Rohlf, *Biometry WH Freeman.* New York, 1995. **887**.

35. Gascuel, O., *BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data.* Molecular Biology and Evolution, 1997. **14**(7): p. 685-695.

36. Felsenstein, J., *DNADIST version 3.5 c: Program to compute distance matrix from nucleotide sequences*. 1993, Joseph FelsensteinUniversity of Washington.

37. Felsenstein, J., *{PHYLIP}: phylogenetic inference package, version 3.5 c.* 1993.

38. Saitou, N. and M. Nei, *The neighbor-joining method: a new method for reconstructing phylogenetic trees.* Molecular Biology and Evolution, 1987. **4**(4): p. 406-425.

39. Eswar, N., et al., *Comparative protein structure modeling using Modeller.* Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.], 2006. **Chapter 5**: p. Unit 5 6.

40. Kuang, X., et al., *DOMMINO: a database of macromolecular interactions.* Nucleic Acids Res, 2012. **40**(Database issue): p. D501-6.

41. Stevens, J., et al., *Structure of the uncleaved human H1 hemagglutinin from the extinct 1918 influenza virus.* Science, 2004. **303**(5665): p. 1866-70.

42. Noton, S.L., et al., *Identification of the domains of the influenza A virus M1 matrix protein required for NP binding, oligomerization and incorporation into virions.* J Gen Virol, 2007. **88**(Pt 8): p. 2280-90.

43. Harris, A., et al., *The crystal structure of the influenza matrix protein M1 at neutral pH: M1-M1 protein interfaces can rotate in the oligomeric structures of M1.* Virology, 2001. **289**(1): p. 34-44.

44. Phongphanphanee, S., et al., *Proton transport through the influenza A M2 channel: three-dimensional reference interaction site model study.* J Am Chem Soc, 2010. **132**(28): p. 9782-8.

45. Stouffer, A.L., et al., *Structural basis for the function and inhibition of an influenza virus proton channel.* Nature, 2008. **451**(7178): p. 596-9.

46. Wang, J., et al., *Structural and dynamic mechanisms for the function and inhibition of the M2 proton channel from influenza A virus.* Curr Opin Struct Biol, 2011.

47. Kochendoerfer, G.G., et al., *Total chemical synthesis of the integral membrane protein influenza A virus M2: role of its C-terminal domain in tetramer assembly.* Biochemistry, 1999. **38**(37): p. 11905-13.

48. Biswas, S.K., P.L. Boutz, and D.P. Nayak, *Influenza virus nucleoprotein interacts with influenza virus polymerase proteins.* J Virol, 1998. **72**(7): p. 5493-501.

49. Kobayashi, M., et al., *Molecular dissection of influenza virus nucleoprotein: deletion mapping of the RNA binding domain.* J Virol, 1994. **68**(12): p. 8433-6.

50. Elton, D., et al., *Oligomerization of the influenza virus nucleoprotein: identification of positive and negative sequence elements.* Virology, 1999. **260**(1): p. 190-200.

51. Wang, W., et al., *RNA binding by the novel helical domain of the influenza virus NS1 protein requires its dimer structure and a small number of specific basic amino acids.* RNA, 1999. **5**(2): p. 195-205.

52. Darapaneni, V., V.K. Prabhaker, and A. Kukol, *Large-scale analysis of influenza A virus sequences reveals potential drug target sites of non-structural proteins.* J Gen Virol, 2009. **90**(Pt 9): p. 2124-33.

53. Akarsu, H., et al., *Crystal structure of the M1 protein-binding domain of the influenza A virus nuclear export protein (NEP/NS2).* EMBO J, 2003. **22**(18): p. 4646-55.

54. Boulo, S., et al., *Nuclear traffic of influenza virus proteins and ribonucleoprotein complexes.* Virus Res, 2007. **124**(1-2): p. 12-21.

55. Hemerka, J.N., et al., *Detection and characterization of influenza A virus PA-PB2 interaction through a bimolecular fluorescence complementation assay.* J Virol, 2009. **83**(8): p. 3944-55.

56. Honda, A., K. Mizumoto, and A. Ishihama, *Two separate sequences of PB2 subunit constitute the RNA cap-binding site of influenza virus RNA polymerase.* Genes Cells, 1999. **4**(8): p. 475-85.

57. Guindon, S. and O. Gascuel, *A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood.* Syst Biol, 2003. **52**(5): p. 696-704.

58. Nelson, M.I., et al., *Multiple reassortment events in the evolutionary history of H1N1 influenza A virus since 1918.* PLoS pathogens, 2008. **4**(2): p. e1000012.

59. Zinder, D., et al., *The roles of competition and mutation in shaping antigenic and genetic diversity in influenza.* PLoS pathogens, 2013. **9**(1): p. e1003104.

60. Greenbaum, J.A., et al., *Pre-existing immunity against swine-origin H1N1 influenza viruses in the general human population.* Proceedings of the National Academy of Sciences, 2009. **106**(48): p. 20365-20370.

61. Igarashi, M., et al., *Predicting the antigenic structure of the pandemic (H1N1) 2009 influenza virus hemagglutinin.* PLoS One, 2010. **5**(1): p. e8553.

62. Wang, Y.-T., et al., *Homology modeling, docking, and molecular dynamics reveal HR1039 as a potent inhibitor of 2009 A (H1N1) influenza neuraminidase.* Biophysical chemistry, 2010. **147**(1): p. 74-80.

63. Abdussamad, J. and S.p. Aris-Brosou, *The nonadaptive nature of the H1N1 2009 Swine Flu pandemic contrasts with the adaptive facilitation of transmission to a new host.* BMC evolutionary biology, 2011. **11**(1): p. 6.

64. Tharakaraman, K., et al., *Antigenically intact hemagglutinin in circulating avian and swine influenza viruses and potential for H3N2 pandemic.* Scientific reports, 2013. **3**.

65. Xu, R., et al., *Structural basis of preexisting immunity to the 2009 H1N1 pandemic influenza virus.* Science, 2010. **328**(5976): p. 357-60.

66. Ghedin, E., et al., *Deep sequencing reveals mixed infection with 2009 pandemic influenza A (H1N1) virus strains and the emergence of oseltamivir resistance.* The Journal of infectious diseases, 2011. **203**(2): p. 168-74.

67. Guharoy, M. and P. Chakrabarti, *Conserved residue clusters at protein-protein interfaces and their use in binding site identification.* BMC bioinformatics, 2010. **11**: p. 286.

68. Panjkovich, A. and X. Daura, *Assessing the structural conservation of protein pockets to study functional and allosteric sites: implications for drug discovery.* BMC structural biology, 2010. **10**: p. 9.

69. Ma, B., et al., *Protein-protein interactions: structurally conserved residues distinguish between binding sites and exposed protein surfaces.* Proc Natl Acad Sci U S A, 2003. **100**(10): p. 5772-7.

70. Swapna, L.S., et al., *Roles of residues in the interface of transient protein-protein complexes before complexation.* Scientific reports, 2012. **2**: p. 334.

71. Wei, C.J., et al., *Cross-neutralization of 1918 and 2009 influenza viruses: role of glycans in viral evolution and vaccine design.* Sci Transl Med, 2010. **2**(24): p. 24ra21.

72. Woolhouse, M.E., et al., *Biological and biomedical implications of the co-evolution of pathogens and their hosts.* Nat Genet, 2002. **32**(4): p. 569-77.

73. Li, D., et al., *Genetic analysis of influenza A/H3N2 and A/H1N1 viruses circulating in Vietnam from 2001 to 2006.* J Clin Microbiol, 2008. **46**(2): p. 399-405.

74. Gaydos, J.C., et al., *Swine Inftuenza A at Fort Dix, New Jersey (January,ÄìFebruary 1976). I. Case Finding and Clinical Study of Cases.* Journal of Infectious Diseases, 1977. **136**(Supplement 3): p. S356-S362.

75. Neumann, G., T. Noda, and Y. Kawaoka, *Emergence and pandemic potential of swine-origin H1N1 influenza virus.* Nature, 2009. **459**(7249): p. 931-939.

76. Smith, G.J., et al., *Origins and evolutionary genomics of the 2009 swine-origin H1N1 influenza A epidemic.* Nature, 2009. **459**(7250): p. 1122-1125.

77. Fedson, D.S. and H.A. Kessler, *A hospital-based influenza immunization program, 1977-78.* American journal of public health, 1983. **73**(4): p. 442-445.

78. Sencer, D.J. and J.D. Millar, *Reflections on the 1976 swine flu vaccination program.* Emerging infectious diseases, 2006. **12**(1): p. 29.

79. Pensaert, M., et al., *Evidence for the natural transmission of influenza A virus from wild ducts to swine and its potential importance for man.* Bulletin of the World Health Organization, 1981. **59**(1): p. 75-8.

80. Schultz, U., et al., *Evolution of pig influenza viruses.* Virology, 1991. **183**(1): p. 61-73.

81. Guan, Y., et al., *Emergence of avian H1N1 influenza viruses in pigs in China.* Journal of virology, 1996. **70**(11): p. 8041-6.

82. Bachmann, P.A., ed. *Swine influenza virus.* Virus Infections of Porcines., ed. M.B. Pensaert. 1989, Elsevier: Amsterdam, Netherlands. 193-207.

83. Brown, I.H., *The epidemiology and evolution of influenza viruses in pigs.* Veterinary microbiology, 2000. **74**(1-2): p. 29-46.

84. Yin, C., et al., *Conserved surface features form the double-stranded RNA binding site of non-structural protein 1 (NS1) from influenza A and B viruses.* J Biol Chem, 2007. **282**(28): p. 20584-92.

85. Okuno, Y., et al., *A common neutralizing epitope conserved between the hemagglutinins of influenza A virus H1 and H2 strains.* J Virol, 1993. **67**(5): p. 2552-8.

86. Fraser, H.B., et al., *Evolutionary rate in the protein interaction network.* Science, 2002. **296**(5568): p. 750-2.

87. Korber, B.T., et al., *Covariation of mutations in the V3 loop of human immunodeficiency virus type 1 envelope protein: an information theoretic analysis.* Proc Natl Acad Sci U S A, 1993. **90**(15): p. 7176-80.

88. Travers, S.A., et al., *A study of the coevolutionary patterns operating within the env gene of the HIV-1 group M subtypes.* Mol Biol Evol, 2007. **24**(12): p. 2787-801.

89. Corti, D., et al., *A neutralizing antibody selected from plasma cells that binds to group 1 and group 2 influenza A hemagglutinins.* Science, 2011. **333**(6044): p. 850-6.

90. Ekiert, D.C., et al., *A highly conserved neutralizing epitope on group 2 influenza A viruses.* Science, 2011. **333**(6044): p. 843-50.

91. Ekiert, D.C., et al., *Antibody recognition of a highly conserved influenza virus epitope.* Science, 2009. **324**(5924): p. 246-51.

92. Sui, J., et al., *Structural and functional bases for broad-spectrum neutralization of avian and human influenza A viruses.* Nature structural & molecular biology, 2009. **16**(3): p. 265-73.

93. Throsby, M., et al., *Heterosubtypic neutralizing monoclonal antibodies cross-protective against H5N1 and H1N1 recovered from human IgM+ memory B cells.* PLoS One, 2008. **3**(12): p. e3942.

94. Pielak, R.M., J.R. Schnell, and J.J. Chou, *Mechanism of drug inhibition and drug resistance of influenza A M2 channel.* Proceedings of the National Academy of Sciences of the United States of America, 2009. **106**(18): p. 7379-84.

95. Gubareva, L.V., *Molecular mechanisms of influenza virus resistance to neuraminidase inhibitors.* Virus research, 2004. **103**(1-2): p. 199-203.

96. Chen, G.L. and K. Subbarao, *Attacking the flu: neutralizing antibodies may lead to 'universal' vaccine.* Nature medicine, 2009. **15**(11): p. 1251-2.

97. Blok, V., et al., *Inhibition of the influenza virus RNA-dependent RNA polymerase by antisera directed against the carboxy-terminal region of the PB2 subunit.* J Gen Virol, 1996. **77 ( Pt 5)**: p. 1025-33.

98. Edgar, R.C., *MUSCLE: multiple sequence alignment with high accuracy and high throughput.* Nucleic acids research, 2004. **32**(5): p. 1792-1797.

99. Valdar, W.S., *Scoring residue conservation.* Proteins: structure, function, and bioinformatics, 2002. **48**(2): p. 227-241.

100. Kahraman, A., L. Malmström, and R. Aebersold, *Xwalk: computing and visualizing distances in cross-linking experiments.* Bioinformatics, 2011. **27**(15): p. 2163-2164.

101. McCarthy, F.M., et al., *AgBase: a functional genomics resource for agriculture.* BMC Genomics, 2006. **7**: p. 229.

102. McCarthy, F.M., et al., *AgBase: a unified resource for functional analysis in agriculture.* Nucleic Acids Res, 2007. **35**(Database issue): p. D599-603.

103. Zhi-Liang, H., J. Bao, and J. Reecy, *CateGOrizer: a web-based program to batch analyze gene ontology classification categories.* Online J Bioinformatics, 2008. **9**(2): p. 108-112.

104. Rivals, I., et al., *Enrichment or depletion of a GO category within a class of genes: which test?* Bioinformatics, 2007. **23**(4): p. 401-407.

105. Bernstein, F.C., et al., *The Protein Data Bank. A computer-based archival file for macromolecular structures.* Eur J Biochem, 1977. **80**(2): p. 319-24.

106. Punta, M., et al., *The Pfam protein families database.* Nucleic Acids Res, 2012. **40**(Database issue): p. D290-301.

107. Murzin, A.G., et al., *SCOP: a structural classification of proteins database for the investigation of sequences and structures.* J Mol Biol, 1995. **247**(4): p. 536-40.

108. Kumar, R. and B. Nanduri, *HPIDB--a unified resource for host-pathogen interactions.* BMC Bioinformatics, 2010. **11 Suppl 6**: p. S16.

109. Finn, R.D., M. Marshall, and A. Bateman, *iPfam: visualization of protein-protein interactions in PDB at domain and amino acid resolutions.* Bioinformatics, 2005. **21**(3): p. 410-2.

110. Benson, D.A., et al., *GenBank.* Nucleic Acids Res, 2005. **33**(Database issue): p. D34-8.

111. Stothard, P. and D.S. Wishart, *Circular genome visualization and exploration using CGView.* Bioinformatics, 2005. **21**(4): p. 537-9.

112. Gough, J., et al., *Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure.* J Mol Biol, 2001. **313**(4): p. 903-19.

113. Altschul, S.F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.* Nucleic Acids Res, 1997. **25**(17): p. 3389-402.

114. Bairoch, A. and R. Apweiler, *The SWISS-PROT protein sequence data bank and its new supplement TREMBL.* Nucleic Acids Res, 1996. **24**(1): p. 21-5.

115. Yu, N.Y., et al., *PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes.* Bioinformatics, 2010. **26**(13): p. 1608-15.

116. Yu, C.S., et al., *Prediction of protein subcellular localization.* Proteins, 2006. **64**(3): p. 643-51.

117. Bostock, M., V. Ogievetsky, and J. Heer, *$D^3$ Data-Driven Documents.* Visualization and Computer Graphics, IEEE Transactions on, 2011. **17**(12): p. 2301-2309.

118. Liu, S., et al., *A soybean cyst nematode resistance gene points to a new mechanism of plant resistance to pathogens.* Nature, 2012. **492**(7428): p. 256-260.

119. Gray, J.J., et al., *Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations.* Journal of molecular biology, 2003. **331**(1): p. 281-299.

# VITA

Samantha Warren was born in Fort Riley, Kansas in 1989 and grew up in Raytown, Missouri – a suburb of Kansas City. She attended Raytown South high school where she graduated at the top of her class. In fall of 2007 she began her studies at the University of Missouri – Columbia where she began studying Biochemistry and Mathematics. Though she graduated in December 2010 with a B.S. in Mathematics, she continued studying biochemistry independently as part of her research which she began as a freshman. In January of 2011, she began her Doctoral studies with Dr. Dmitry Korkin at UMC. During this time, she worked as a researcher and teaching assistant and helped run the HHMI Summer Biomedical Informatics Institute (SBII) for four consecutive years. Beginning in fall of 2015, Samantha will be employed as a tenure-track faculty member at Fontbonne University to begin their Bioinformatics program.