BSc THESIS

"*In silico* mutagenesis and docking studies of an influenza hemagglutinin/receptor complex"

Boukoura Theodora (1406)

Supervisor:
Dr. Nicholas M. Glykos
Assistant Professor of Structural and Computational Biology
Department of Molecular Biology and Genetics
Health – Sciences School
Democritus University of Thrace

Alexandroupolis, Greece, December 2018

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor for his guidance and support but also for the fact that he allowed me to be independent during this project and find solutions on my own. Through this hard process I realized that doing research can be an abyss most of the times – but it remains interesting. He is a true mentor, one of my inspirations both as a scientist and as a person and I couldn't have done a better supervisor choice for this thesis. I hope that I will be able to develop as good scientific thinking as his some day.

I would also like to thank my best friend and colleague Panagiotis Athanasopoulos for his constant, tireless support and love and for putting up with my frustration during the entire project as well as the rest of the people in the group, for helping create a nice working environment. Also my friend Thanos Panagiotidis, for hosting me during the dark times of writing this thesis.

In addition, I have to thank Dr. Diomedes E. Logothetis and Sophia Hirakis for their valuable piece of advice regarding protein-ligand docking and Professor Derek Smith for his inspirational lecture during the 2016 Amgen Scholars Europe Symposium in Cambridge that motivated me to work on *Influenza A virus* Hemagglutinin.

In the end I couldn't forget to thank the Educational Institution of George and Anna Sakellariou, for financially supporting my undergraduate studies through their scholarship as well as my family.

Contents

ABSTRACT

Hemagglutinin is the protein found on the surface of *Influenza virus* and is responsible, among others, for its attachment to the host cells' sialic acids, the first step of the virus' entrance to the host cell. Many antibodies target the head of the hemagglutinin trimer, where the Receptor Binding Site is located and substitutions that occur near it, rendering the virus capable of escaping immunity, may also affect the binding efficiency.

The goal of this project was to test whether a system employing docking tools can be built in order to test the effect of single substitutions, known to allow the virus to escape antibodies(26), on the binding efficiency of hemagglutinin to the sialic acid receptors as well as the orientation that the ligand adopts inside the binding pocket of hemagglutinin.

As the system was being prepared, it became clear that the parameters applied were not suitable to be used for the set of substitutions originally aimed to be tested, as these were laying on the borders of the system built. I thus focused on performing redocking simulations of a sialic receptor analogue to hemagglutinin and assessing the results, as well as performing simulations using the parameters determined in these redocking experiments, after introducing either the 224EA or the 91YA substitution in the hemagglutinin molecule. The first one is known to escape immunity, while leaving the binding efficiency intact(26), while tyrosine 91 is a conserved amino acid, forming part of the Receptor Binding Site.

The docking program used (Autodock Vina) did manage to suggest models similar to the crystal pose of the ligand in the redocking experiments, although with less predictability and consistency when more degrees of freedom were allowed to the ligand.

This lack of predictability and consistency as more degrees of freedom were allowed, meaning the system was approaching more the real conditions, renders the current parameters of the process built unsuitable to be used to predict changes in binding efficiency and ligand orientation inside the binding pocket.

However, the results of the redocking simulations are encouraging, suggesting that if a more fine – tuned set of parameters is used, conclusions could be drawn regarding the binding efficiency of hemagglutinin molecules bearing single mutations to sialic acids, making it possible to predict if immunity escaping strains of the virus are still able to bind to the host receptors. Also, having created a protocol including processes from the editing of the PDB structures and docking to calculating distance matrices as well as to cluster analysis and multi-dimensional scaling, fine – tuning the process to be more suitable is now a step closer.

1. INTRODUCTION

1.1 Influenza ("the flu")

The flu is a contagious illness of the respiratory tract, caused by influenza viruses; it can lead from minor to severe illness and has also caused many deaths. The virus is airborne and can be spread in tiny droplets produced by the affected animals. The symptoms vary from fever and cough to vomiting and diarrhea, especially in children. Flu can lead to complications (such as pneumonia), especially when it comes to people who are at high risk, such as the elderly, people with asthma or heart disease and many others.

The differences between seasonal and pandemic flu should also be noted. While seasonal flue happens every year, with a peak between December and February on the northern hemisphere, pandemic flu rarely happens and may cause a major impact on the general public on financial, health and social level. Regarding the immunity, most people are protected when it comes to seasonal flu, as they have been infected with other viruses in the past and/or they have been vaccinated. However they are not protected against pandemic flu as they have never been exposed to the virus or similar viruses before. Death rates are very different between pandemic and seasonal flu as well. While the Centers for Disease Control and Prevention (CDC) estimates that each year (since 2010) deaths range between 12,000 and 56,000 per year, pandemic flu causes many times more deaths (Table 1).

While there are four different types of *Influenza virus* (see Section 1.2.2.), only *Influenza A virus* can cause a global outbreak (pandemic). *Influenza A virus* undergo constant changes (see Section 1.2.4.), thus making it possible for non-human influenza viruses to change in a way they can infect humans and spread among them.

The last pandemic was 2009 H1N1; the World Health Organization declared an end to it on August 10, 2010(1). After the pandemic, this specific virus continues to circulate as a seasonal influenza virus, infecting people all over the world and replaced the previous H1N1 virus circulating in humans.

An example of a virus that could cause a pandemic currently is the Highly pathogenic avian influenza A (HPAI) subtype H5N1, which often crosses the barriers and is transmitted between avians and humans(2). What's more, human-to-human transmission has been detected but not confirmed to be sustained. If the virus gains the ability to transmit from human to human and not only from avian to human, it could cause pandemic, since little immunity against it exists in the population.

It is also very important to not confuse the flu with "common cold" which is caused by other types of viruses, usually rhinoviruses and coronaviruses and is a mild illness of the upper respiratory tract.

1.2 The *Influenza virus*

1.2.1 General Information

Influenza is caused by a virus named *Influenza virus*. The different types of influenza viruses (A, B, C and D) belong to the *Orthomyxoviridae* family(3,4,5) and are ssRNA negative – strand viruses. A and B can cause epidemics in humans but not C, while D only affects cattle(6). *Influenza A virus* is capable of causing pandemics (Table 1). The structure used in the simulations of this thesis comes from an *Influenza A virus;* thus only type A will be further discussed.

Influenza A virus genome consists of eight RNA segments, each coding for 1-2 of the 11 proteins of the virus. 9 of them are packaged in virions, while 2 of them, hemagglutinin (HA) and neuraminidase (NA) are envelope (/surface) glycoproteins. The ratio of NA/HA on the surface of a single virion is 17/80, with approximately 100 NA copies and 500 HA copies. It should be stressed that **these surface proteins act as antigens**. Also, it is important to note that the protein M2 is an integral membrane protein that acts as ion channel and is essential for the uncoating of the virion, by lowering the pH inside the virion. It is only found in 16 to 20 copies per virion(7).

As far as the viral life cycle is concerned, after the attachment and receptor binding (virus adsorption), the virus enters the cell by either clathrin – dependent endocytosis (2/3) or a clathrin– and caveolin – independent pathway (1/3)(8). Then the viral membrane is fused with the endosome/caveosome/macropinosome/ lysosome membrane, after the pH drops inside the virion. Both the adsorption of the virion on the cell as well as the fusion are mediated by hemagglutinin molecules (HA). After uncoating, the RNA segments are imported into the nucleus, where transcription and replication take place. Protein synthesis takes place in the cytoplasm and after post-translational trafficking (where needed) the virions assemble and the RNA segments and essential proteins are packaged inside it; then budding and release take place. Release is mediated by the second envelope protein, neuraminidase (NA)(9) (Figure 1).

The virus infects a variety of cells, including alveolar and bronchial epithelial tissue (BET) cells, alveolar macrophages (AM), lung epithelial tissue (LET) cells and more specifically type II pneumocytes, plasmacytoid dendritic cells (pDCs) and natural killer cells (NKs).

Name of pandemic	Period	Deaths (worldwide)
Spanish flu	1918 – 1919(10)	50 million
Asian flu	1957 – 1959(11)	1.1 million
Hong Kong flu	1968 – 1969(12)	1 million
Russian flu(13)	1977 – 1978	moderate pandemic
2009 H1N1 pandemic	2009 – 2010	151,700 – 575,400(14)

Table 1. *Past influenza pandemics and number of deaths each caused*(15).

1.2.2 Types, Subtypes, Strains and Nomenclature

Influenza viruses are divided in types A, B, C and D; the different types are determined based on antigenic differences on the nuclear and matrix proteins. Type A is further subdivided in subtypes, based on differences of the surface proteins hemagglutinin and neuraminidase. Thus, there are at least 18 types of HA and 11 of NA that, combined, determine the subtype of *Influenza A virus*.

Influenza B viruses are not further subdivided in subtypes; however they are subdivided in strains and lineages. Influenza A viruses are further subdivided in strains. Each strain is named according to a specific nomenclature system, published by WHO in 1979(16). According to this, influenza viruses names include the antigenic type, the host of origin (except for human originated viruses), the geographical origin, the strain number and the year of isolation, separated by slashes (/) as in the following example:

A/duck/Italy/574/1966(H10N2)

This is the name of an *Influenza A virus* isolated from a duck in Italy in 1966, with strain number 574 and belongs to H10N2 subtype, meaning the virion has H10 and N2 subtypes of HA and NA proteins respectively on its surface.

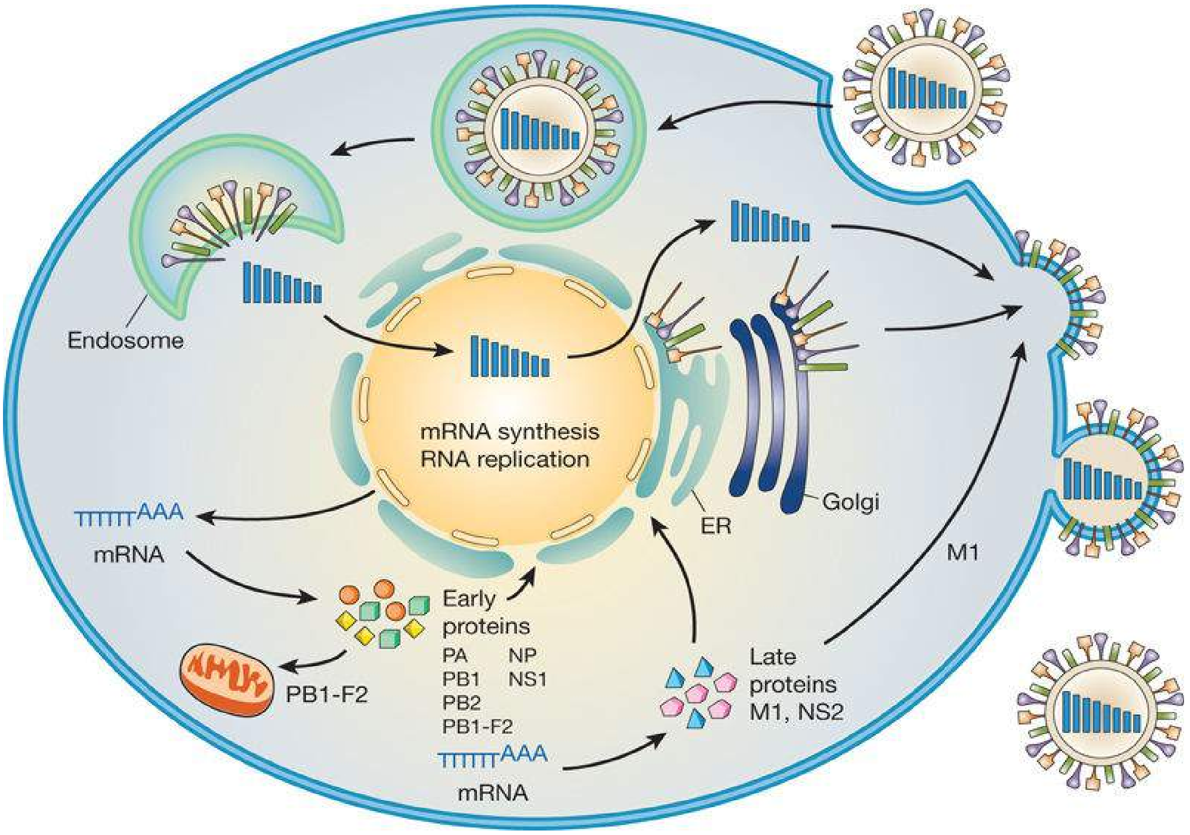


Figure 1. *Schematic diagram of the influenza viral life cycle.* (17)

1.2.3 Vaccines

Currently only strains that belong to the subtypes H1N1 and H3N2 circulate amongst humans(18), while the circulating Influenza B viruses belong to two lineages: either B/Yamagata or B/Victoria. These also determine which strains are included each year in the seasonal flu vaccine, which are usually different for the northern and the southern hemisphere. WHO releases recommended composition of influenza virus vaccines for each flu season.

- **Recommended composition of influenza virus vaccines for use in the 2017- 2018 northern hemisphere influenza season**(19)

✔ an A/Michigan/45/2015 (H1N1)pdm09-like virus;
✔ an A/Hong Kong/4801/2014 (H3N2)-like virus; and
✔ a B/Brisbane/60/2008-like virus.

- **Recommended composition of influenza virus vaccines for use in the 2018 southern hemisphere influenza season**(20)

✔ an A/Michigan/45/2015 (H1N1)pdm09-like virus;
✔ an A/Singapore/INFIMH-16-0019/2016 (H3N2)-like virus; and
✔ a B/Phuket/3073/2013-like virus.

This was slightly different for the previous flu season; for example the vaccine for the northern hemisphere in 2016 – 2017 flu season contained:

✔ A/California/7/2009 (H1N1)pdm09-like virus,
✔ A/Hong Kong/4801/2014 (H3N2)-like virus and a
✔ B/Brisbane/60/2008-like virus (B/Victoria lineage).

Influenza viruses change antigenically which is why the strains included in each years vaccine may be different.

1.2.4 Antigenic shift and antigenic drift

But how exactly do Influenza viruses change? The antigenic domains in HA and NA can change either slowly and continuously, through a process named “antigenic drift” or dramatically and suddenly, by “antigenic shift”.

Antigenic drift is the result of accumulations of point mutations in the genes that code for these two proteins. The viral RNA polymerase doesn't possess proofreading ability and is thus prone to making many errors; it has been estimated that it makes approximately one error per replicated genome(21). The resultant protein variants may not be well recognized by the immune system, which makes it easier for the virions carrying them to replicate and propagate.

Antigenic shift is usually the cause of pandemics, although this is not always the case (see for example the 1918 pandemic). If a host is infected with two or more different virus subtypes, reassortment events can take place. These events refer to the exchange of genetic material between these different viruses, resulting to a new virus that has novel antigenic behavior. For example the 2009 pandemic is an example of reassortment, with segments coming from swine and a single segment coming from avian host. As the proteins these segments code for hadn't been circulating in the population before, little immunity existed in the population, or otherwise the virions could “escape the antibodies” or “escape immunity”, thus leading to pandemic.

1.3 Hemagglutinin

1.3.1 General Information

As mentioned before, hemagglutinin is one of the surface glycoproteins of the influenza virion. It has two functions: first, it binds to cell receptors, a crucial step for the virus to be able to enter the cell. Second, it mediates the fusion step of the virus envelope with the organelle membrane, after endocytosis. Its name comes from the fact that it gives the virus the ability to agglutinate (clump) red blood cells, which is a feature scientists use to detect the virus (or its absence) as well as its inhibition (or not) by antisera/antibodies etc (see Section 1.3.3).

It is coded by RNA segment 4 and is synthesized as a single protein (HA0) in the endoplasmic reticulum (ER). It is then translocated via the Golgi network to the surface of the host cell, near lipid rafts, namely part of the cell membrane that are rich in cholesterol(3). In this initial form, HA is fusion-incompetent and incorporated into the virions as a homo-trimer. It is only after each monomer is cleaved by cellular proteases to create HA1 (heavy) and HA2 (light), that it can carry out membrane fusion (see Section 1.3.4).

1.3.2 Hemagglutinin binds to sialic receptors

Hemagglutinin binds to sialic receptors found on host cells(22). Human HAs bind to sialic acids (SA) that are linked to a galactose by an α(2,6) linkage while avian HAs bind preferentially to SA linked by an α(2,3) linkage (Figure 2). SAs are monosaccharides that have a backbone of 9 carbons(23) and they, together with other glycans bind to surface proteins, thus creating glycoproteins. Viruses that infect mostly avian hosts have a preference for α2,3 – SAs, while human viruses prefer α2,6 – SAs; swine viruses bind to both. However, while the upper respiratory tract in humans mostly contains α2,6 – SAs, α2,3 – SAs are found in the lower respiratory tract, which explains why avian viruses occasionally infect humans. Also, infections of the lower respiratory tract have been correlated with more severe symptoms of the flu.

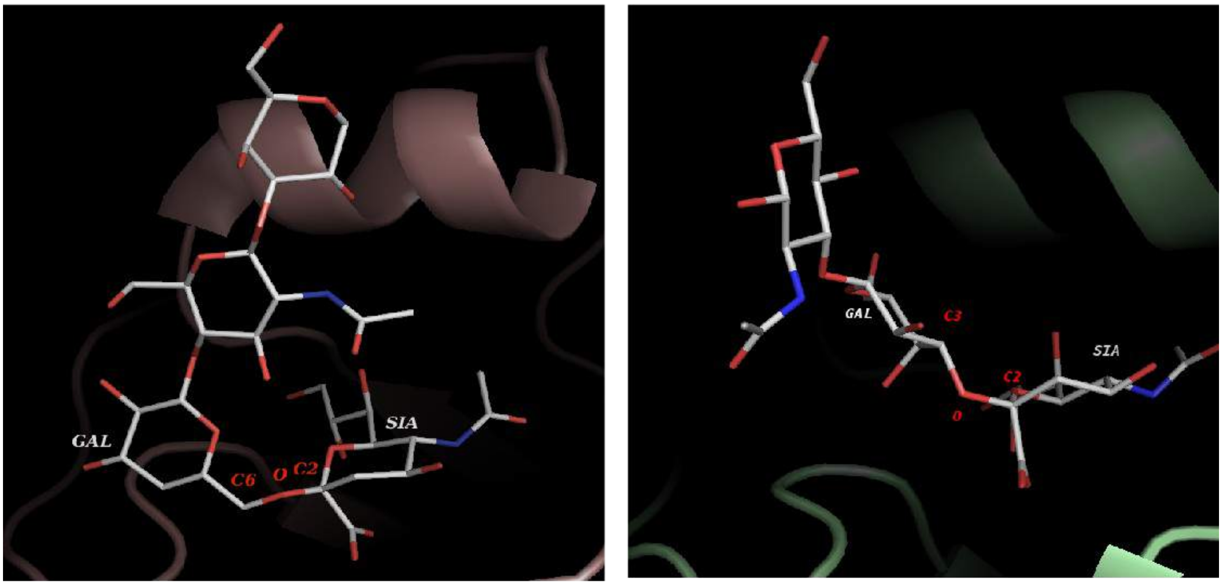


Figure 2. *Hemagglutinin in complex with sialic acid receptor analogs. Oxygen is in red, Nitrogen is in blue and Carbon in white. Left picture: LSTc, a human receptor analog, in complex with a human HAof the 2009 pandemic (PDB ID: 4JTV). Right picture: avian receptor analog, LSTa, in complex with an avian H1N1 HA (PDB ID: 3HTP). Notice how the human receptor has an α(2, 6) linkage, while the avian has an α(2, 3) linkage.*

Lactoseries tetrasaccharide c or LSTc is an example of a human α2,6-linked glycan SA receptor analog. It is a linear sialyated pentasaccharide consisting of(24):

Neu5Acα2–6Galβ1 – 4GlcNacβ1 – 3Galβ1 – 4Glc

Notice the α2,6 linkage between the SA and the galactose ring.

1.3.3 Hemagglutination assay & HI assay

Before discussing in detail the 3D structure of hemagglutinin, it is useful to comprehend two relevant to each other *in vitro* assays named Hemagglutination assay and Hemagglutination Inhibition assay, in order to be able to understand the results this project's hypotheses are based on (see Section 1.5).

As mentioned before, influenza viruses have the ability to agglutinate red(25) blood cells (RBCs), for example turkey red blood cells, because hemagglutinin can bind to their surface SA receptors. In the presence of adequate viral particles and if the HA present is able to bind erythrocytes, a network will be formed, thus making the solution red and blurry and not letting the red blood cells precipitate. This is called Hemagglutination assay (Figure 3).

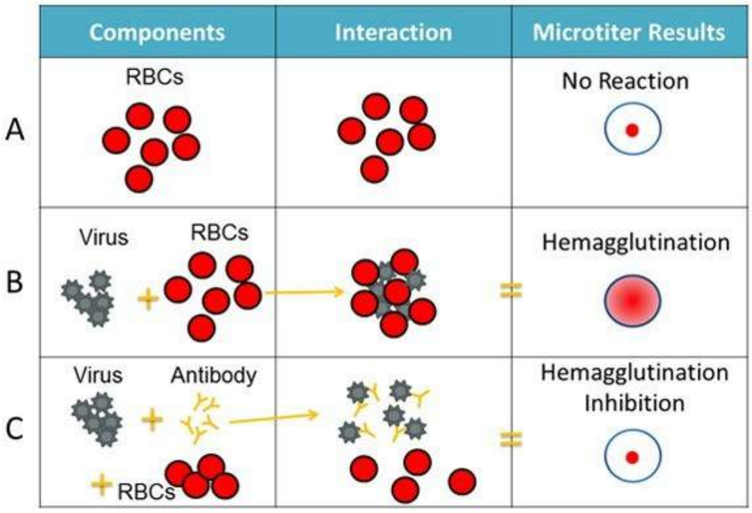


Figure 3. *Hemagglutination assay and Hemagglutinaton Inhibitionassay. A) RBCs precipitate. B) RBCs in the presence of sufficient amounts of virus with HA able to bind RBCs' SA receptors agglutinate and do not precipitate; instead a network is created and the solution turns blurry and red. C) If subtype – specific antibodies are added, the virus is inhibited from binding to RBCs, which now precipitate. (Figure from the CDC website)*

It is possible that the binding of HA to RBCs is inhibited, if subtype – specific antibodies/antisera are added in sufficient quantities and this is called Hemagglutination Inhibition assay or HI assay (Figure 3). HI titer is the reciprocal value of the highest serum dilution that completely inhibited agglutination(26) (Figure 4). Higher dilution means that the antibodies bind more efficiently to this HA. (Note that this is not always the case and in some studies partial agglutination is considered as inhibition as well.)

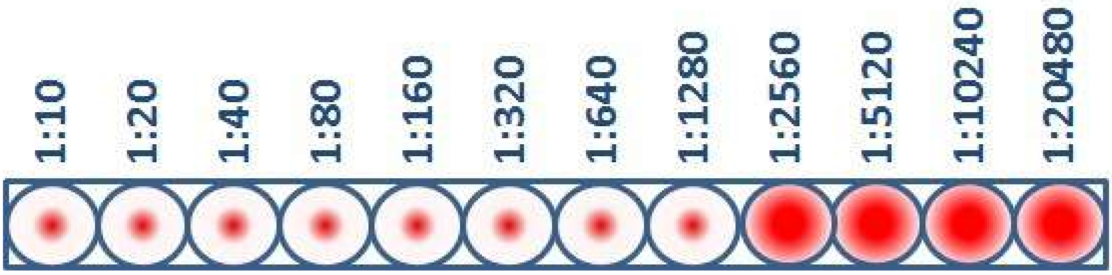


Figure 4. *HI titer. In this example, the highest dilution of antibodies that was still able to prevent agglutination was 1:1280. (Figure from the CDC website)*

HI assay is used to antigenically characterize a virus. For example the ability of a new circulating *Influenza A virus* to bind to the antibodies produced after vaccination with the seasonal flu vaccines can be tested using hemagglutination inhibition. According to public health experts, if the HI titer differs by a specific value (two dilutions or less) then the viruses are considered to be antigenically similar. In the example of a new circulating virus compared to the vaccine virus, this would mean that this vaccine would be efficient against this virus (Figure 5).

The similarity between the HAs of two different influenza viruses can of course be determined using sequencing.

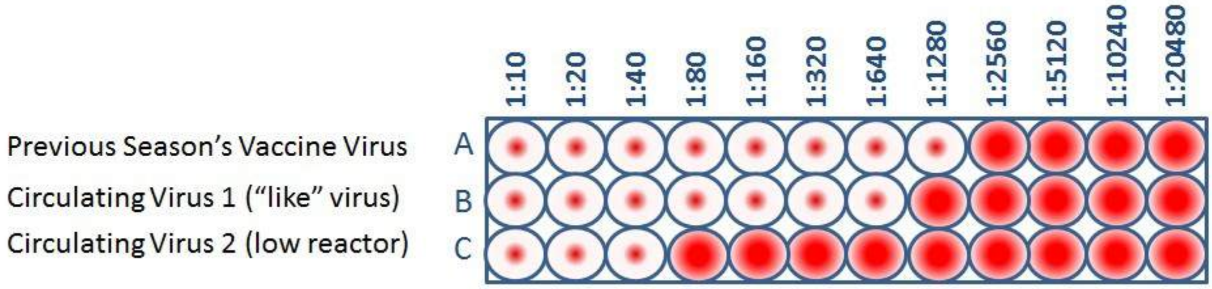

1:10
1:20
1:40
1:80
1:160
1:320
1:640
1:1280
1:2560
1:5120
1:10240
1:20480
Previous Season's Vaccine Virus
A
Circulating Virus 1 ("like" virus)
B
Circulating Virus 2 (low reactor)
C

Figure 5. *Antigenic characterization of two circulating viruses by comparison to a vaccine virus. While virus 1 is antigenically similar to the vaccine virus, virus 2 differs by a lot from the vaccine virus, as the antibodies that efficiently inhibit agglutination against the vaccine virus are not that efficient (low HI titer = 1:40) against virus 2. This means that the current vaccine will not protect against virus 2. (Figure from the CDC website)*

1.3.4 Hemagglutinin 3D Structure(26, 27, 28)

Understanding a few things about the 3D structure of hemagglutinin is an important part if one wants to understand the basis of this project.

As mentioned before HA is synthesized as precursor HA0 which is then cleaved by cellular proteases to HA1 and HA2, which are held together by a disulfide bond or an S – S bond (Figure 6). Each HA1-HA2 complex is one monomer and the homotrimer is formed inside the ER and when it is translocated via the Golgi network to the surface of the host cell(3), it anchors via the HA2 tails, with its HA1 functional parts on the outside. HA is a single pass type I integral membrane glycoprotein, meaning that the protein spans the membrane once, with its N – terminus on the extracellular side. It has a transmembrane domain and a short cytoplasmic tail. The soluble part of the protein, that is found on the outside of the envelope is 13.5nm long and consists of a stem-like structure that comprises amino acids (aa) from both HA1 and HA2 and a globular head that only has aa from the HA1 chain. This head contains the receptor

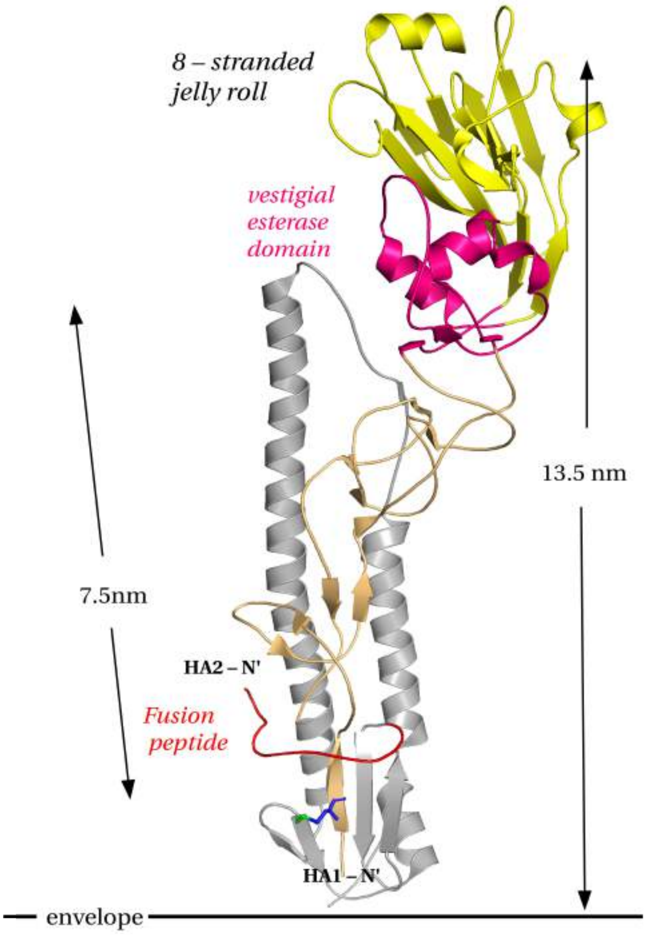


Figure 6. *The ectodomain of a hemagglutinin monomer. HA2 in gray; the jelly roll in the globular head of HA1 is in yellow; the vestigial esterase domain of HA1 is in pink; the fusion peptide is in red; the rest is in brown. The N-termini are marked. Notice how the β–sheet of five antiparallel strands is formed by four HA2 strands and one HA1 strand. In this sheet, between a β-strand of HA1 and one of HA2, an intermolecular disulfide bond is formed. PDB ID: 4JTV*

binding cite, a broad pocket in a jelly roll fold.

Underneath that, a vestigial esterase domain is found. The N terminus of HA2 is a fusion peptide which can penetrate the host cell membrane, thus initiating infection (Figure 6); it is glycine – rich and also highly conserved(30). It is notable that HA2 has one of the longest α-helices among the known globular structures (7.5nm)(28).

Regarding the secondary structure, HA1 contains approximately 8% α-helix and 32% β-sheet while HA2 has a more regular secondary structure with approximately 45% α-helix 12% β-sheet(30).

The trimeric structure (Figure 7) is formed basically by the three long HA2 α-helices, which form a coiled – coil structure and thus a core 40Å long; internal salt bridges also participate in the stabilization of the trimeric molecule. Furthermore, the globular heads are in contact and offer to the stabilization as well. Assembly of the trimer is required for stabilization of the elements in the stem-like region in the tertiary structure.

The Receptor Binding Site or RBS is located at the jelly roll fold (Figure 6), in the globular head of the protein and is responsible for binding the SAs of glycosylated receptor proteins on the surface of the target cell; the receptor binding domain is a member of the lectin superfamily(31). Each monomer has one RBS, thus each hemagglutinin molecule on the surface of a virion possesses three RBSs.

It extends from amino acid 55 to aa 271; the structures that form the RBS are 130 – loop, 150 – loop, 190 – helix and 220 – loop(29) while the amino acids that are important and highly conserved are Tyr98, Trp153, His183 and Tyr195(32) (H3 numbering), forming the base of RBS (Figure 8). Inside the RBS two disulfide bonds are found: one between Cys59 and Cys71 and one between Cys94 and Cys139 (Figure 9).

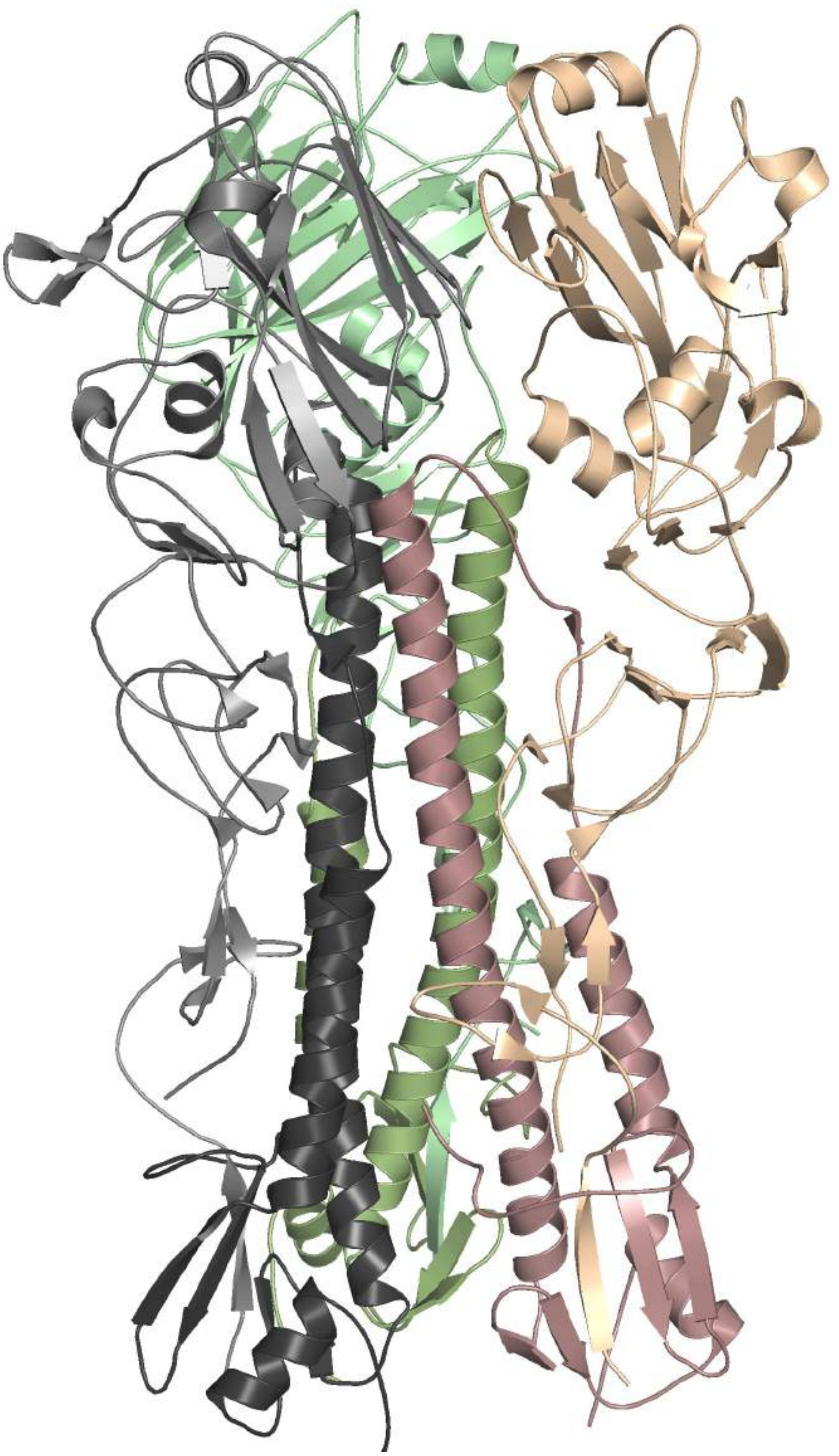


Figure 7: *Trimeric hemagglutinin. Each subunit is shown in different color; for each subunit, HA1 is shown in lighter color than HA2. Notice the coiled – coil in the middle of the molecule. PDB ID: 4JTV*

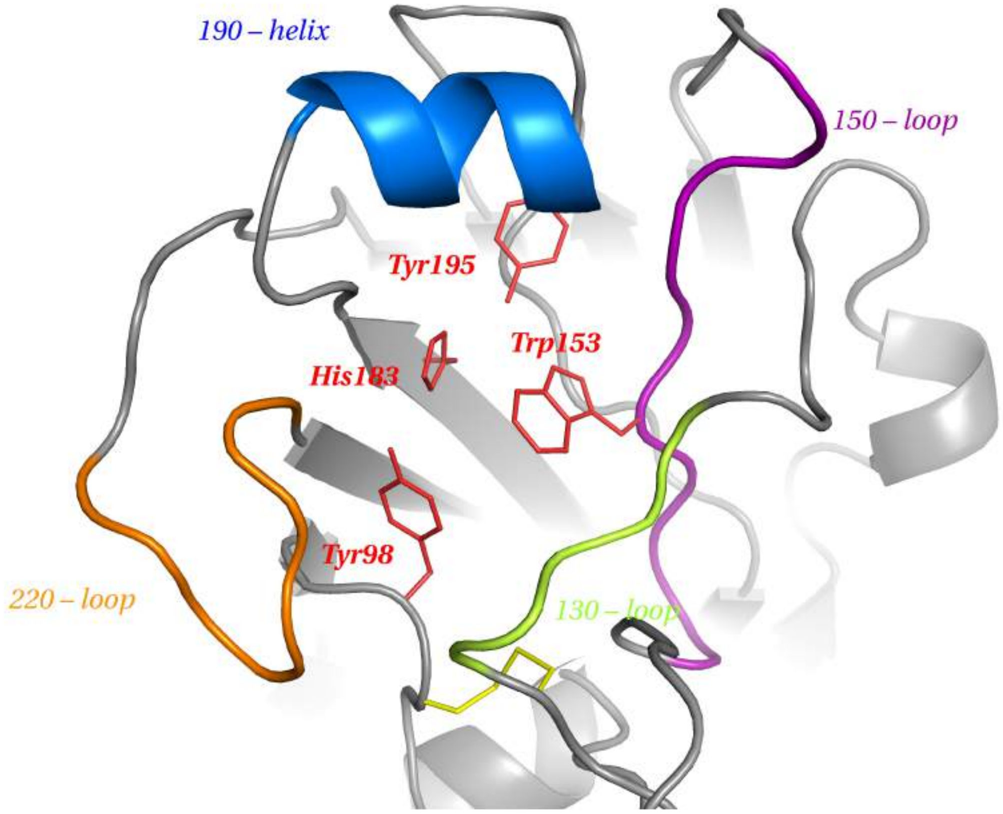


Figure 8. *Receptor Binding Site (RBS) of a hemagglutinin monomer. The important amino acids that form the base of the site are shown in red. 130, 150 and 220 loops as well as 190 – helix are shown in different colors; these form the RBS. On the bottom of the figure, a disulfide bond inside the RBS is shown (see Figure 9). PDB ID: 4JTV*

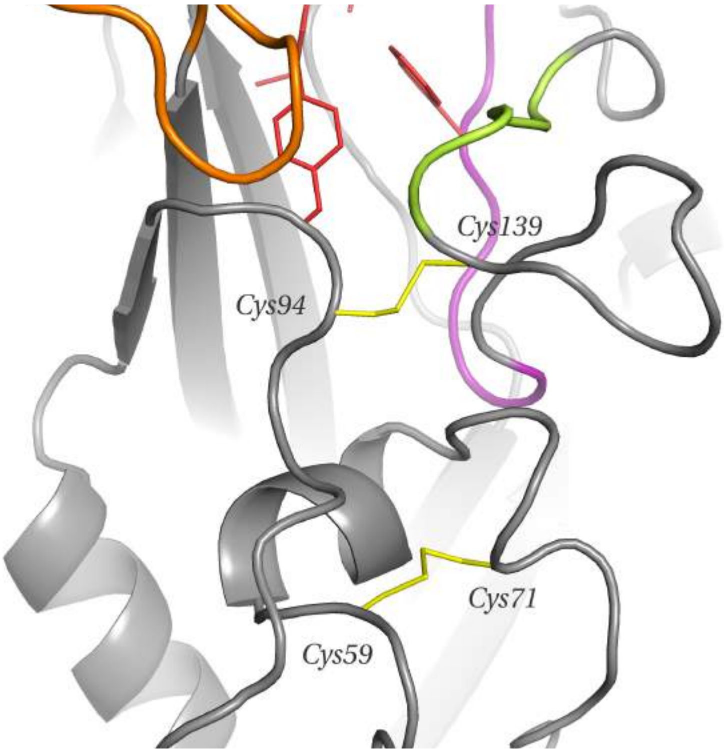


Figure 9. *Two disulfide bonds inside (Cys94 – Cys139) and underneath (Cys59 – Cys71) the main receptor binding site of a hemagglutinin monomer.*

PDB ID: 4JTV

The antigenic epitopes on the HA1 should also be mentioned. These are Sa, Sb and Cb(33), which are formed in a single protomer and Ca, which spans two monomers, with Ca1 belonging to the first and Ca2 belonging to the second monomer. These epitopes are the target of neutralizing antibodies. Some substitutions in these epitopes, that occur close to the RBS, determine major antigenic change during *Influenza virus* evolution(34). A consequence of the substitutions happening so close to the RBS is that they can affect HA function and then, only when co-mutations occur it is possible for the virus to retain its ability to bind SAs and thus retain replication efficiency; namely the antigenic evolution may be slowed down because of the resulting reduction in receptor binding function, since the substitutions occur in key positions near RBS.

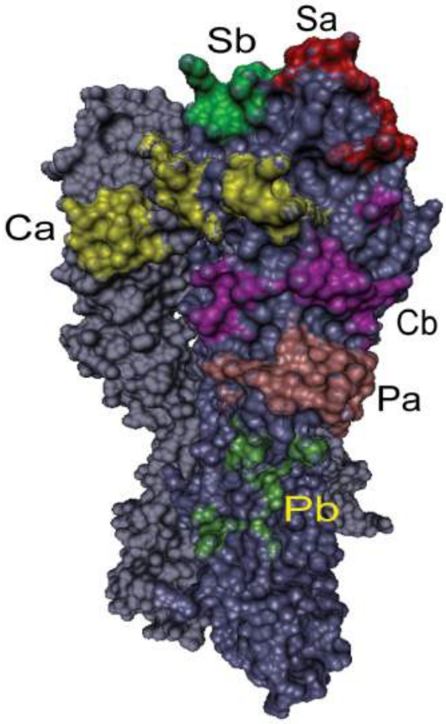


Figure 10. *Antigenic epitopes on a hemagglutinin molecule using the nomenclature suggested by Gerhard et al., demonstrated on a 1934 H1 hemagglutinin (PDB ID: 1RVZ)*(35)

1.3.4 (H3) Numbering

Before discussing the structural determinants of receptor specificity, namely the host tropism of Influenza viruses, it is useful to explain the numbering used for hemagglutinin sequences. As a convention in the field, the H3 numbering is used, in order to be able to make comparisons across different subtypes. In 2014, Burke and Smith recommended a numbering scheme(36) for Influenza A subtypes; based on known HA structures they defined aa that possess equivalent structure and function across all subtypes. Table A1 in the Appendix contains all the numbering conversions that were needed during this project.

1.3.5 Structural determinants of receptor specificity

Specific amino acid substitutions in HA lead to changes in receptor specificity and thus changes in host specificity and tropism. Different amino acids determine the receptor specificity in each subtype; here the focus is on H1. Positions 190 and 225 have been proven to be important for the receptor specificity of H1 and different combinations yield different specificity (Table 2). 190 position is occupied by either glutamic acid (E) or Aspartic acid (D) and 225 by either Glycine (G) or Aspartic acid or Glutamic acid.

D190/D225 are found in human HA and they interact with GlcNac3 and Gal2 respectively. These interactions are absent in the avian HA – receptor complexes because of the extended conformation the avian receptor adopts (Figure 2).

Regarding the dual specificity of D190/G225, that resulted from a human D190/D225 HA, two different models have been proposed, which are both accepted currently. Either a loss of a salt bridge between D225 and K222 relaxes the 220-loop and allows to Q226 to interact with the avian receptor(27) or the D225G substitution results in a general conformational change that relaxes the 220-loop(37).

190 position	225 position	Specificity
Glutamic acid	Glycine	avian & human
Glutamic acid	Aspartic acid	avian & human
Aspartic acid	Glycine	avian & human
Aspartic acid	Aspartic acid	human
Aspartic acid	Glutamic acid	human

Table 2. *Amino acid composition in 190 and 225 positions in H1 isolates. While D190/D225 has human specificity, D190/G225, which was found in isolates during late 1918 and 2009 pandemics has dual specificity.*

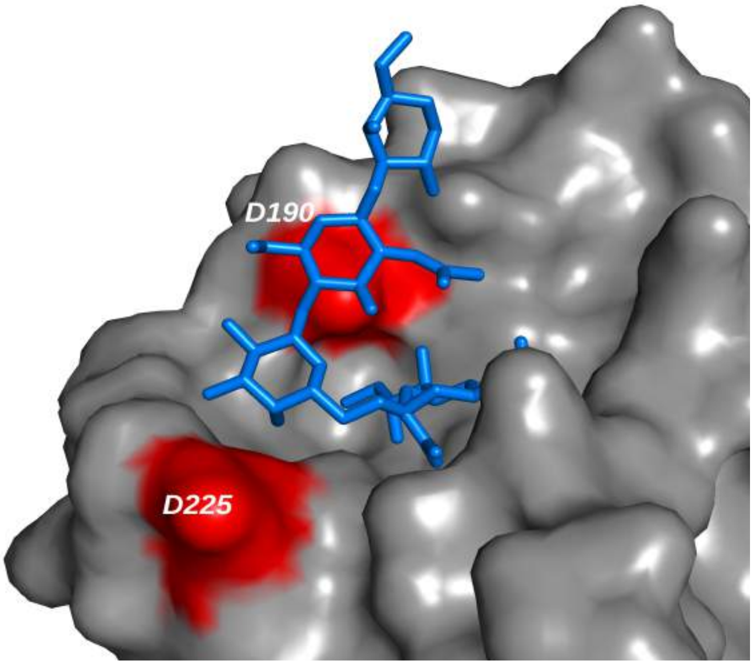


Figure 11. *An example of D190/D225 combination in a hemagglutinin H1 molecule. Notice how both aspartic acids face the ligand. Remember that 220-loop and 190-helix have already be mentioned as important structures that form the receptor binding site. This hemagglutinin molecule can only bind to human receptors (namely α2,6 SAs). PDB ID: 4JTV*

1.4 Docking – Autodock Vina

Let's take a breath from all this virology and structural biology information to discuss docking. Molecular docking is a computational method aiming to calculate noncovalent binding of macromolecules (for example protein – protein interactions) or of a macromolecule and a small molecule using their 3D structures. In the later case, the macromolecule (for example the protein) is termed as "receptor" and the small molecule as "ligand", thus the procedure is called "protein – ligand docking". Since the protein this project is about is hemagglutinin, in a docking study the receptor could be hemagglutinin and the ligand an α2,6 SA, an α2,3 SA, an analog of theirs etc.

Docking programs use scoring functions that attempt to approximate the chemical potentials which determine the preferred binding conformation and the free energy of binding. Autodock Vina (hereinafter vina) specifically is a C++ program which uses an algorithm that attempts to minimize the sum of both inter– and intra–molecular contributions. After this, the resultant conformations are ranked from the lowest to the highest sum. The free energy of binding is predicted based on the inter– molecular part of the scoring function. The scoring function of Autodock Vina is mostly based on "machine learning" approaches rather than pure physics – based and it was tuned using PDBbind, namely a large data set, which by the time the paper describing Autodock Vina was published (2010), contained a number of complexes between 1,091 (in 2004) and 2,897 (in 2012)(38).

There are a lot of assumptions behind this algorithm; first of all, the protonation state and charge of the molecules is considered the same between the unbound and bound states. Second, the biggest part of the receptor is considered rigid (only a few flexible residues are allowed) while the ligand can be treated as flexible, with the number of active rotatable bonds ranging from 0 to 32. This clashes with what structural biochemists know about induced fit; according to this, the protein structure constantly changes during binding of a substrate.

1.5 The goal

The idea for this project came from a Koel et al. publication (hence referred as "Koel paper")(26), in which the attempts to identify amino acid substitutions near the RBS supporting antigenic change of Influenza A viruses from 2009 pandemic are described. In other words, the researchers explored molecular changes that contribute to antibody escape of A(H1N1)pdm09 virus from ferret antisera and some human antisera after primary infection, exploiting both *in vitro* and *in vivo* approaches.

Part of this study was to test if the substitutions causing antibody escape altered the receptor binding efficiency and/or specificity. In brief, the mutants were tested with Hemagglutination Assays for their ability to agglutinate turkey red blood cells (TRBCs), either normal or stripped from their SAs and resialylated to contain either α2,3– or α2,6–SAs (Table 3).

Our goal was to obtain these relative changes in binding (or absence of changes) in computational simulations using Autodock Vina. More specifically, I attempted to test if using a hemagglutinin 3D structure and a human receptor analog, the differences in the calculated binding efficiency among wild type HA and mutants would correspond to the ones obtained in the experiments (Table 3) described in Koel et al., 2014. and what the changes in the ligands' poses would be.

2. Preparing the simulations

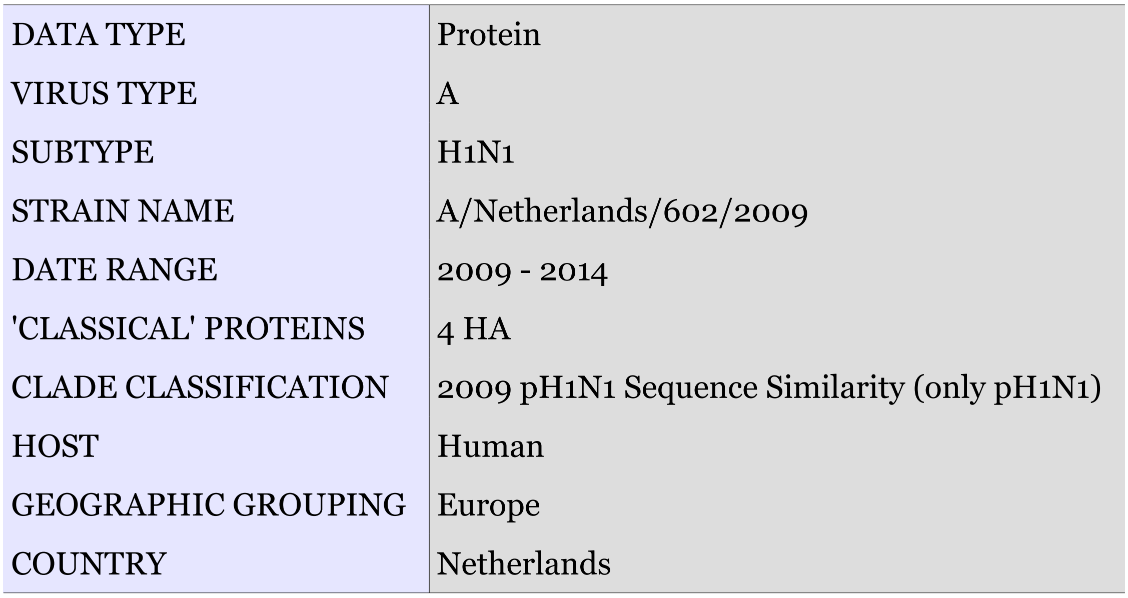
2.1 Selecting the PDB structure

The experiments described in the Koel paper were performed using the strain A/Netherlands/602/2009(39) to represent the antigenic properties of A(H1N1)pdm09 viruses. It wasn't possible to find a PDB structure of this strain, but one of a hemagglutinin-LSTc complex (PDB ID: 4JTV) which is the 3D structure of an A/California/04/2009 strain (Uniprot ID: C3W5S1_I09A0) complexed with the LSTc human receptor analog was found.

	HI titer			
Antigen	**TRBC**	**VCNA**	**α2,3–TRBC**	**α2,6–TRBC**
A/Netherlands/602/09	512	0	0	32
127DT mutant	128	0	0	32
153KE	128	0	1	64
155GE	512	0	1	16
156ND	512	0	2	128
156NG	256	0	0	4
156NS	256	0	0	2
156NY	64	0	0	32
224EA	128	0	0	64
152VT156NS	128	0	0	8
155GE224EA	1,024	0	0	1,024
A/Vietnam/1194/2004	128	0	256	0
A/Netherlands/213/2003	256	0	0	256

Table 3. *Agglutination of TRBCs by viruses with wild–type or mutant HAs. The mutations shown are in H1pdm09 numbering and have been found (Koel et al., 2014) to escape immunity(26). The TRBC column refers to unmodified TRBCs. VCNA is an enzyme that removes SAs from cells, thus no virus agglutinates cells that have been treated with VCNA. A/Netherlands is the wild – type virus used in this study and the mutations were done on this backbone. A/Vietnam/1194/2004 is an avian influenza virus and A/Netherlands/213/2003 is a human virus.*

In order to investigate if this PDB structure could be used for our simulations, the level of sequence similarity between A/Netherlands/602/09 HA used in the Koel paper and A/California/04/2009 HA needed to be investigated. Koel et al. didn't provide the exact A/Netherlands/602/09 sequence, so it was attempted to find which one is more probable to have been used. Data was obtained from the NIAID Influenza Research Database (IRD) [Zhang Y, et al. (2017)] through the web site at http://www.fludb.org. IRD was searched to find the different A/Netherlands/602/09 sequences available, with the selection parameters shown in Table 4.



DATA TYPE	Protein
VIRUS TYPE	A
SUBTYPE	H1N1
STRAIN NAME	A/Netherlands/602/2009
DATE RANGE	2009 - 2014
'CLASSICAL' PROTEINS	4 HA
CLADE CLASSIFICATION	2009 pH1N1 Sequence Similarity (only pH1N1)
HOST	Human
GEOGRAPHIC GROUPING	Europe
COUNTRY	Netherlands

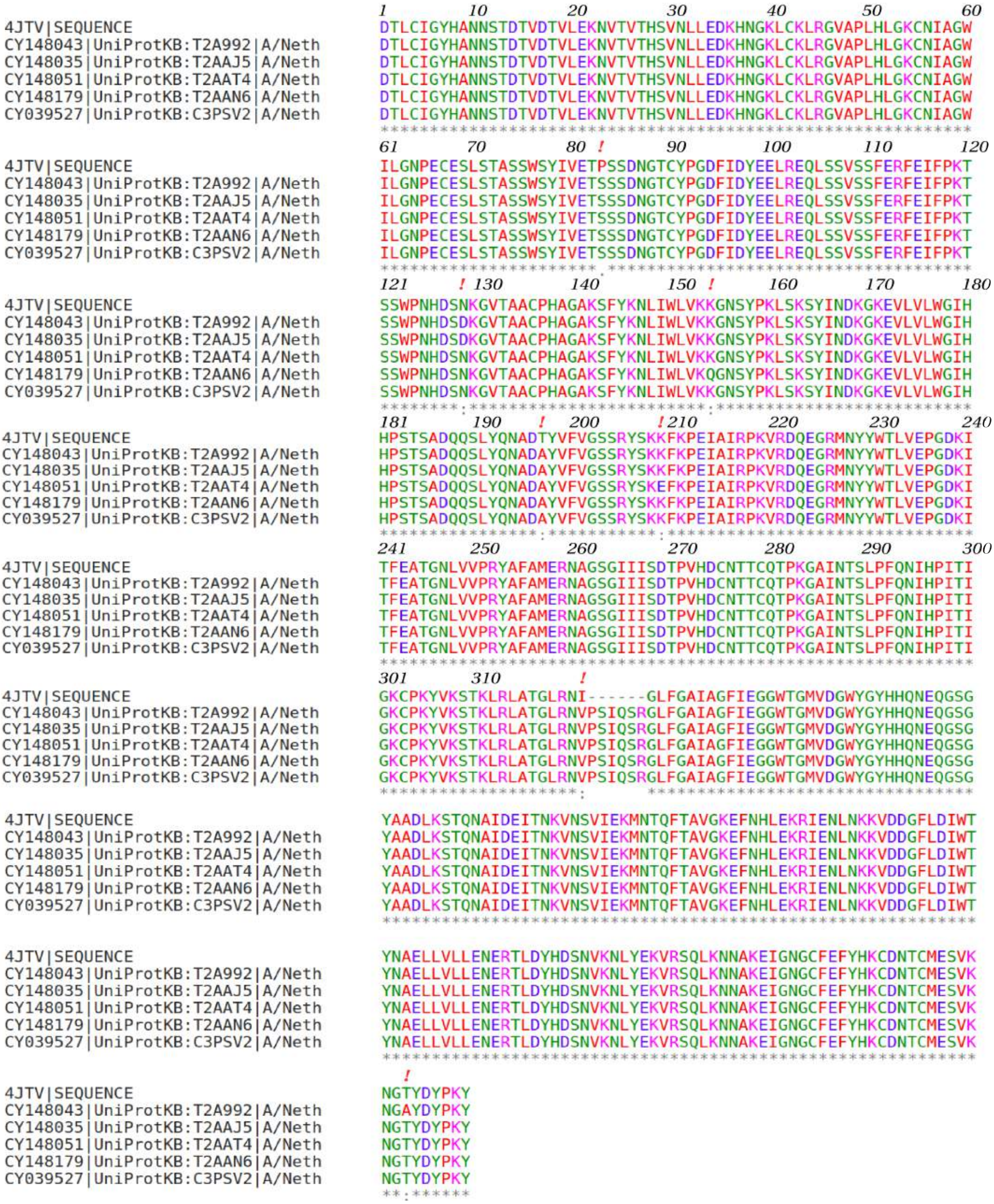
Table 4. *Selection parameters used for searching IRD. 4 HA refers to HA, which is coded by RNA-segment 4 of Influenza virus; clade classification was set to proteins similar to 2009 original reassortant pandemic strains; date range was set from 2009 (the date of the last pandemic) to 2014 (the date the Koel paper was submitted).*

This search returned 21 protein sequences (22/2/2017) of which the identical entries were removed leaving 11 unique entries. The next step was to rule out, based on the amino acid composition for the positions referred in the Koel paper, those entries that couldn't be the ones that were used for the experiments. This procedure ruled out 6 out of the 11 sequences. The remaining 5 A/Netherlands/602/2009 sequences left are the candidates to have been used in the Koel paper. Aligning these with the sequence used to obtain 4JTV structure, with Multiple Sequence Alignment (40) using MUSCLE (parameters set to default), yielded the results shown in Figure 12.

A sum of the multiple sequence alignment results for HA1 chain is presented in Table 5. Notice how positions 83, 197 and 321 have the same amino acid in all 5 A/Netherlands/602/09 sequences and a different one in 4JTV, while positions 129 and 154 are identical among 4JTV and some of the A/Netherlands/602/09 sequences.

After pair-wise alignment between 4JTV and the five different A/Netherlands/602/09 sequences, the identities ranged from 317/321 to 318/321 and the similarities for all alignments were 319/321, while both, when transformed to percentages were 99%.

These positions were also plotted on the 4JTV structure to investigate their relative position to the RBS on which the downstream simulations are focused (Figure 13 & 14).



```
                                    1        10        20        30        40        50        60
4JTV|SEQUENCE                       DTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGW
CY148043|UniProtKB:T2A992|A/Neth    DTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGW
CY148035|UniProtKB:T2AAJ5|A/Neth    DTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGW
CY148051|UniProtKB:T2AAT4|A/Neth    DTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGW
CY148179|UniProtKB:T2AAN6|A/Neth    DTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGW
CY039527|UniProtKB:C3PSV2|A/Neth    DTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGW
                                    ************************************************************
                                    61       70        80 !      90        100       110       120
4JTV|SEQUENCE                       ILGNPECESLSTASSWSYIVETPSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKT
CY148043|UniProtKB:T2A992|A/Neth    ILGNPECESLSTASSWSYIVETSSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKT
CY148035|UniProtKB:T2AAJ5|A/Neth    ILGNPECESLSTASSWSYIVETSSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKT
CY148051|UniProtKB:T2AAT4|A/Neth    ILGNPECESLSTASSWSYIVETSSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKT
CY148179|UniProtKB:T2AAN6|A/Neth    ILGNPECESLSTASSWSYIVETSSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKT
CY039527|UniProtKB:C3PSV2|A/Neth    ILGNPECESLSTASSWSYIVETSSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKT
                                    ********************** *************************************
                                    121     !130       140       150 !     160       170       180
4JTV|SEQUENCE                       SSWPNHDSNKGVTAACPHAGAKSFYKNLIWLVKKGNSYPKLSKSYINDKGKEVLVLWGIH
CY148043|UniProtKB:T2A992|A/Neth    SSWPNHDSDKGVTAACPHAGAKSFYKNLIWLVKKGNSYPKLSKSYINDKGKEVLVLWGIH
CY148035|UniProtKB:T2AAJ5|A/Neth    SSWPNHDSDKGVTAACPHAGAKSFYKNLIWLVKKGNSYPKLSKSYINDKGKEVLVLWGIH
CY148051|UniProtKB:T2AAT4|A/Neth    SSWPNHDSNKGVTAACPHAGAKSFYKNLIWLVKKGNSYPKLSKSYINDKGKEVLVLWGIH
CY148179|UniProtKB:T2AAN6|A/Neth    SSWPNHDSNKGVTAACPHAGAKSFYKNLIWLVKQGNSYPKLSKSYINDKGKEVLVLWGIH
CY039527|UniProtKB:C3PSV2|A/Neth    SSWPNHDSNKGVTAACPHAGAKSFYKNLIWLVKKGNSYPKLSKSYINDKGKEVLVLWGIH
                                    ********:************************:**************************
                                    181      190    !  200      !210       220       230       240
4JTV|SEQUENCE                       HPSTSADQQSLYQNADTYVFVGSSRYSKKFKPEIAIRPKVRDQEGRMNYYWTLVEPGDKI
CY148043|UniProtKB:T2A992|A/Neth    HPSTSADQQSLYQNADAYVFVGSSRYSKKFKPEIAIRPKVRDQEGRMNYYWTLVEPGDKI
CY148035|UniProtKB:T2AAJ5|A/Neth    HPSTSADQQSLYQNADAYVFVGSSRYSKKFKPEIAIRPKVRDQEGRMNYYWTLVEPGDKI
CY148051|UniProtKB:T2AAT4|A/Neth    HPSTSADQQSLYQNADAYVFVGSSRYSKEFKPEIAIRPKVRDQEGRMNYYWTLVEPGDKI
CY148179|UniProtKB:T2AAN6|A/Neth    HPSTSADQQSLYQNADAYVFVGSSRYSKKFKPEIAIRPKVRDQEGRMNYYWTLVEPGDKI
CY039527|UniProtKB:C3PSV2|A/Neth    HPSTSADQQSLYQNADAYVFVGSSRYSKKFKPEIAIRPKVRDQEGRMNYYWTLVEPGDKI
                                    ****************:***********:*******************************
                                    241      250       260       270       280       290       300
4JTV|SEQUENCE                       TFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPITI
CY148043|UniProtKB:T2A992|A/Neth    TFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPITI
CY148035|UniProtKB:T2AAJ5|A/Neth    TFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPITI
CY148051|UniProtKB:T2AAT4|A/Neth    TFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPITI
CY148179|UniProtKB:T2AAN6|A/Neth    TFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPITI
CY039527|UniProtKB:C3PSV2|A/Neth    TFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPITI
                                    ************************************************************
                                    301      310        !
4JTV|SEQUENCE                       GKCPKYVKSTKLRLATGLRNI------GLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSG
CY148043|UniProtKB:T2A992|A/Neth    GKCPKYVKSTKLRLATGLRNVPSIQSRGLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSG
CY148035|UniProtKB:T2AAJ5|A/Neth    GKCPKYVKSTKLRLATGLRNVPSIQSRGLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSG
CY148051|UniProtKB:T2AAT4|A/Neth    GKCPKYVKSTKLRLATGLRNVPSIQSRGLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSG
CY148179|UniProtKB:T2AAN6|A/Neth    GKCPKYVKSTKLRLATGLRNVPSIQSRGLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSG
CY039527|UniProtKB:C3PSV2|A/Neth    GKCPKYVKSTKLRLATGLRNVPSIQSRGLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSG
                                    ********************:      *********************************

4JTV|SEQUENCE                       YAADLKSTQNAIDEITNKVNSVIEKMNTQFTAVGKEFNHLEKRIENLNKKVDDGFLDIWT
CY148043|UniProtKB:T2A992|A/Neth    YAADLKSTQNAIDEITNKVNSVIEKMNTQFTAVGKEFNHLEKRIENLNKKVDDGFLDIWT
CY148035|UniProtKB:T2AAJ5|A/Neth    YAADLKSTQNAIDEITNKVNSVIEKMNTQFTAVGKEFNHLEKRIENLNKKVDDGFLDIWT
CY148051|UniProtKB:T2AAT4|A/Neth    YAADLKSTQNAIDEITNKVNSVIEKMNTQFTAVGKEFNHLEKRIENLNKKVDDGFLDIWT
CY148179|UniProtKB:T2AAN6|A/Neth    YAADLKSTQNAIDEITNKVNSVIEKMNTQFTAVGKEFNHLEKRIENLNKKVDDGFLDIWT
CY039527|UniProtKB:C3PSV2|A/Neth    YAADLKSTQNAIDEITNKVNSVIEKMNTQFTAVGKEFNHLEKRIENLNKKVDDGFLDIWT
                                    ************************************************************

4JTV|SEQUENCE                       YNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNAKEIGNGCFEFYHKCDNTCMESVK
CY148043|UniProtKB:T2A992|A/Neth    YNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNAKEIGNGCFEFYHKCDNTCMESVK
CY148035|UniProtKB:T2AAJ5|A/Neth    YNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNAKEIGNGCFEFYHKCDNTCMESVK
CY148051|UniProtKB:T2AAT4|A/Neth    YNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNAKEIGNGCFEFYHKCDNTCMESVK
CY148179|UniProtKB:T2AAN6|A/Neth    YNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNAKEIGNGCFEFYHKCDNTCMESVK
CY039527|UniProtKB:C3PSV2|A/Neth    YNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNAKEIGNGCFEFYHKCDNTCMESVK
                                    ************************************************************
                                      !
4JTV|SEQUENCE                       NGTYDYPKY
CY148043|UniProtKB:T2A992|A/Neth    NGAYDYPKY
CY148035|UniProtKB:T2AAJ5|A/Neth    NGTYDYPKY
CY148051|UniProtKB:T2AAT4|A/Neth    NGTYDYPKY
CY148179|UniProtKB:T2AAN6|A/Neth    NGTYDYPKY
CY039527|UniProtKB:C3PSV2|A/Neth    NGTYDYPKY
                                    **:******
```

Figure 12. *Multiple sequence alignment results. Five A/Netherlands/602/09 HA sequences that correspond to the information provided by the Koel paper were aligned with the HA sequence used to determine the 4JTV PDB structure. The positions that differ among them are marked. Please notice that only part of the HA2 chain is included – the part missing was identical between the five A/Netherlands/602/09 sequences and was missing from the 4JTV sequence.*

	Sequences		
Position (chain HA1)	4JTV	A/Netherlands/602/09	
83	P	S	
129	N	N: 3/5	D: 2/5
154	K	K: 4/5	Q: 1/5
197	T	A	
321	I	V	

Table 5. *Differences among 4JTV and A/Netherlands/602/09 HA sequences. The color code used for 4JTV HA corresponds to the one used in Figures 13 and 14.Notice how 83, 197 and 321 are different between 4JTV HA sequence and all the A/NL/602/09 sequences, while 129 and 154 are the same for 4JTV and some of the A/NL/602/09 sequences.*

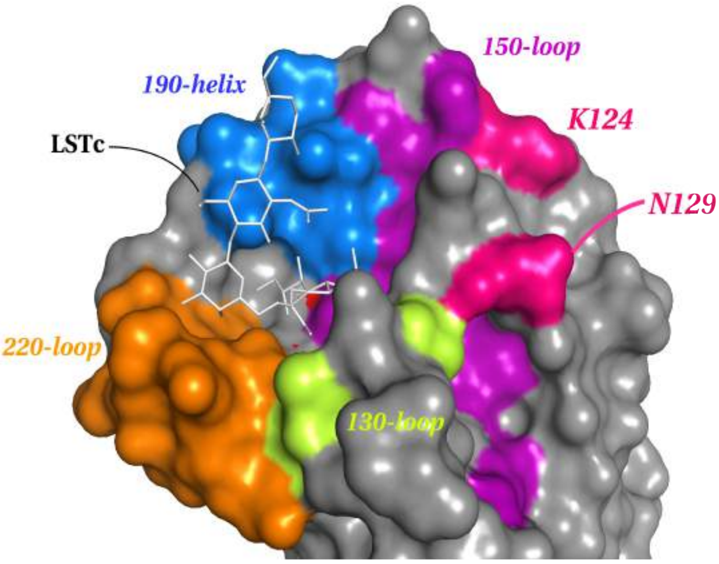


Figure 13. *The relative position of amino acids 124&129 (shown in pink) and RBS using a surface display. Remember that 4JTV and some A/Netherlands/602/09 have the exact same amino acid in each of these positions (namely they just differ among the A/NL/602/09 sequences).The color code used is the same as in Figure 8. PDB ID: 4JTV*

In conclusion, the differences between the 4JTV sequence and the A/NL/602/09 sequences are either far from the RBS (83 and 321) or not pointing towards the ligand. However, 4JTV has a proline at position 83 while A/NL/602/09 sequences have a serine. This is not only a difference in hydrophobicity (proline is non-polar and serine is polar) but can also cause major differences in the general structure, since it is known that proline can cause major changes on the geometry of the backbone. What's more, amino acids 197, 124 and 129 may not point towards the ligand but are inside or near the RBS, making the potential candidates to affect the overall structure of the RBS thus causing differences in binding efficiency and/or specificity between A/NL/602/09 HA and A/California/04/09.

After these analyses it was hypothesized that using 4JTV, which is the HA coming from an A/California/04/09 strain wouldn't significantly affect our results, which aim to reproduce the *in vitro* experiments of Koel et al. described in Section 1.5, in which an A/Netherlands/602/09 HA was used.

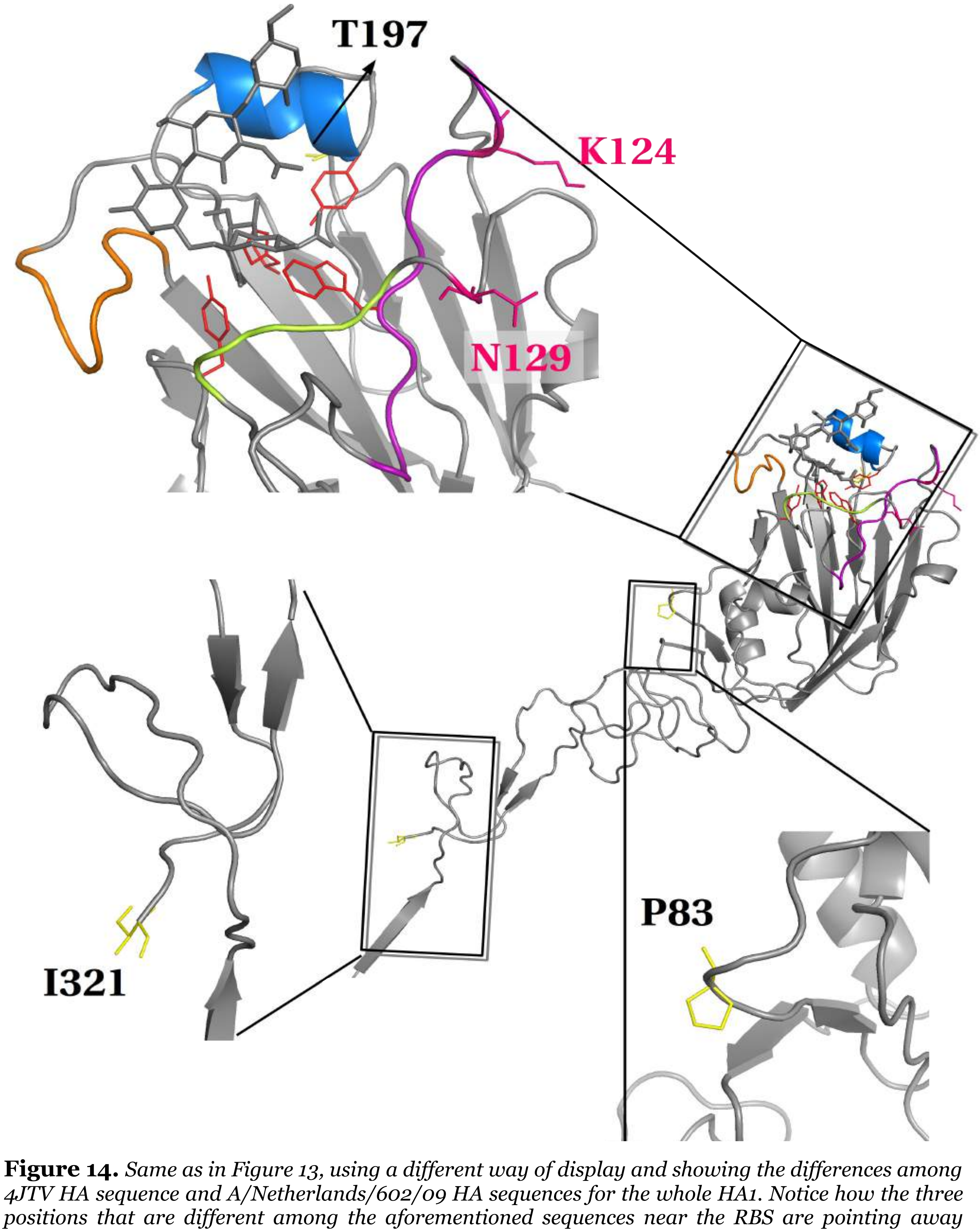


Figure 14. *Same as in Figure 13, using a different way of display and showing the differences among 4JTV HA sequence and A/Netherlands/602/09 HA sequences for the whole HA1. Notice how the three positions that are different among the aforementioned sequences near the RBS are pointing away from the RBS and not to the ligand. Color code for the RBS is the same as in Figure 8; amino acids that are different between 4JTV and all A/NL/602/09 are shown in yellow; positions with the same amino acid in 4JTV and some of the A/NL/602/09 are in pink. PDB ID: 4JTV*

2.2 Preparing the structures for the docking simulations with rigid molecules

As it was discussed in Section 1.4 in the case of a docking simulation, the protein used is the “receptor” and the small molecule which binding to the protein is tested is the “ligand”. In this project, “protein – ligand” docking was performed, with the receptor being a hemagglutinin molecule (and as shown in the previous section, the PDB structure 4JTV was used) and the ligand being the human receptor analog LSTc, which structure was obtained from the same PDB file, 4JTV, since, as mentioned in Section 1.9, 4JTV is the 3D structure of an A/California/04/2009 strain (Uniprot ID: C3W5S1_I09A0) complexed with the LSTc human receptor analog. **One should pay attention, that since 4JTV is the structure of a protein – ligand complex, the ligand, as well as the participating amino acids have acquired the conformations the adopt in the bound state.**

The process followed for the preparation has elements from protocols about docking in genera and from papers about docking experiments using hemagglutinin specifically(35).

2.2.1 Preparing the receptor

The asymmetric unit of 4JTV is shown in Figure 15. The one homotrimer in the unit cell consists of chains A, C and E (which are HA1) and B, D, F (which are HA2). The second one has chains G-L. I focused on the first trimer, and worked only with that in the simulations, as more of the rings of the ligand were visible.

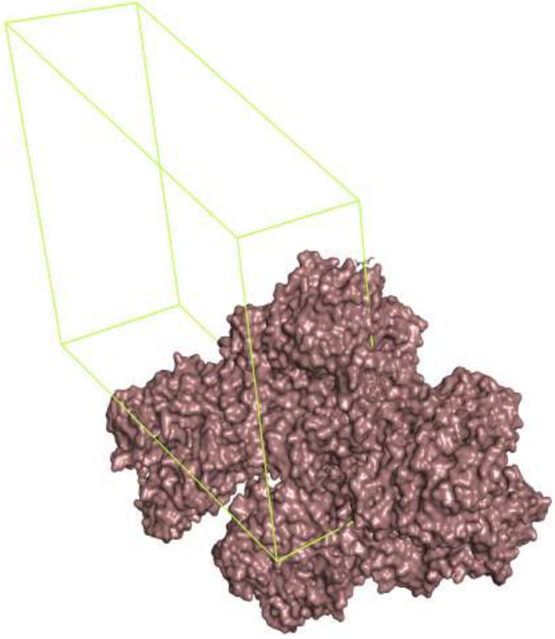


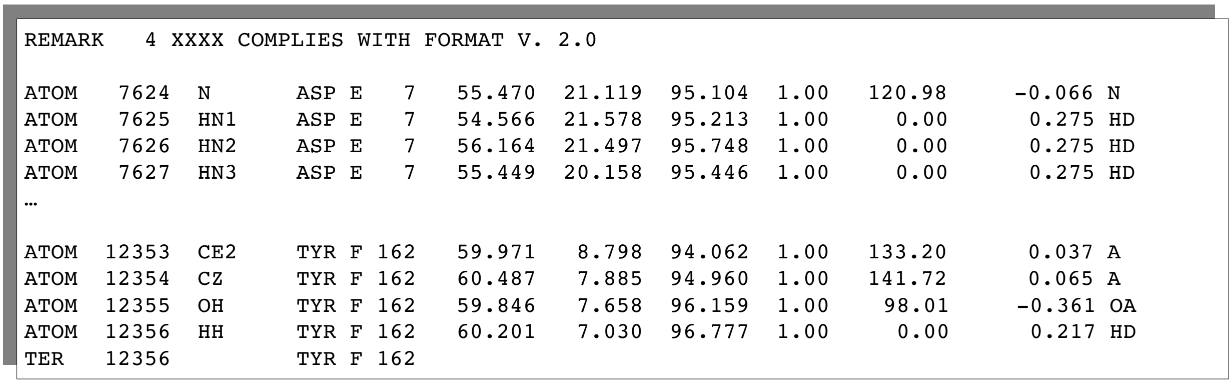
Figure 15. *4JTV asymmetric unit (shown in ugly pink). PDB ID: 4JTV*

For the simulations, only the first trimer from the assymetric unit was used, as more of the rings of the ligand are visible in this one.

Docking simulations were performed using the trimeric form as well as the single monomers. The process for the preparation of one monomer, consisting of chains E and F, will be presented; it is the same for the trimer.

First, using Autodock Tools, chains A, B, C and D were deleted, leaving only E and F. Then, the LSTc ligand as well as all water molecules were removed except the ones inside the RBS mediating hydrogen bonds between the ligand and the protein and more specifically those within a radius of 6.0 Ångström from the ligand's rings. After, hydrogens were added and at this stage the protonation state of H183 (see Figure 8) to the known state, as previously done(41): only one hydrogen is assigned, to the nitrogen atom at the epsilon position. Then the the non-polar hydrogens were merged; AutoDock Vina uses them to assign the hydrogen bonding state of the heteroatoms, but does not use explicit hydrogens during the docking. Lastly, a simplified typing of atoms (including identification of aromatic and aliphatic carbon atoms and identification of the hydrogen bonding state of heteroatoms) was assigned, as well as charges, although these last ones are ignored by vina. The last three steps are done automatically by AutodockTools, by running "Grid → Macromolecule".

This process produces a file in PDBQT format(42), that includes the atomic coordinates, the partial charges (ignored by vina) and the simplified Autodock 4 atom-types (Figure 16).



```
REMARK   4 XXXX COMPLIES WITH FORMAT V. 2.0

ATOM   7624  N      ASP E   7   55.470  21.119  95.104  1.00   120.98     -0.066 N
ATOM   7625  HN1    ASP E   7   54.566  21.578  95.213  1.00     0.00      0.275 HD
ATOM   7626  HN2    ASP E   7   56.164  21.497  95.748  1.00     0.00      0.275 HD
ATOM   7627  HN3    ASP E   7   55.449  20.158  95.446  1.00     0.00      0.275 HD
…

ATOM  12353  CE2    TYR F 162   59.971   8.798  94.062  1.00   133.20      0.037 A
ATOM  12354  CZ     TYR F 162   60.487   7.885  94.960  1.00   141.72      0.065 A
ATOM  12355  OH     TYR F 162   59.846   7.658  96.159  1.00    98.01     -0.361 OA
ATOM  12356  HH     TYR F 162   60.201   7.030  96.777  1.00     0.00      0.217 HD
TER   12356         TYR F 162
```

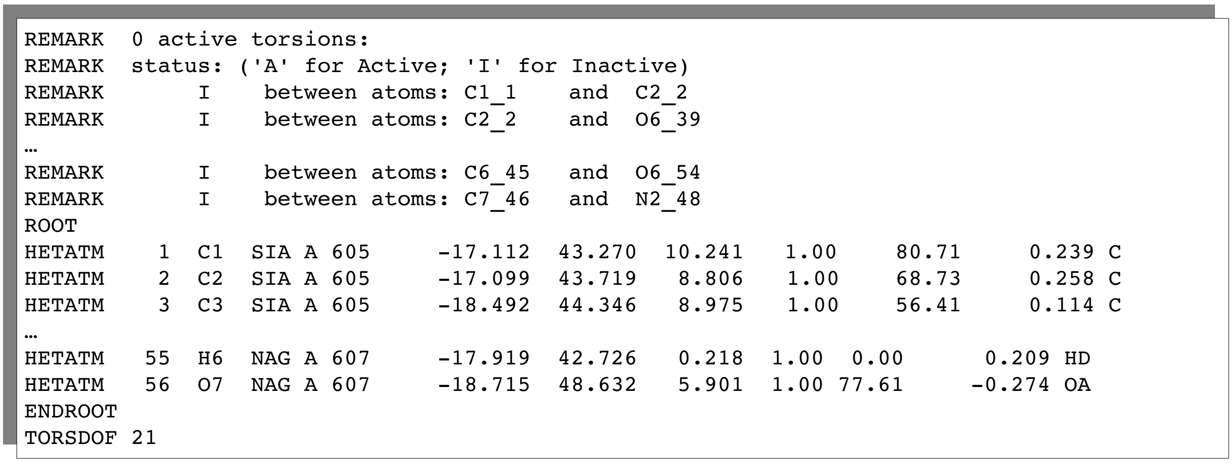
Figure 16. *Part of the PDBQT file of chain E and F (namely one monomer) of the hemagglutinin molecule of the 4JTV PDB structure. The third column includes the atom names assigned to atoms in the PDB file; the twelfth column includes the partial charges and the last one Autock 4 atom-types.*

The importance of checking for erroneous amino acids in the structure should also be mentioned. In chain A, E230 (4JTV numbering, E224 in H1 numbering) was recorded as an alanine and since this position is inside the RBS, it can affect the results regarding binding affinity. The amino acid had to be corrected from Ala to Glu and all the simulations and their analyses regarding chain A had to be repeated.

2.2.2 Preparing the ligand

The process of preparing the ligand is similar to the one for the receptor. Only the process for the ligand bound to chain E will be described, although it is the same for the ligands bound to the other chains of the trimer.

After reading 4JTV in Autodock Tools, all atoms except for the ligand of chain E were removed. Then hydrogens were added and the non-polar ones were merged. ADTools then assigned charges and appropriate atom types. Finally its torsion tree was set to all of the bonds be non-rotatable, namely the ligand was treated as rigid. A part of the resulting PDB file for the ligand of chain E, which is set to be treated as rigid is shown in Figure 17. Note that the ligand coordinates are eventually randomized, to remove any bias resulting from its initial position in the RBS. Some of the simulations were, however performed without randomizing first and no statistically significant difference was observed (data not shown). The randomization step remained in our protocol though, since the creators of vina suggest it and it is not time – consuming.



```
REMARK  0 active torsions:
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK       I    between atoms: C1_1    and  C2_2
REMARK       I    between atoms: C2_2    and  O6_39
…
REMARK       I    between atoms: C6_45   and  O6_54
REMARK       I    between atoms: C7_46   and  N2_48
ROOT
HETATM    1  C1  SIA A 605     -17.112  43.270  10.241   1.00    80.71     0.239 C
HETATM    2  C2  SIA A 605     -17.099  43.719   8.806   1.00    68.73     0.258 C
HETATM    3  C3  SIA A 605     -18.492  44.346   8.975   1.00    56.41     0.114 C
…
HETATM   55  H6  NAG A 607     -17.919  42.726   0.218  1.00  0.00      0.209 HD
HETATM   56  O7  NAG A 607     -18.715  48.632   5.901  1.00 77.61     -0.274 OA
ENDROOT
TORSDOF 21
```

Figure 17. *An example of a part of the PDBQT file for the ligand from the E chain, which is set to be treated as rigid. The first line declares the degrees of freedom (0 here) and for the "REMARK" rows, the second column specifies whether this bond is set to be rotatable or not. The "ROOT" record precedes the rigid part of the molecule. The "ENDROOT" record is after the last atom in the rigid "root" record. Since for now the ligand is treated as rigid, a single"ROOT" is before all atoms and a single "ENDROOT" after all the atoms. Finally the "TORSDOF" record is the number of torsional degrees of freedom in the ligand and is independent from what the user has set as rotatable or not.*

2.2.3 Setting the search space

The search space determines where the movable atoms should lie. This is the only space the algorithm will explore and it should be as small as possible, but not smaller. The creators of vina suggest a way of calculating the search box(43) which I also tried, but eventually chose to use the eBoxSize(44), a Perl script developed to return the optimal edge length of a cubic docking box, based on the ligand to be docked. The scientists who developed this script found that higher accuracy in docking is achieved when the search space is 2.9 times larger than the radius of gyration of the ligand.

The box suggested by eBoxSize for the ligands bound in chains A, C and E are summed in Table 6, along with the coordinates of the center of the search box. It should be noted that according to the instructions from the vina creators, the search space size was always increased to final 22.5Å, if it resulted to be less than that when one follows their instructions. An example of what this search box looks like when visualized in AutodockTools(45, 46) is shown in Figure 18.

Ligand in chain:	Search space dimensions (in Å)	x	y	Z
A	19.193	-16.338	44.862	6.688
C	23.316	23.524	54.753	-11.388
E	23.019	4.341	82.384	19.377

Table 6. *Search space size for the ligands of the three chains of the trimeric Hemagglutinin (PDB ID: 4JTV) as suggested by eBoxSize.*

2.2.4 Complications of the choice of system with the original goal

As mentioned in Section 1.5, the original goal of this BSc thesis was to test the effects of the mutations in the Koel et al. paper on the binding efficiency of the H1 Hemagglutinin molecule to α2,6 – SAs. However, while setting the system, I realized that almost all substitutions were outside the search space.

Not having enough time to choose and prepare a different system, I decided to test only the 224EA mutation, which is inside the search space and mutations on Tyr91 (H1 numbering), which is an important amino acid in the Receptor Binding Site(32). However, the substitutions in Koel et. al paper could be tested under a similar system in the future.

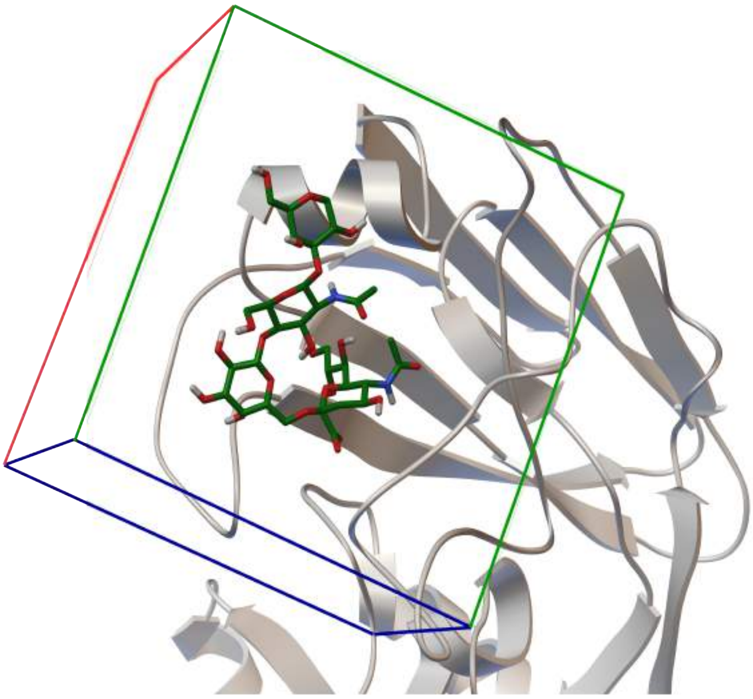
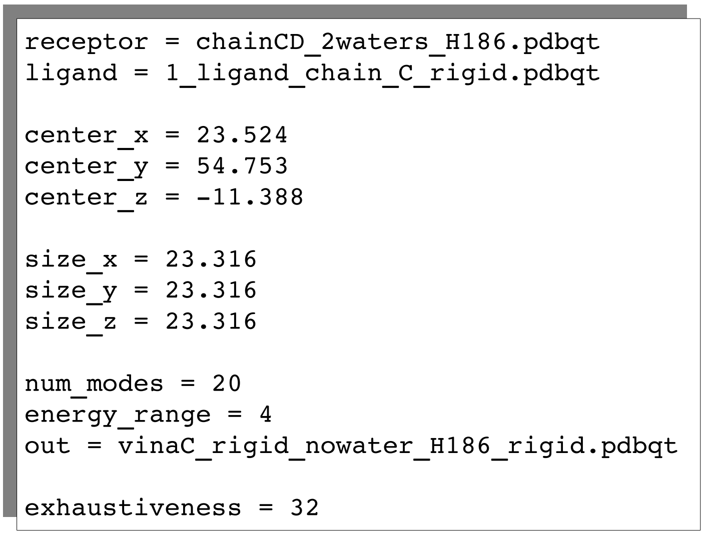


Figure 18. *The borders of the search space for docking with chain E from hemagglutinin in 4JTV PDB file as receptor. The bound ligand is also shown. The dimensions and coordinates of the box are the ones mentioned in Table 6.*

2.2.5 Configuration file

It is more convenient, especially when performing multiple docking simulations, to create configuration files that include the commands needed to run a vina simulation. An example of such a file is shown in Figure 19. Notice that the files needed as input have to either be in the current directory or their absolute path should be included.



```
receptor = chainCD_2waters_H186.pdbqt
ligand = 1_ligand_chain_C_rigid.pdbqt

center_x = 23.524
center_y = 54.753
center_z = -11.388

size_x = 23.316
size_y = 23.316
size_z = 23.316

num_modes = 20
energy_range = 4
out = vinaC_rigid_nowater_H186_rigid.pdbqt

exhaustiveness = 32
```

Figure 19. *An example of a configuration file for a vina simulation. The first two lines specify the receptor and ligand PDBQT files to be used and the next six the characteristics of the search space. The "out" command is to specify the path and filename of the output. The num_modes refer to the number of models to be returned (default : 9, max:20) and the energy_range to the energy difference between the best binding mode and the worst binding mode to be displayed. Finally exhaustiveness is proportional to time and performs additional docking simulations; the default value is 8 but depending on the system it is suggested to be increased to 24 or 32*(47).

3. The simulations

3.1 Running the redocking simulations with rigid molecules

In order to validate whether docking can indeed predict the global energy minimum and the best ligand pose for this specific hemagglutinin molecule, redocking experiments were run. **Simply, the ligand was separated from each chain and the original ligand pose was attempted to be obtained after docking with vina. For this first group of redocking experiments, both the receptor and the ligand were considered to be rigid.**

For every chain, the redocking simulation was run and repeated independently 100 times, in order to perform statistical analysis to draw conclusions, as Autodock Vina employs stochastic global optimization approaches. Each docking simulation results in 10 models/poses, ranked from the one with the highest binding affinity, to the one with the lowest, or from the more probable to the less probable. An example of a file resulting from a vina simulation is shown in Figure 20.

Both the monomers and the trimer were used as the receptor molecule but no statistically significant difference was found; thus the information about the simulations using the monomers only was included. This didn't come as a great surprise, since docking is focused on the RBS only (which by the way doesn't participate extensively in intermolecular interactions for the trimer formation(28)).

```
MODEL 2
REMARK VINA RESULT:     -10.5      2.468      7.070
REMARK  0 active torsions:
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK       I    between atoms: C1_1  and  C2_2
REMARK       I    between atoms: C2_2  and  O6_39
…
REMARK       I    between atoms: C7_46  and  N2_48
ROOT
HETATM    1  C1  SIA A 605    -13.769  45.869   9.481  1.00  80.71     0.239 C
HETATM    2  C2  SIA A 605    -14.893  46.009   8.492  1.00  68.73     0.258 C
….
HETATM   55  H6  NAG A 607    -21.385  50.124   4.452  1.00  0.00       0.209 HD
HETATM   56  O7  NAG A 607    -15.875  44.199   2.929  1.00  77.61    -0.274 OA
ENDROOT
TORSDOF 21
ENDMDL
```

Figure 20. *Part of the file resulting from a vina simulation. The "REMARK VINA RESULT" record includes the predicted binding affinity of the specific pose (in kcal/mol); the next two entries on this record inform about the RMSD from the best mode: the first one is "rmsd lower bound", which takes into account the symmetry in the molecule, while the second one ("rmsd upper bound") calculates the distance between the different poses between the exact same atoms, namely the ones with the same label. Each model starts with the "Model X" record, where X is the number/ranking of the model and ends with the "ENDMDL" record. The rest of the information have already been explained in Figure 17.*

Since for every chain 100 repeats of independent docking simulation runs were performed, each returning 10 models, 1000 poses for the ligand of each chain were collected. For these models, RMSD matrices were calculated in order to cluster the results and to compare them to the pose of the ligand in the crystallographic experiment, which was also included in the RMSD matrices calclulation.

*There is a trap here (in which I of course fell in at first) and it's that one must not assume that the 3rd for example model in the first run is structurally the same as in the 3rd model in the next run, although they may have been ranked the same and they may have the same binding affinity. Not only is it possible for one pose to be present in one run and not in the next one, but also it is possible that the models ranked the same don't have the same binding affinity in each run – and even if they do, they could be just different poses with the same binding affinity. This is why, **when it comes to statistical analysis regarding the poses, the structural related ones should be taken into account**, namely the ones clustered together, as described in the beginning of the paragraph.*

3.1.1 RMSD matrices

In order to cluster the different models to find the structurally similar but also see how relevant they are to the crystallographic pose of the ligand, RMSD matrices were calculated: one for every chain. This matrix includes the RMSD between every given pose and among them and the crystal pose of the ligand.

The matrices were calculated using crossDCD (Appendix A2), a Perl script modified to calculated RMSD without performing least squares fitting. This script accepts two dcd files and a psf file as input. The dcd files were produced from the pdbqt files that include all the models produced from vina, using VMD, after editing using bash shell scripting; the psf files used with crossDCD are pseudo – psf and they are not suitable for any other program that works with psf files. They were produced using another perl script, pdb2psf.

It is important to note that the dcd and the psf files need to have the same number of atoms, and since the models produced from vina include the hydrogen atoms (see Section 2.2.2 Preparing the ligand), the psf file should be produced using a ligand with the polar hydrogens added and not the ligand originally found in the PDB file, in which no hydrogens were included.

3.1.2 Clustering based on RMSD matrices

The RMSD matrices were used to cluster the models in order to determine the structurally related ones but also to see how relevant they are to the crystallographic pose of the ligand. The clustering and visualization of the results were performed using the R programming language(48, 49, 50, 51, 52, 53).

The cluster analysis performed was hierarchical and the agglomeration method used is named “average” and refers to the UPGMA agglomeration method or Unweighted Pair Group Method with Arithmetic Mean. An example of the visualization of the results for chain A redocking experiments, in the form of dendrogram, is shown in Figure 21; since for this amount of models this size of image renders the dendrograms completely useless, a larger version of them can be accessed here. Together were clustered the poses that differ by 1.1 Ångström or less.

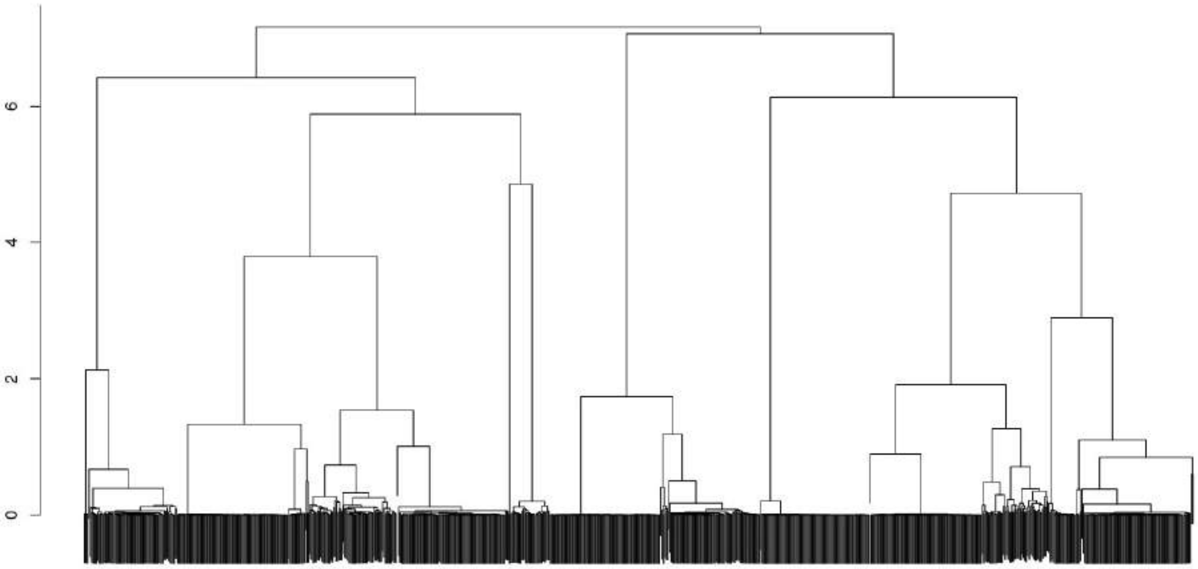


Figure 21. *Dendrogram of the 1000 models derived from the 100 repeats of the redocking vina simulations, where the ligand from chain A, treating it as rigid, was redocked to chain A. The crystal pose was also included in the clustering process. The dendrograms of better quality can be accessed here.*

The different clusters of vina models and their relative distance to the crystal pose were also visualized using metric Multi-Dimensional Scaling or MDS in 2D and 3D that can be explored by clicking here. More specifically, the cmdscale function in R was used(48) as well as the plotly package(52) for the visualization. It is important to explore and compare these diagrams: for example while some clusters seem mixed in the two-dimensional diagram, they look separate in the 3D one, indicating that there is important information in the third dimension, thus making it difficult to visualize the clusters in only two dimensions.

The most important part of these diagrams though is that the crystal pose is close to a very compact cluster comprising either the first models

of each run for chain A and C or the second models of each run for chain E. This is also visible by exploring the dendrograms: the models clustered together with the crystal pose have an RMSD close to zero.

Since for chain A and C, the first model from each of the 100 runs is the same (RMSD among them is zero; see also the MDS analysis discussed above as well as the dendrograms) and they differ from the crystal pose of the ligand by less than 1.1 Ångström, only one of them is compared to crystal pose below, instead of all of them or their mean. For chain E the same apply, but for the second model of each of the 100 runs (Figure 22).

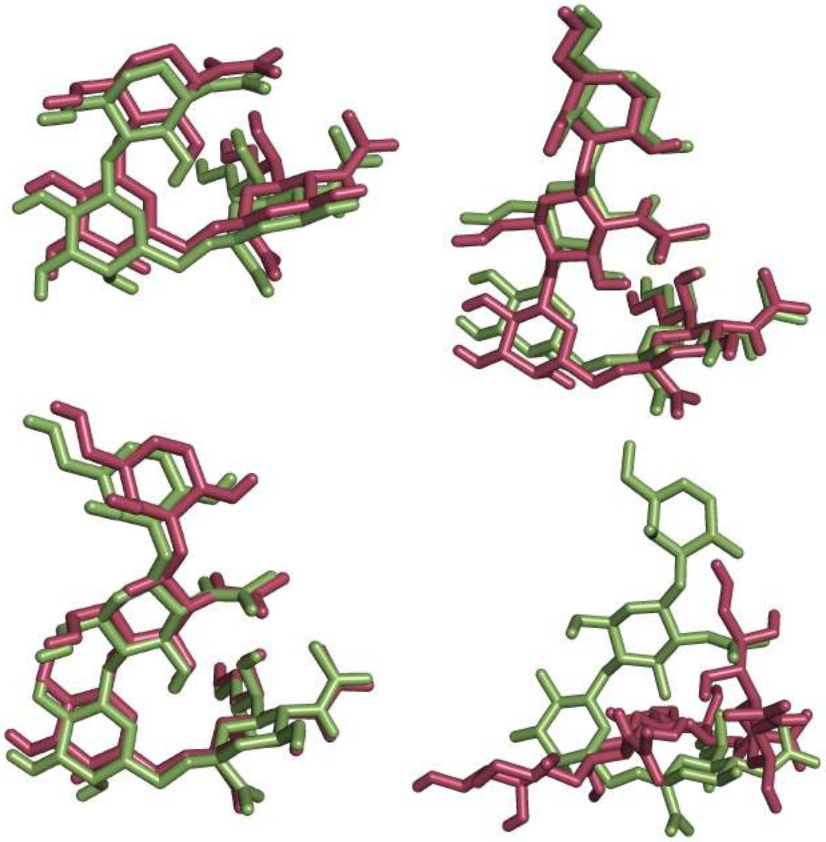


Figure 22. *The ligands from chains A, C and E (top left, top right and bottom respectively) are shown in green; a representative from the cluster of the models that have less than 1.1A RMSD with the crystal pose is shown in pink. For chains A and C this is the first model of a run, but for chain E, it is the* ***second*** *model. The first model for chain E is on the bottom right; notice the difference between this pose and the original ligand from chain E.*

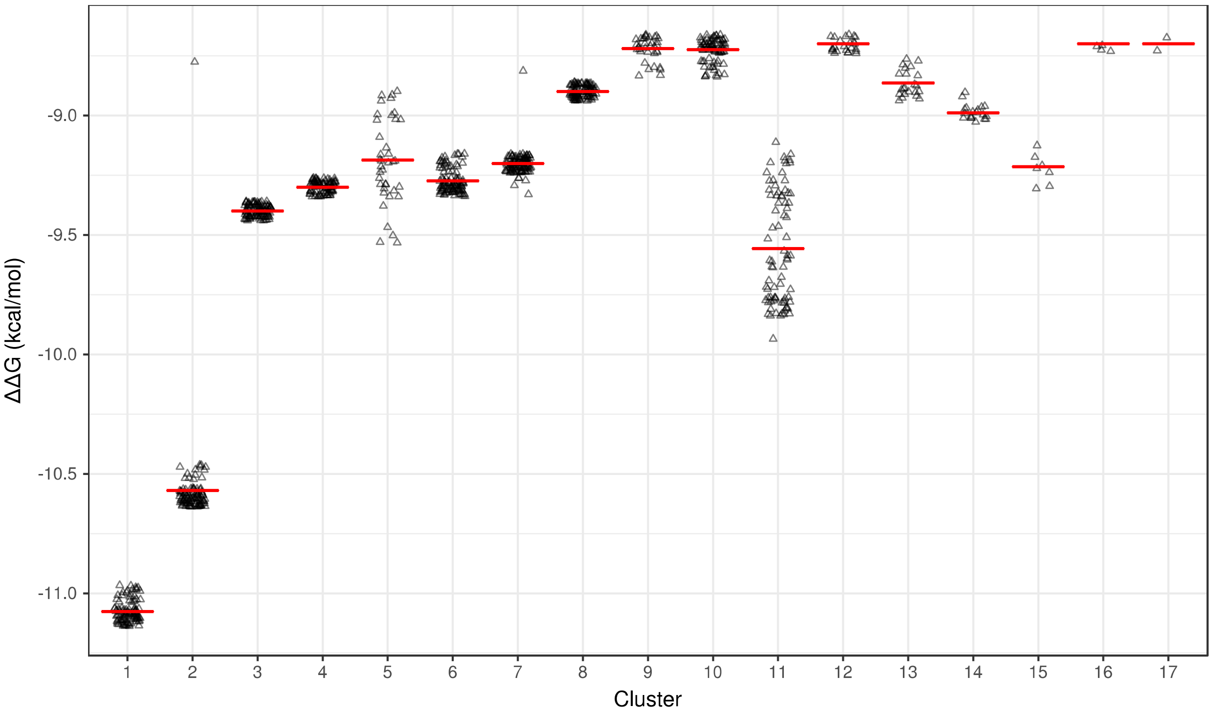
These results suggest that vina manages to find the correct binding pose in these redocking experiments, where both HA and the ligand are considered rigid. Attention should be paid regarding chain E, as vina ranks second the pose that is known to correspond to the bound state of the ligand, namely the models that are identical to the crystal pose of the ligand.

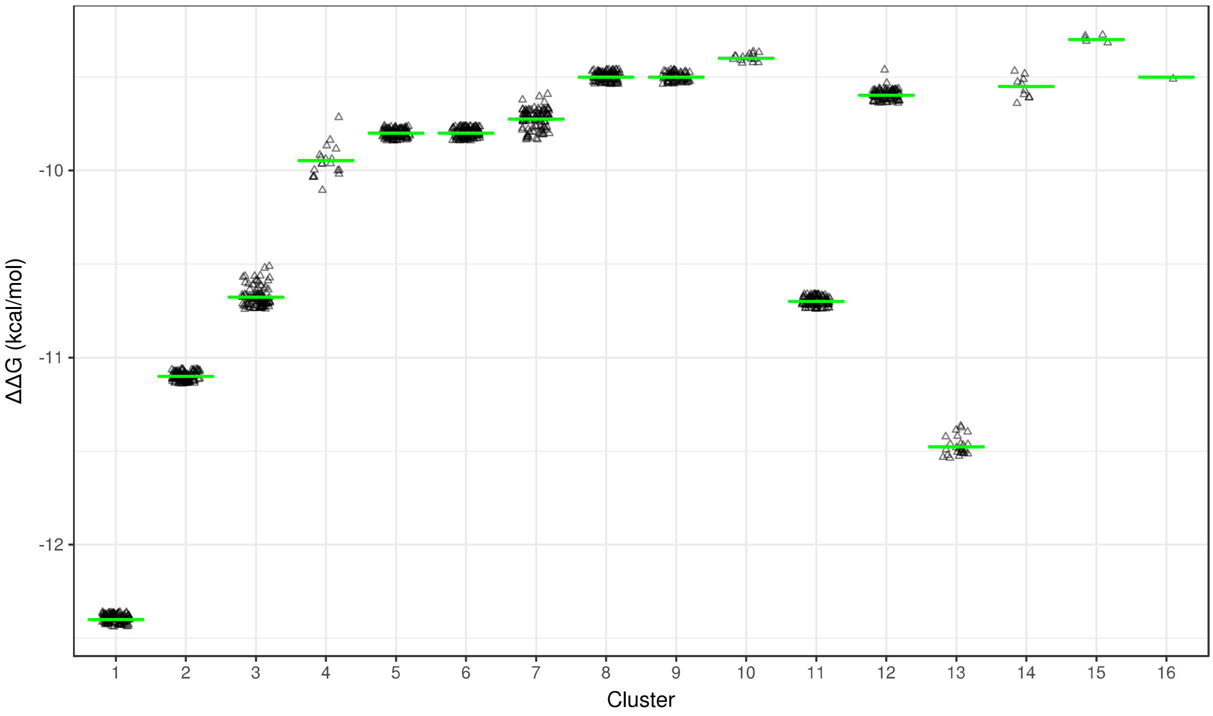
Biases in the above redocking simulation

In real life molecules are not rigid as both the ligand and the protein have rotatable bonds; what's more, using the crystal structures of the bound state to simulate the procedure of transition from the free state to the bound state adds bias, as these already possess the angles and bond lengths they adopt in the bound state. In addition, molecules are actually diluted in solutions, surrounded by other molecules, their binding is aided by additional molecules etc. Keeping all these and many more restrictions in mind, one shouldn't assume that Vina can predict the correct global minimum of energy. This is more obvious in next experiments, where some of the ligand bonds are considered rotatable, namely where degrees of freedom are allowed.

3.1.3 ΔΔGs of the different models

Vina returns an estimated free energy of binding for every pose of the ligand it suggests; these models are actually ranked from most probable to less probable according to this ΔΔG value. For the clusters calculated and discussed in the two previous sections, plots were created showing the different ΔΔG values of each cluster's model and their mean value. (Figure 23). Something worth noticing is that clusters one to two or one to three are represented by a mean value that is remarkably lower than the mean of the rest of the clusters. What's more the clusters with the lower means represent models which ΔΔGs are less dispersed.





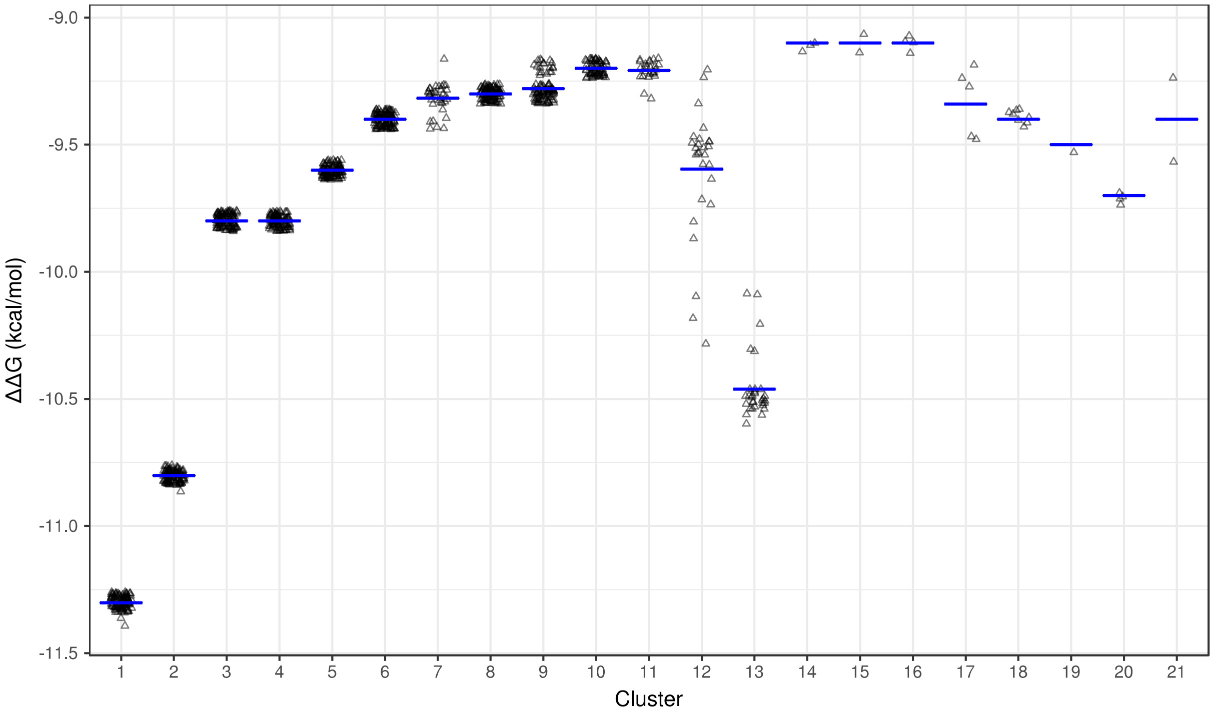


Figure 23. *Scatter plots presenting the ΔΔG value (in kcal/mol) of every model of each cluster, for the redocking simulations of chains A, C and E with their rigid ligand from top to bottom. Mean value is shown as a line. Notice that the free energy of the bound state has negative value.*

3.1.4 Comparison between experimental and estimated K_d

Vina also returns an estimated ΔΔG for each model (namely for each predicted pose of the ligand in its bound state) in kcal/mol. One can convert this to K_d (namely dissociation constant) and compare the values obtained from vina from the redocking simulations to the experimentally determined values provided from the authors who solved the structure used (namely 4JTV).

The affinity and kinetics of the binding of soluble HA to LSTc were analyzed at 25°C or 298K(27). Taking into account that the gas constant is R = 1,987cal/mol and the equation: $K_d = e^{-\Delta G/RT}$ the ΔΔGs can be calculated from K_ds and vice versa. The K_d for the HA – LCTc complex used for the simulations was calculated to be 3.74μM for the complex *in vitro*(27) which corresponds to free energy equal to -7.4 kcal/mol. In the previous section, the vina predicted ligand poses that are structurally more similar to the experimentally determined ligand pose were determined and for them, vina also returned an estimated ΔΔG value. For these, the average and standard deviation were calculated and are shown in Table 7. For the average value, the conversion to K_d was performed: for these simulations an underestimation of the dissociation constant is observed.

Chain	**Mean ΔΔG (kcal/mol)**	**SD**	**K_d(μM)**
A	-11.1	0.04	0.0072
C	-12.4	0.00	0.0008
E	-10.8	0.01	0.0119
Experiment	-7.4	–	3.74

Table 7. *The mean and SD of the ΔΔGs of the ligand poses that belong to the cluster that differs from the experimentally determined ligand conformation by less than 1.1*Ångström *, for chains A, C and E respectively. (Remember that the first model of each of the 100 runs was the most similar to the respective ligand pose of the crystal structure, for chains A and C; for chain E the same apply but for the second model) The experimentally determined K_d(27) the PDB structure used, is also shown, as well as its conversion to ΔΔG.*

3.2 Redocking simulations using flexible ligand

3.2.1 Determining the rotatable bonds

Autodock Vina is successful with systems with approximately 20 torsions and allows a maximum of 32(54). When the ligands from chain A, C and E were redocked to the chains A, C and E of the HA molecule respectively, treating all bonds as rotatable, no model similar to the crystallographic pose was returned (data not included). Thus, specific bonds were treated eventually as rotatable, based on the bond angles of the same ligand (LSTc) in six different PDB structures (Table 8). The PDB structures selected contained more than one ring of the LSTc molecule. The different ligand molecules bound to different chains of the same homotrimer were taken into account as separate molecules and not as the mean of the three LSTc ligands of the specific HA PDB structure.

In order to calculate the intramolecular bond angles, PLATON(57) was used. PLATON automatically generates a variety of geometrical entities such as bond distances, bond angles, torsion angles, least-squares planes and ring-puckering parameters of a structure. For every ligand (16 in total) of the structures mentioned in Table 8, PLATON was run and the results regarding bond angles that differ more than 10 degrees among the ligands are shown in Table 9. Information regarding the 5th ring of LSTc is not presented here, as the ligands in 4JTV PDB file are consisted of 3 (chain A) or 4 (chain C and E) rings. The bonds that were eventually considered rotatable participate in the formation of angles that differ more than 10 degrees among the different ligand and are plotted on LSTc in Figure 24. Note that the bonds in aromatic rings and amide bonds were excluded. The cutoff of 10 degrees is arbitrary.

PDB ID	TITLE	Resolution (A)
1RVT	1930 H1 Hemagglutinin in complex with LSTc	2.5
1RVZ	1934 H1 Hemagglutinin in complex with LSTc	2.25
3UBE	Influenza hemagglutinin from the 2009 pandemic in complex with ligand LSTc	2.15
4JTV	Crystal structure of 2009 pandemic influenza virus hemagglutinin complexed with human receptor analogue LSTc	3.0
4JU0	Crystal structure of 2009 pandemic influenza virus hemagglutinin mutant D225E complexed with human receptor analogue LSTc	2.91
4JUJ	Crystal structure of 1918 pandemic influenza virus hemagglutinin mutant D225G complexed with human receptor analogue LSTc	3.01

Table 8. *PDB IDs of the structure used to infer possible rotatable bonds. The resolution each structure was solved at is also shown.*

Atom 1 (PDB nomenclature)	Atom 2 (PDB nomenclature)	Atom 3 (PDB nomenclature)	Angle (PLATON nomenclature)	Mean	SD	max – min	counts
GAL – O6	SIA – C2	SIA – C3	O(13) – C(2) – C(3)	98.7	14	33	14
GAL – C1	NAG – O4	NAG – C4	C(12) – O(15) – C(21)	118.1	12.9	31.7	13
SIA – C1	SIA – C2	SIA – C3	C(1) – C(2) – C(3)	100.6	11.9	28.3	16
GAL – O5	GAL – C1	NAG – O4	O(12) – C(12) – O(15)	106.3	10.8	26.4	13
SIA – O6	SIA – C2	SIA – C3	O(4) – C(2) – C(3)	99.8	8.7	23.6	16
SIA – C2	SIA – o6	SIA – C6	C(2) – O(4) – C(6)	123.1	10.2	22.5	16
SIA – O6	SIA – C2	GAL – O6	O(4) – C(2) – O(13)	114.4	8	18.5	14
SIA – O6	SIA – C2	SIA – C1	O(4) – C(2) – C(1)	115.8	7.6	18.1	16
SIA – O6	SIA – C6	SIA – C5	O(4) – C(6) – C(5)	105.3	6.2	17.5	16
NAG – O4	GAL – C1	GAL – C2	O(15) – C(12) – C(13)	112.4	6.3	17.3	13
NAG – C1	GAL – O3	GAL – C3	C(18) – O(20) – C(28)	111.4	5.8	15.4	6
SIA – C2	GAL – O6	GAL – C6	C(2) – O(13) – C(17)	111.4	5	14.3	14
NAG – O5	NAG – C1	GAL – O3	O(16) – C(18) – O(20)	108.7	4.9	13.1	6
SIA – O6	SIA – C6	SIA – C7	O(4) – C(6) – C(7)	113.3	5.3	12.9	16
SIA – C5	SIA – N5	SIA – C10	C(5) – N(1) – C(10)	124.7	3.7	12	16

Table 9. *Analysis of bond angles of LSTc ligands in different PDB structures that differ more than 10 degrees among the ligands. Average, SD, difference between maximum and minimum angle size and number of ligands that include each specific angle.*

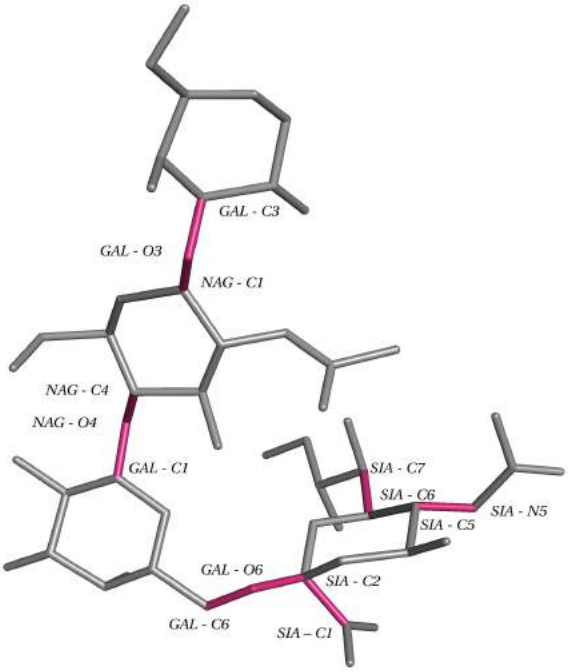


Figure 24. *Bonds that are treated as rotatable in vina simulations. Refer to Table 9 for more details.*

Note that the ligand of chain A has three rings (SIA – GAL – NAG), thus 7 degrees of freedom were allowed for it. Ligands of chain C and E have four rings (as shown in this Figure) and this 9 degrees of freedom were allowed.

3.2.2 Preparation and analysis of redocking simulations using flexible ligands

For this round of experiments, the protein was treated as rigid and the ligand as flexible, with the degrees of freedom described above. Some steps regarding the preparation of a flexible ligand are the same as the ones described in Section 2.2.2., namely adding the hydrogens, merging the non-polar ones, assigning charges and appropriate atom types and randomizing the coordinates, to remove any bias resulting from the ligand's initial position in the RBS. After these steps, the torsion tree was determined, leaving the aforementioned bonds rotatable. An example of a file of a flexible ligand is shown in Figure 25.

After finally redocking the ligands of chains A, C and E treating them as flexible, the same analyses were performed, same as for the simulations with rigid ligands (Section 3.1). For these simulations, higher cutoffs were used to create clusters, as the most similar to the crystal structure cluster of poses differed by 1.3 Ångström or more. More specifically for chain A the cutoff used was 1.3Å, for chain C 2.1Å , while no clusters were created for chain E results, as the most similar docking pose obtained by vina is the 9th model of the 3rd run and the RMSD between them is 5.83Å. Namely for chain E, vina didn't manage to find the correct ligand pose, when its treated as flexible with nine degrees of freedom.

The dendrograms presenting the hierarchical clustering results can be accessed here and the Multi-Dimensional Scaling interactive plots here.

It is worth noticing that for chain A (with the 3 – ringed ligand and 7 degrees of freedom) vina systematically found a docked pose similar to the crystal pose (Figure 26). The same happens for chain C (Figure 27), although the models that were clustered together with the crystal pose are ranked lower (mostly 6th and 7th models, ranked according to ΔΔGs). However, for chain E a docking pose similar to the crystal pose wasn't found by vina.

```
REMARK  9 active torsions:
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK    1  A    between atoms: C1_1  and  C2_2
REMARK    2  A    between atoms: C2_2  and  O6_39
…
REMARK       I    between atoms: C6_62  and  O6_69
ROOT
HETATM    1  C1  GAL E 604   -2.246  -0.671   2.406  1.00 105.68     0.292 C
…
HETATM   12  H4  GAL E 604   -2.194  -3.475   5.527  1.00   0.00     0.210 HD
HETATM   13  O5  GAL E 604   -1.012  -1.173   2.862  1.00  85.80    -0.348 OA
ENDROOT
BRANCH   1  14
HETATM   14  O4  NAG E 605   -2.408  -0.738   0.978  1.00 125.17    -0.348 OA
BRANCH  14  15
HETATM   15  C4  NAG E 605   -1.954   0.469   0.414  1.00 123.19     0.186 C
HETATM   16  C5  NAG E 605   -3.074   1.122  -0.380  1.00 125.33     0.180 C
…
ENDBRANCH   1  14
…
TORSDOF 27
```

Figure 25. *Part of a PDBQT file of a ligand to be treated as flexible in vina simulations. See Figure 17 for more information regarding the different fields of the file.*

These results indicate that using flexible LSTc ligand can in fact return poses similar to the crystal structure; however this happens with higher predictability when using chain A of the 4JTV PDB file and the ligand with three rings, allowing it seven degrees of freedom.

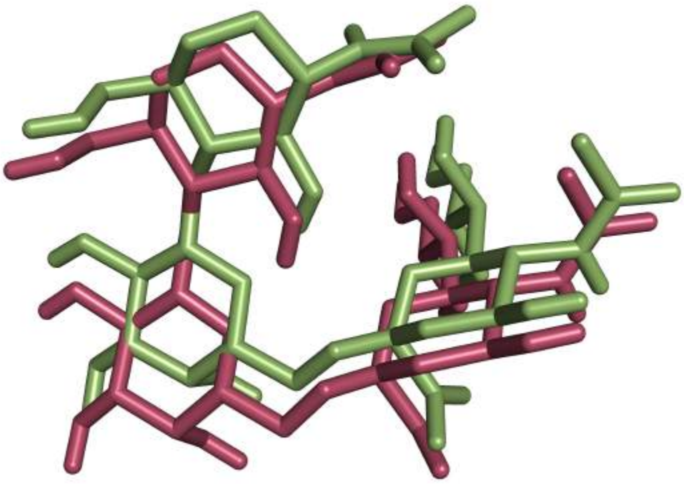


Figure 26. *The crystal pose of the ligand of chain A is shown in green and a model of the ones clustered with it (cluster 2, when using a cutoff of 1.3A) is shown in pink. Remember that this simulation was performed using a flexible ligand, allowing it seven degrees of freedom, as shown in Figure 24.*

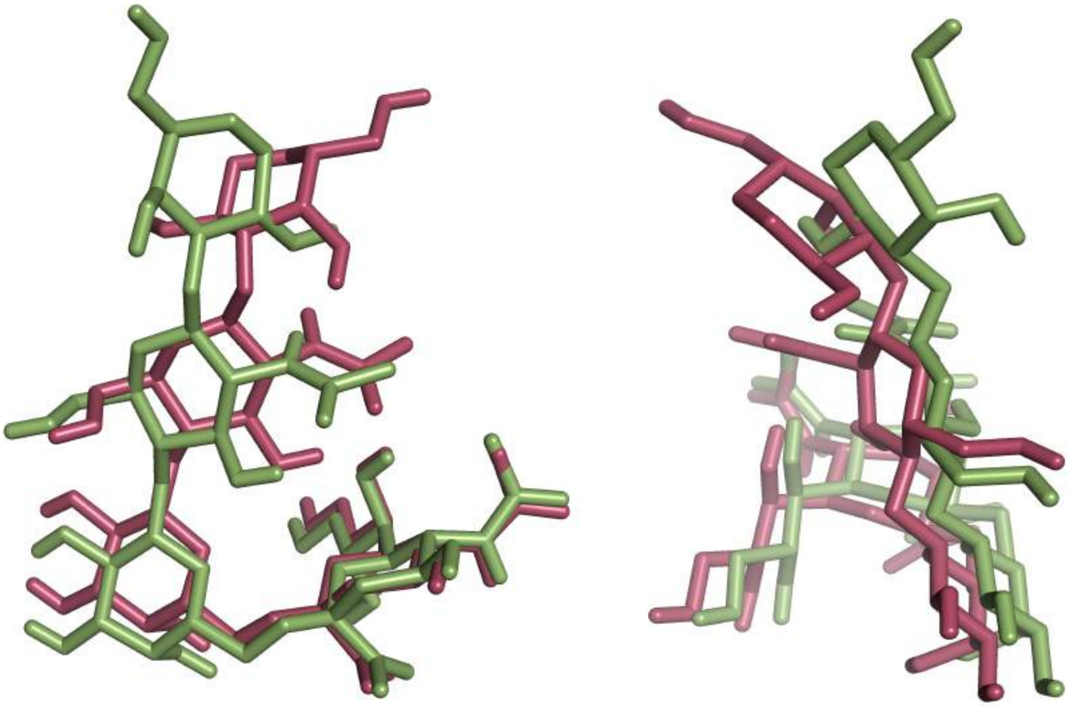


Figure 27. *The crystal pose of the ligand of chain C is shown in green and a model of the ones clustered with it (cluster 7,when using a cutoff of 2.1A) is shown in pink. Remember that this simulation was performed using a flexible ligand, allowing it nine degrees of freedom, as shown in Figure 24. Notice, in the side view, how the second GAL ring (upper ring) obtains a different conformation from the one in the crystal, while the rest of the ligand is more similar to the crystal pose.*

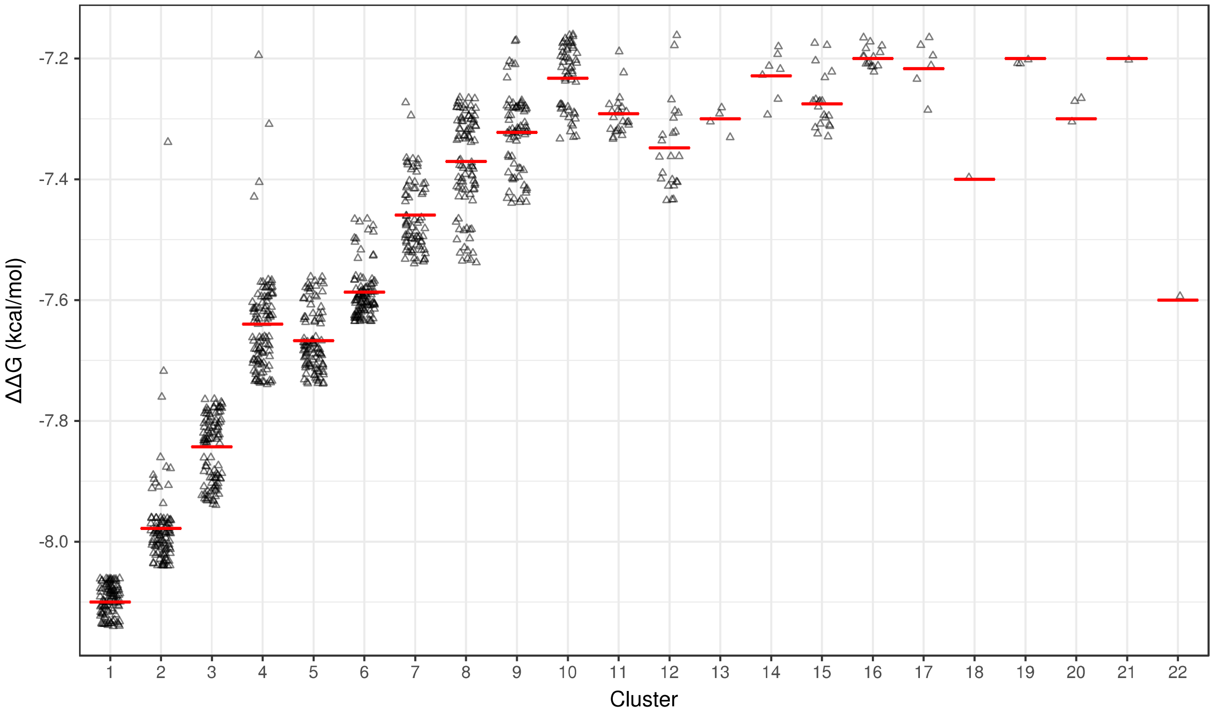
The ΔΔGs of the clusters for chains A and C for the redocking simulations using flexible ligands were also plotted and the mean of each cluster was calculated (Figure 28). As mentioned in Section 3.1.4, the experimentally determined ΔΔG of the bound state of the ligand for 4JTV is -7.4 kcal/mol. The ΔΔGs of the models that were clustered together with the crystal ligand pose are closer to the experimentally determined ΔΔG in this simulation, compared to the ones obtained when using rigid ligands (Section 3.1.4), where the K_d was overestimated in comparison with the experiment.

3.3 Docking simulations after *in silico* mutagenesis of the hemagglutinin molecule

3.3.1 *In silico* mutagenesis of 224E (H1 numbering) to Alanine

The 224EA mutation in A/Netherlands/602/09 HA didn't lead to any significant difference regarding the binding efficiency of the protein to receptors, according to Koel et. al. Aiming to test this mutation *in silico*, in the docking system described above, the substitution was introduced in the PDB structure 4JTV.

In order to replace the Glutamic acid at position 224 (H1 Numbering, corresponding to amino acid 230 in 4JTV PDB file) in chains A, C and E, PyMOL Viewer was used and more specifically the Mutagenesis Wizard. Since Alanine has no rotameres, the conformation produced by the tool was used.



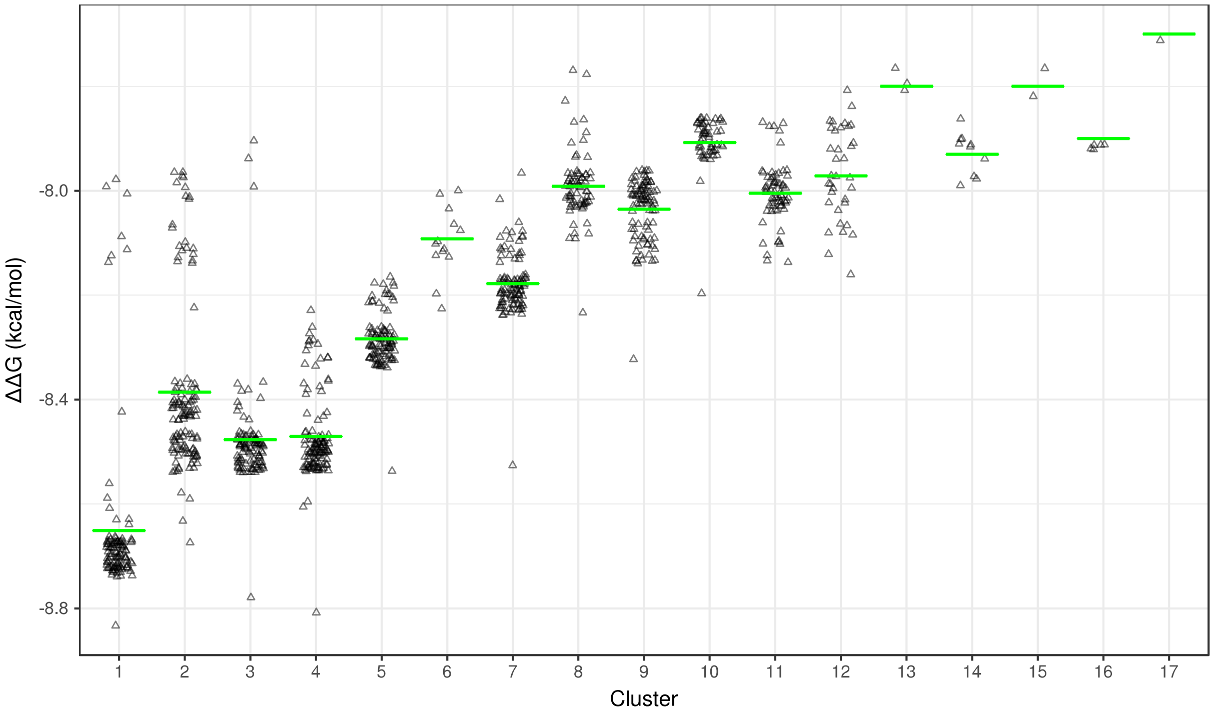


Figure 28. *Scatter plots presenting the ΔΔG value (in kcal/mol) of every model of each cluster, for the redocking simulations of chains A, C and E with their flexible ligand from top to bottom. Notice that the higher dispersion of the values, in comparison with the redocking simulations using rigid ligands, is due to the clustering cutoff chosen; it is higher, thus more diverse models are included in a single cluster. This of course affects the mean. For chain A, the ligand crystal ligand pose belongs to the second cluster, and for chain C to the 7th cluster. Mean value is shown as a line. Notice that the free energy of the bound state has negative value. Also notice that the mean value is not the best representative here because of the many outliers, due to clustering using higher cutoff RMSD.*

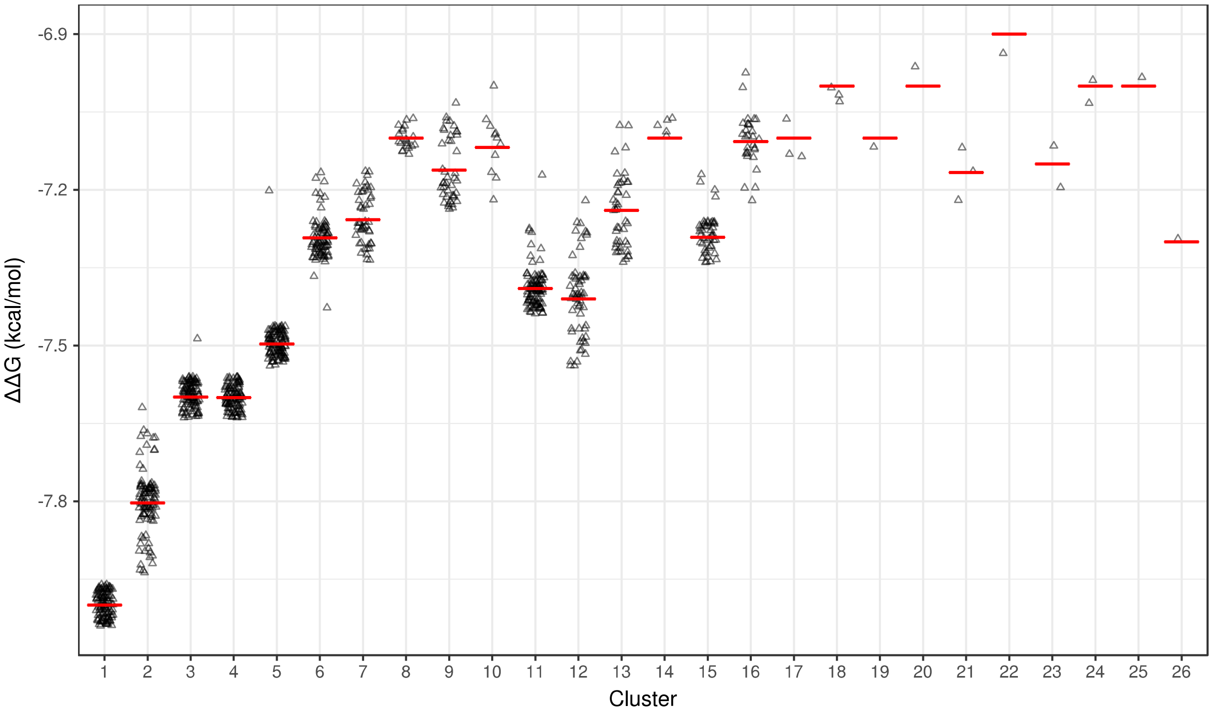
After the introduction of the 224EA mutation, the preparation of the receptor was performed in the same way as described in Section 2.2.1.

The same apply for the 91YA substitution, analyzed later.

3.3.2 Docking of H1 HA bearing the 224EA mutation using rigid ligands

For this round of simulations, all three chains carried the 224EA mutation and the ligands of each chain were redocked, treating them as rigid. Then the usual analyses were performed, namely clustering, dendrograms, MDS and ΔΔGs' plotting, as described more extensively in previous Sections. The resulting dendrograms can be accessed here, while the interactive MDS plots in 2D and 3D can be accessed here. The crystal pose was included in the analyses; the same cutoff for the clustering as in the simulations without the substitution, using rigid molecules was also used, namely 1.1Ångström.

The ΔΔGs of the models per cluster for each chain were also plotted (Figure 29).



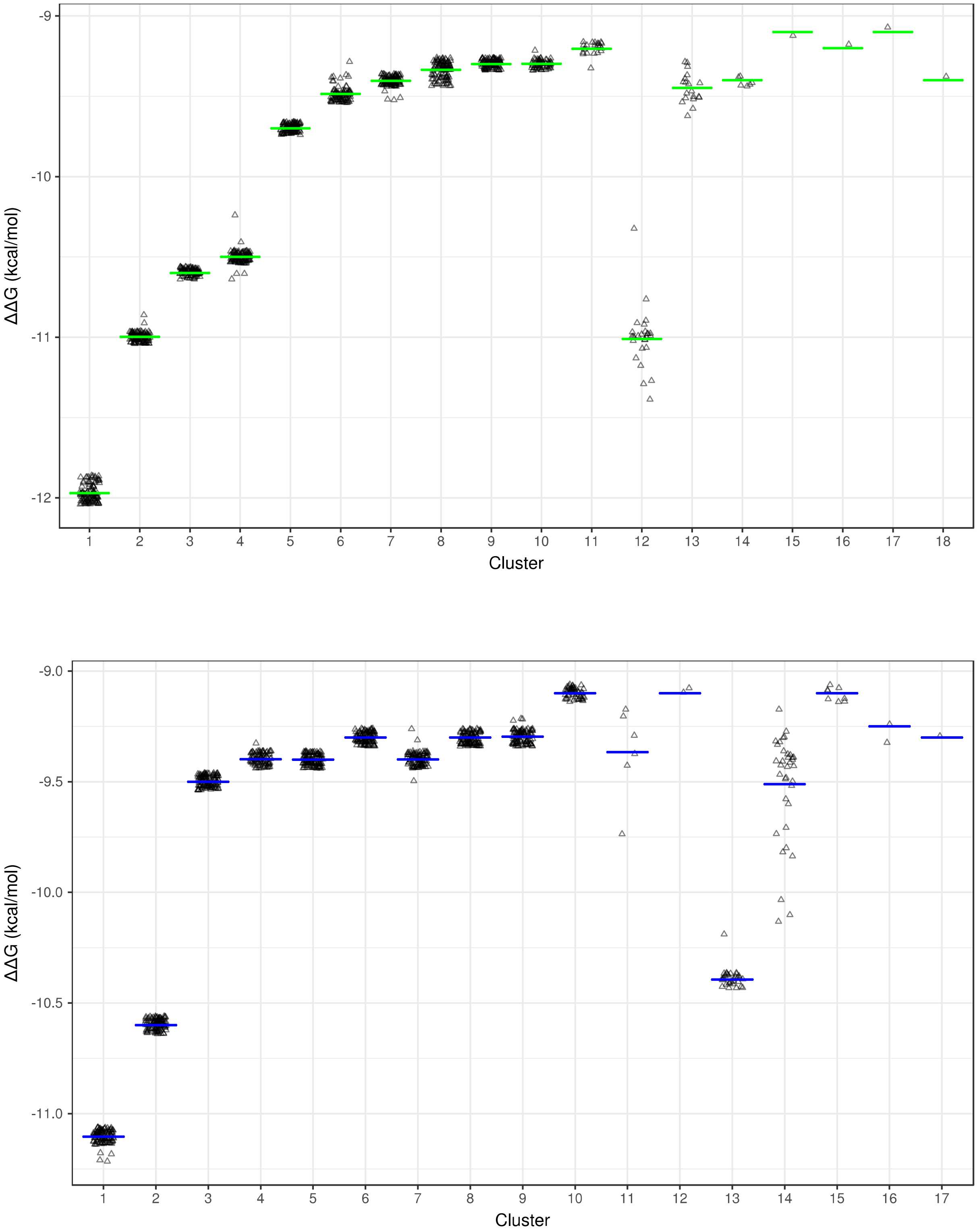


Figure 29. *Scatter plots presenting the ΔΔG value (in kcal/mol) of every model of each cluster, for the redocking simulations of chains A, C and E, after the 224EA mutation, with their rigid ligand from top to bottom. For chain A and C, the ligand crystal ligand pose belongs to the first cluster and for chain E to the second cluster. Remember that the crystal pose was clustered with the same models in the redocking experiments without mutations as well, implying that this mutation didn't affect the resulting suggested models. Mean value is shown as a line. Notice that the free energy of the bound state has negative value.*

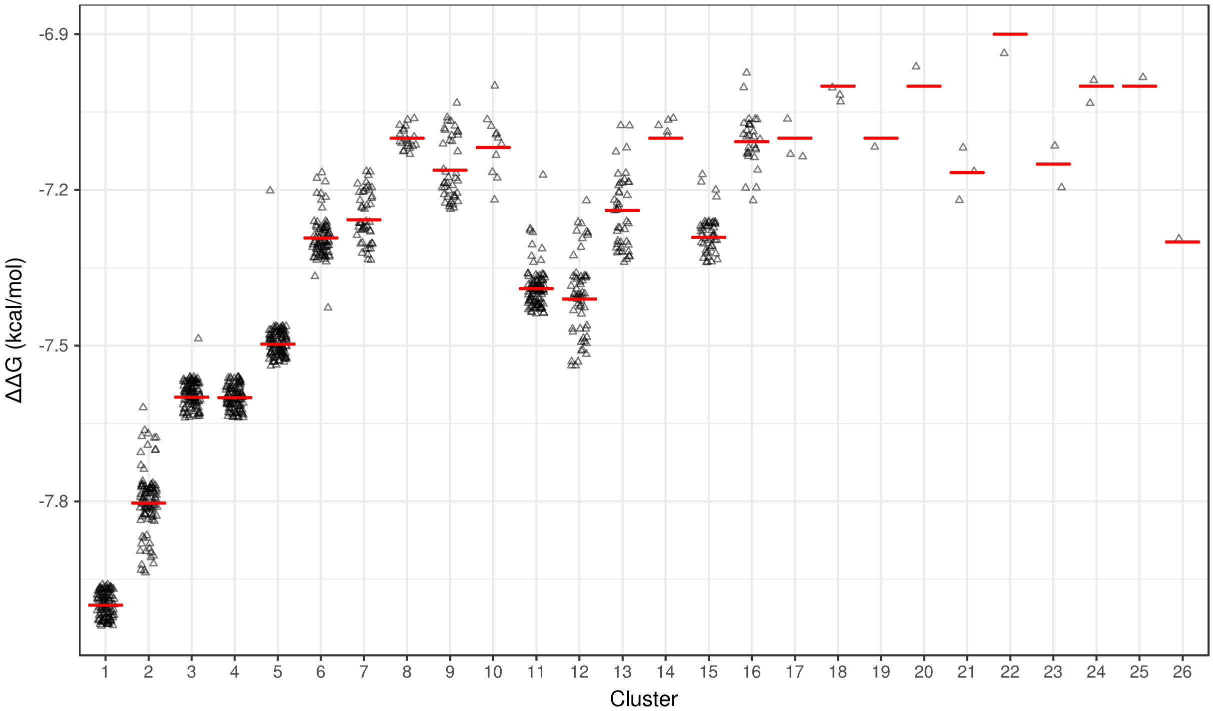
3.3.3 Docking of H1 HA bearing the 224EA mutation using flexible ligands

Same as for the original 4JTV PDB structure, the ligands were redocked to their chains after the 224EA treating them as flexible (with nine degrees of freedom, as described in Section 3.2.1.).

The dendrograms produced after clustering can be accessed here, while the MDS in 2D and 3D here. The ΔΔGs plots per cluster are in Figure 30.

Together the conformations that differ by less than 1.3Ångström for chain A and 2.1Ångström for chain C were clustered, as these where the ones used for the control simulation, namely the simple redocking simulation using flexible ligands.

Same as the redocking simulations with flexible ligands without the mutation, vina didn't manage to obtain a pose similar to the crystal pose for chain E, when allowing these degrees of freedom. Although this could be because of the mutation, given the fact that for chain A and C major changes didn't occur when the 224EA was added, this result is in accordance with the first experiment. However, I was not able to determine what is the reason why vina does manage to obtain a model similar to the crystal pose for chain A and C, but not for chain E. The most similar to the crystal pose of the ligand of chain E in this simulation had an RMSD of 5.8441Ångström.



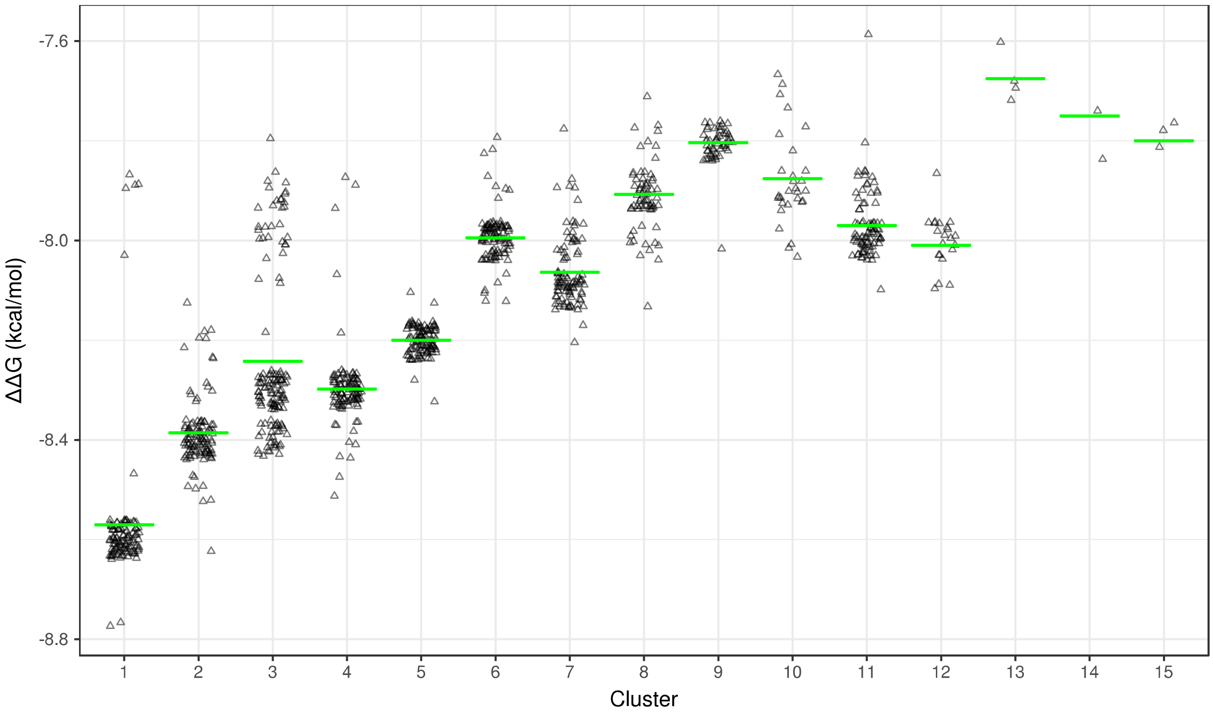


Figure 30. *ΔΔGs of the vina models of the ligands of chain A and C, when redocked to the HA molecule bearing a 224EA mutation, treating them as flexible, with the degrees of freedom mentioned in Section 3.2.1. For chain A, the crystal pose belongs to the second cluster and for chain C to the seventh cluster. Remember that the crystal pose was clustered with the same models in the redocking experiments using flexible ligands without mutations as well, implying that this mutation didn't affect the resulting suggested models. Notice that the mean value is not the best representative here because of the many outliers, due to clustering using higher cutoff RMSD.*

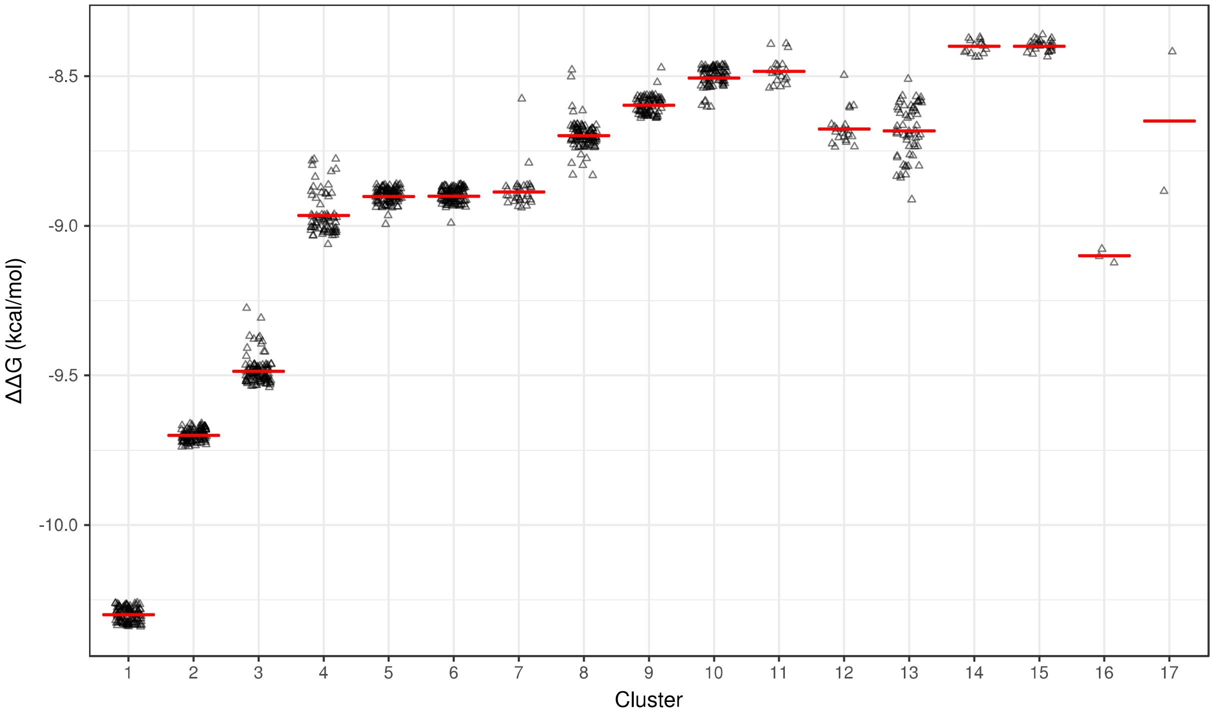
3.3.4 Docking of H1 HA bearing the 91YA mutation using rigid ligands

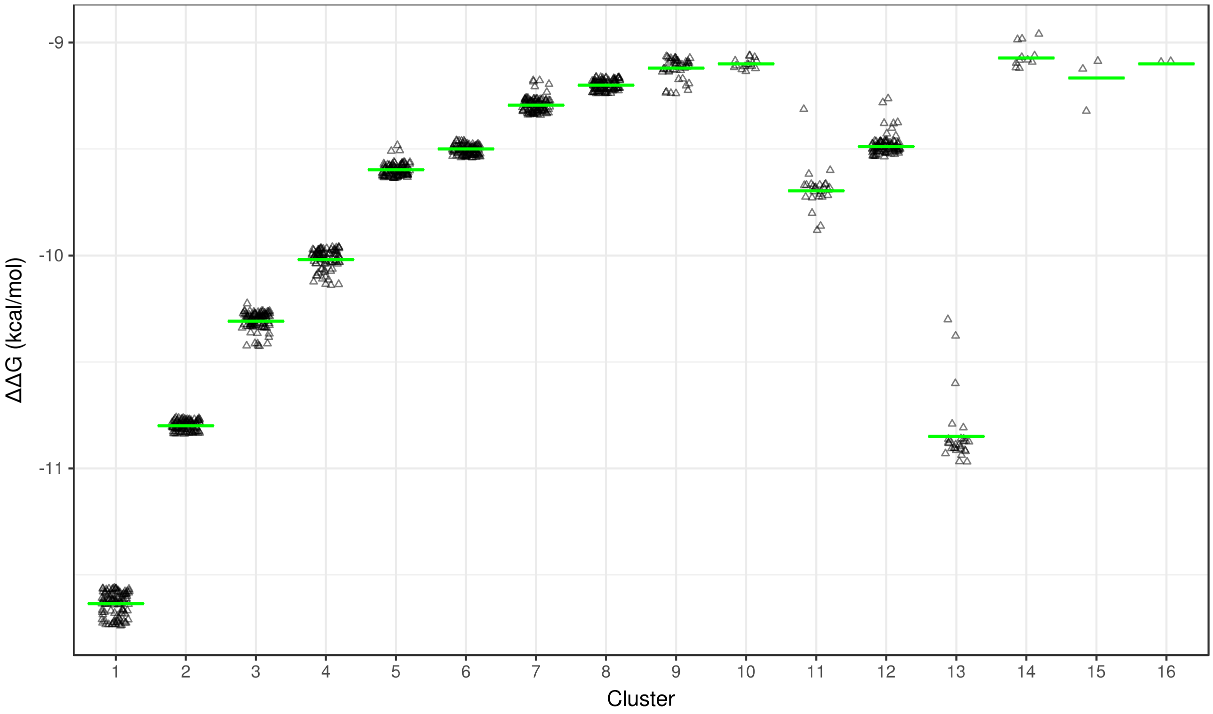
The second substitution to be introduced was not included in the Koel paper; however Tyrosine 91 is an important amino acid in the receptor binding site, highly conserved among hemagglutinin molecules. Since it was not possible to use this system to check the rest of the mutations described in Table 3, because of the complications that arose during the project (Section 2.2.4) I decided to check how this mutation affects the docking results instead.

For this reason, Tyrosine 91 was mutated to Alanine, namely from an amino acid with a polar, large side group to one with a nonpolar, small side group. Same as for the 224EA substitution, Alanine has no rotamers and so the proposed from PyMOL Mutagenesis Wizard conformation was used.

Same as for the 224EA mutation, the crystal pose was included in the analyses and see how the RMSDs between this and the now proposed vina models changed.

The dendrograms presenting the hierarchical clustering of the models can be accessed here, while the MDS analyses here. The ΔΔGs of the resulting clusters are shown in Figure 31. The same cutoff (1.1Ångström) as in the previous simulations with rigid ligands was used.


-8.5
-9.0
-9.5
-10.0
ΔΔG (kcal/mol)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Cluster


-9
-10
-11
ΔΔG (kcal/mol)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Cluster

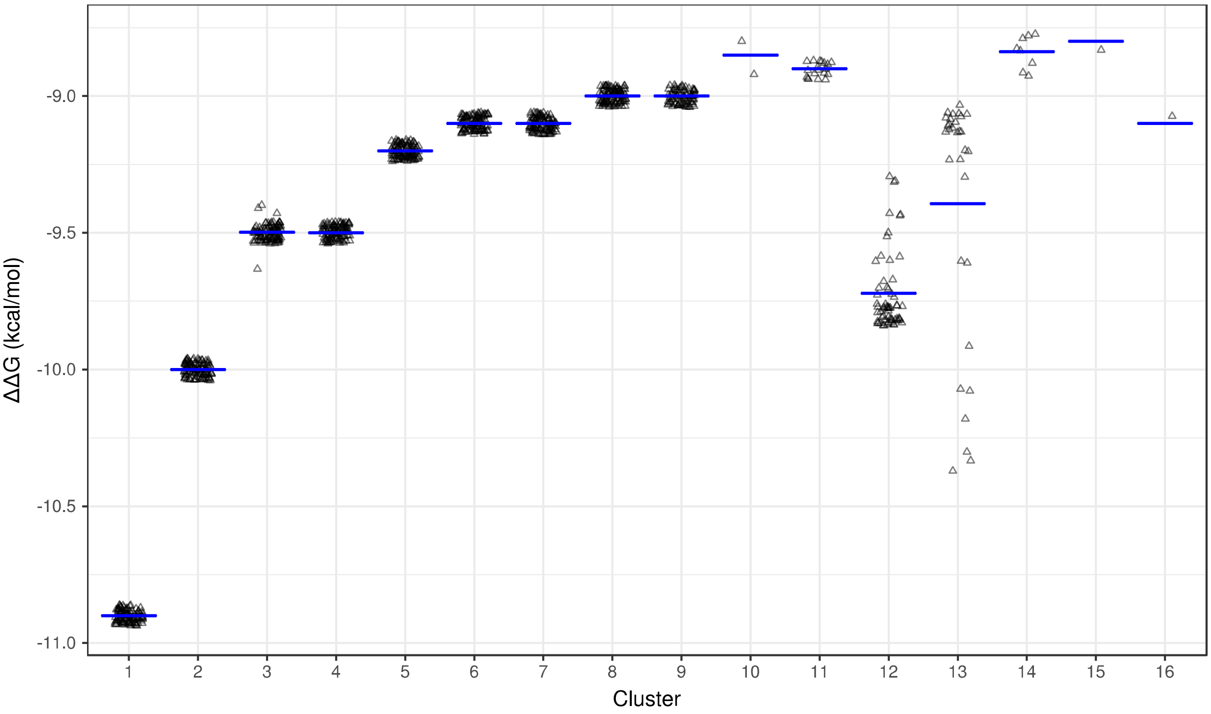
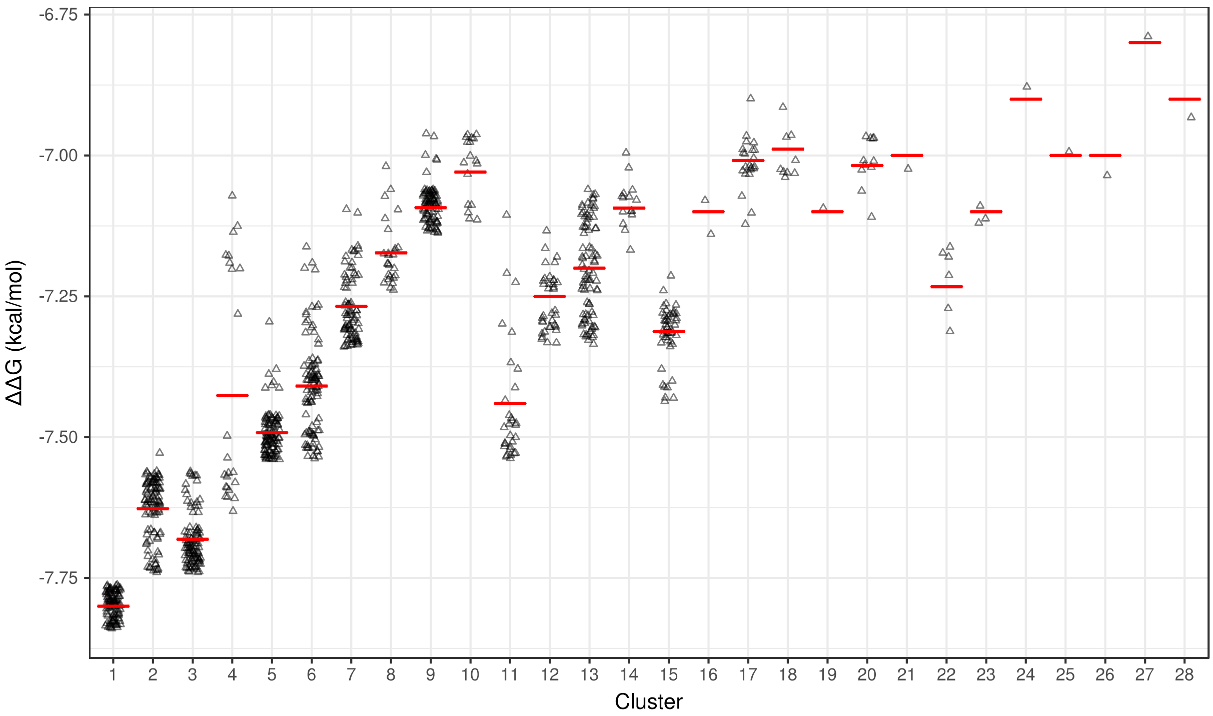


Figure 31. *Scatter plots presenting the ΔΔG value (in kcal/mol) of every model of each cluster, for the redocking simulations of chains A, C and E, after the 91YA mutation, with their rigid ligand from top to bottom. For chain A and C, the ligand crystal ligand pose belongs to the first cluster and for chain E to the second cluster, same as the previous simulations using rigid ligands. Mean value is shown as a line. Notice that the free energy of the bound state has negative value.*

3.3.5 Docking of H1 HA bearing the 91YA mutation using flexible ligands

Same as for the original 4JTV PDB structure, and for the one carrying the 224EA mutation, the ligands were also redocked to their chains after the 91YA treating them as flexible (with nine degrees of freedom, as described in Section 3.2.1.).

Clustered together were conformations that differ by less than 1.3Ångström for chain A and 2.1Ångström for chain C, as done for the previous simulations using flexible ligands, including the control simulation namely the simple redocking simulation using flexible ligands. Again, I didn't focus on chain E, as vina didn't manage to find the crystal pose in the simple redocking experiments, namely the ones with no mutations in the protein, under the parameters set. The dendrograms for this simulations can be accessed here and the MDS interactive plots here. ΔΔGs for these simulations are shown in Figure 32.



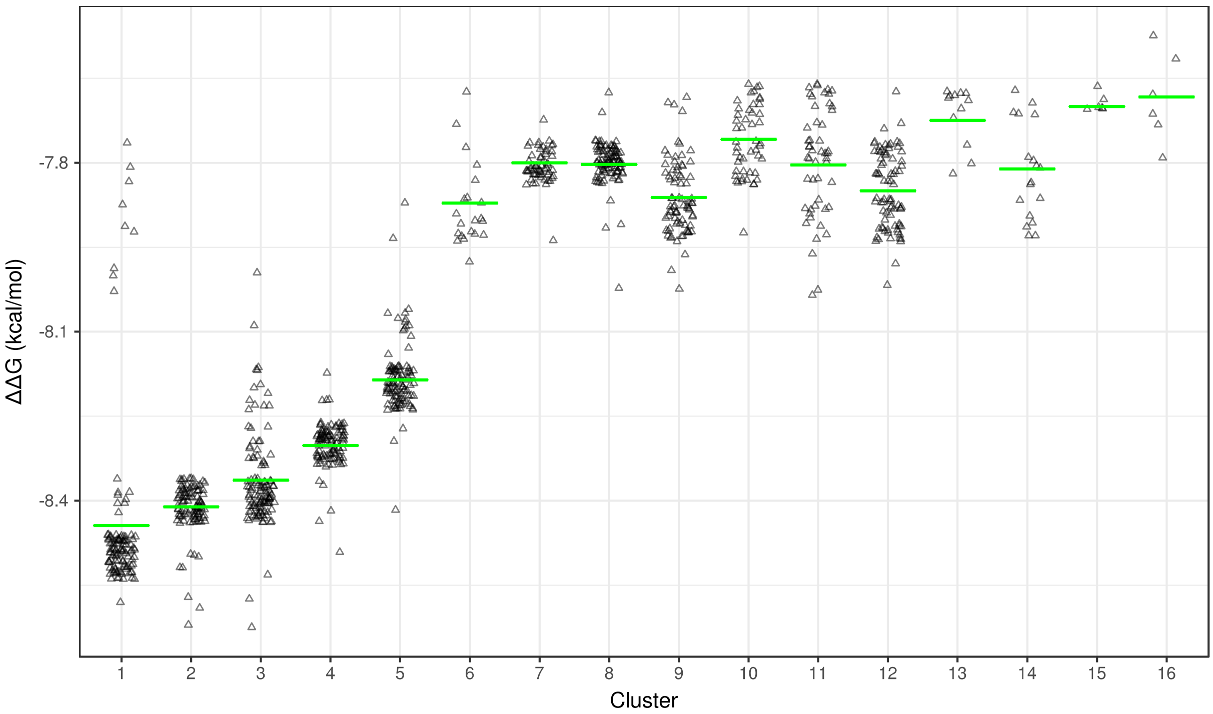


Figure 32. *Scatter plots presenting the ΔΔG value (in kcal/mol) of every model of each cluster, for the redocking simulations of chains A and C, after the 91YA mutation, with their flexible ligand from top to bottom. For chain A the ligand crystal ligand pose belongs to the 15th cluster and for chain E to the 9th cluster, which is different compared to the previous simulations using flexible ligands. Mean value is shown as a line. Notice that the free energy of the bound state has negative value. Also notice that the mean value is not the best representative here because of the many outliers, due to clustering using higher cutoff RMSD.*

4. Summary of results, comparison and discussion

The different cutoffs chosen for every simulation described above, the resulting number of clusters using this cutoff and the cluster the crystal pose of the ligand belongs to are summed in Table 10.

	Cut – off (in Ångström) *Cluster of crystal pose* *Clusters created*		
	Chain A	Chain C	Chain E
Re-docking rigid	**1.1** *1/17*	**1.1** *1/16*	**1.1** *2/21*
Re-docking flexible	**1.3** *2/22*	**2.1** *7/17*	**X**
224EA rigid	**1.1** *1/25*	**1.1** *1/18*	**1.1** *2/17*
224EA flexible	**1.3** *2/26*	**2.1** *2/15*	**X**
91YA rigid	**1.1** *1/17*	**1.1** *1/16*	**1.1** *2/16*
91YA flexible	**1.3** *15/28*	**2.1** *9/16*	**X**

Table 10. *Each of the six simulations in this table was repeated 100 times for each monomer of the trimer of the hemagglutinin molecule. For every 100 repeats, clustering analysis was performed; the cutoffs used for every clustering are shown in bold. The cluster that the ligand belongs to and the total number of clusters that resulted per clustering analysis are also mentioned. Notice that the cutoffs were chosen in a way that the crystal ligand pose would be clustered with its closest group of vina suggested models; this was chosen by studying the dendrograms presenting the hierarchical clustering. Remember that no clusters were created for redocking chain E with its flexible ligand, as vina didn't manage to predict the crystal pose in the redocking experiments.*

It is important to study the consistency with which vina manages to propose a model similar to the crystal pose in the simple redocking experiments; if it succeeds in this control simulation, it is expected that for the same parameters it predicts a probable ligand pose for the docking simulations of the hemagglutinin molecule with a single mutation to the respective ligand. Figure 33 shows the composition of the cluster that the crystal pose belongs to for the redocking/control experiments. Seeing that for the redocking simulations with flexible ligands, vina finds the crystal pose for chain A and C but doesn't rank these models as first, indicates that one cannot blindly trust that the first model of the docking simulation is in fact the correct one.

What's more it seems that the redocking simulations of chain A with its flexible ligand give the more predictable results (as in most of the simulation runs vina manages to find the crystal ligand pose, although it ranks it as the second most probable). This is probably because the ligand of chain A has three rings and is allowed 7 degrees of freedom, while chain C and E ligand have four rings and 9 degrees of freedom.

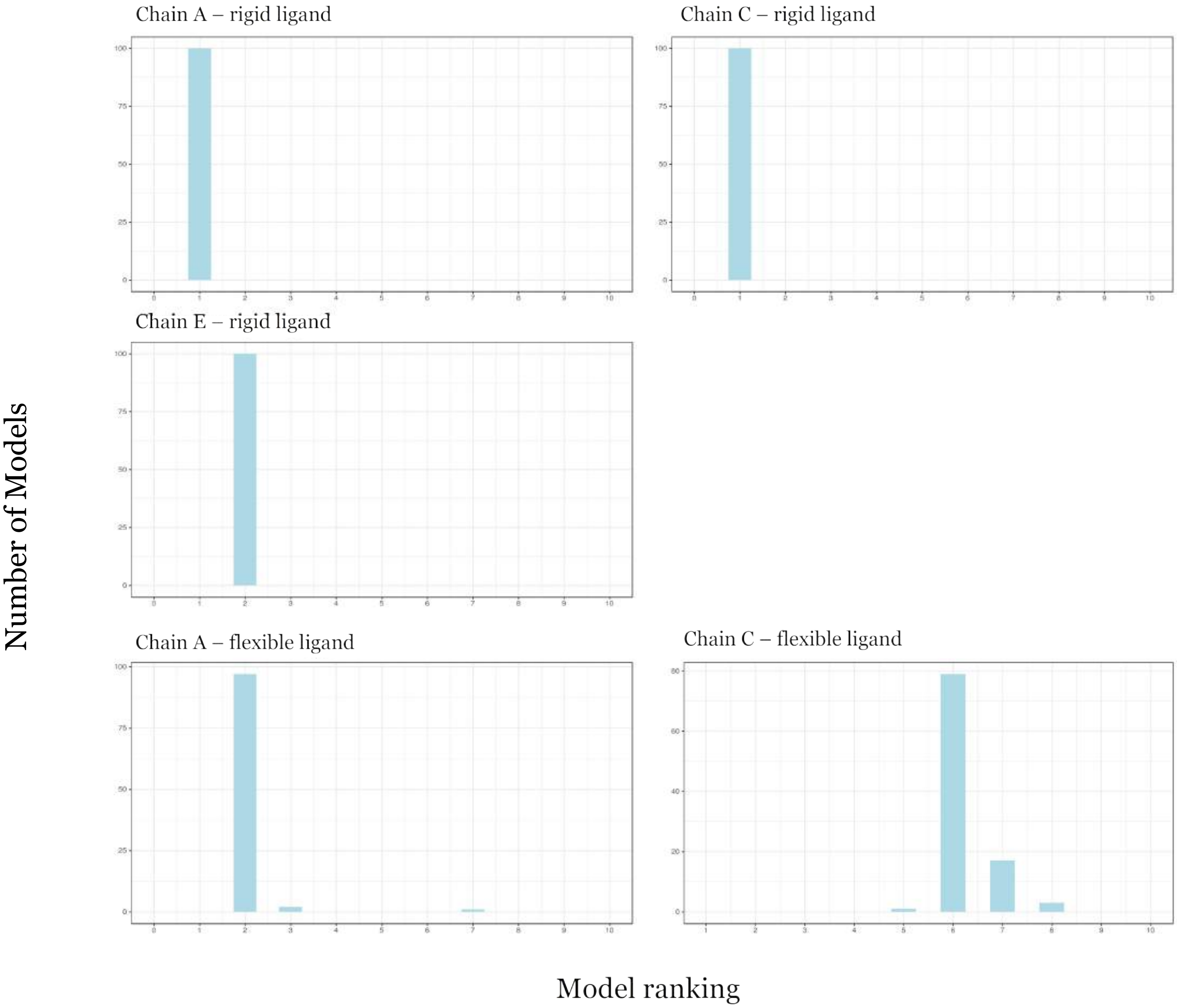


Figure 33. *The composition of the cluster that the crystal pose belongs to in each control simulation. In other words, for every redocking experiment performed, the models of the cluster that contains the crystal pose of the ligand were plotted in histograms. For example, for the simulation redocking chain C with its flexible ligand, one can see that the cluster containing the crystal pose of the ligand (which was cluster 7, as shown in Table 10) contains almost 80 models that were ranked 6th, according to their ΔΔG in the vina run they resulted from, almost 20 models ranked 7th and a few ranked 5th or 8th, indicating that for this round of simulations, vina managed to find the crystal pose most of the times and rank it as the 6th most probable.*

After introducing the 224EA mutation the crystal pose belonged to clusters with almost the same composition as the one in the control experiments, indicating that this substitution doesn't affect the orientation of the ligand in the binding pocket (Figure 34). This is in agreement with the in vitro results of the Koel paper; the 224EA substitution was found to not affect the binding efficiency of the α2,6 – LSTc to the hemagglutinin protein(26).

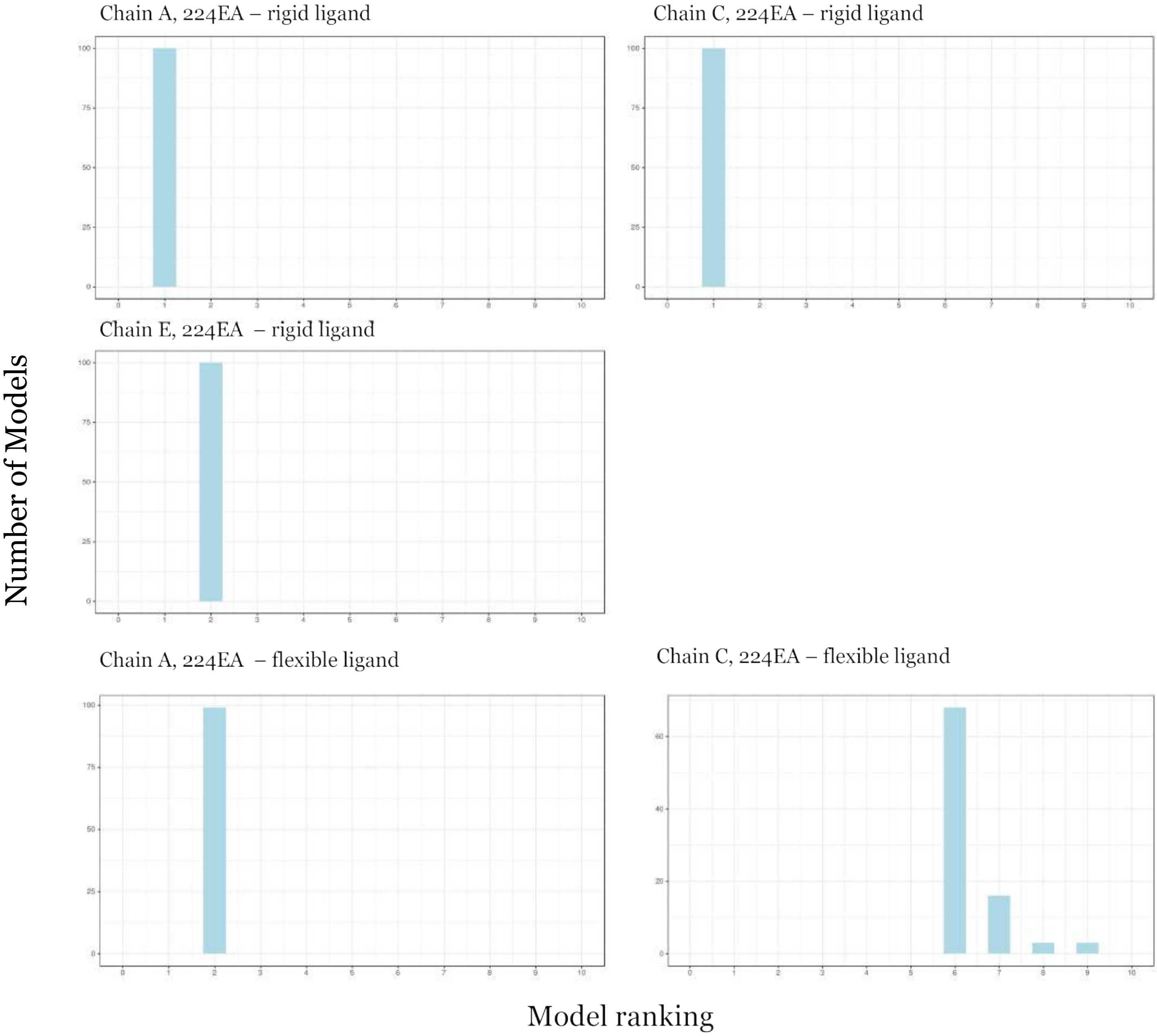


Figure 34. *The composition of the cluster that the crystal pose belongs to in each redocking simulation of hemagglutinin chains carrying the 224EA substitution to their ligands. In other words, for every one of these simulations performed, the models of the cluster that contains the crystal pose of the ligand were plotted in histograms. When compared to Figure 33 one concludes that only minor changes occurred (specifically in the simulation in which Chain C carrying the 224EA mutation was redocked to its flexible ligand). Since the crystal pose is clustered again with models that are ranked the same as in the control experiment, the 224EA didn't affect the orientation of the ligand in the binding pocket.*

However, when introducing the 91YA mutation the crystal pose was not clustered with models with the same ranking, as in the control simulations (Figure 35). What's more, the models the crystal pose was clustered with were ranked lower, meaning they were models that are not considered as probable for these parameters (Figure 35).

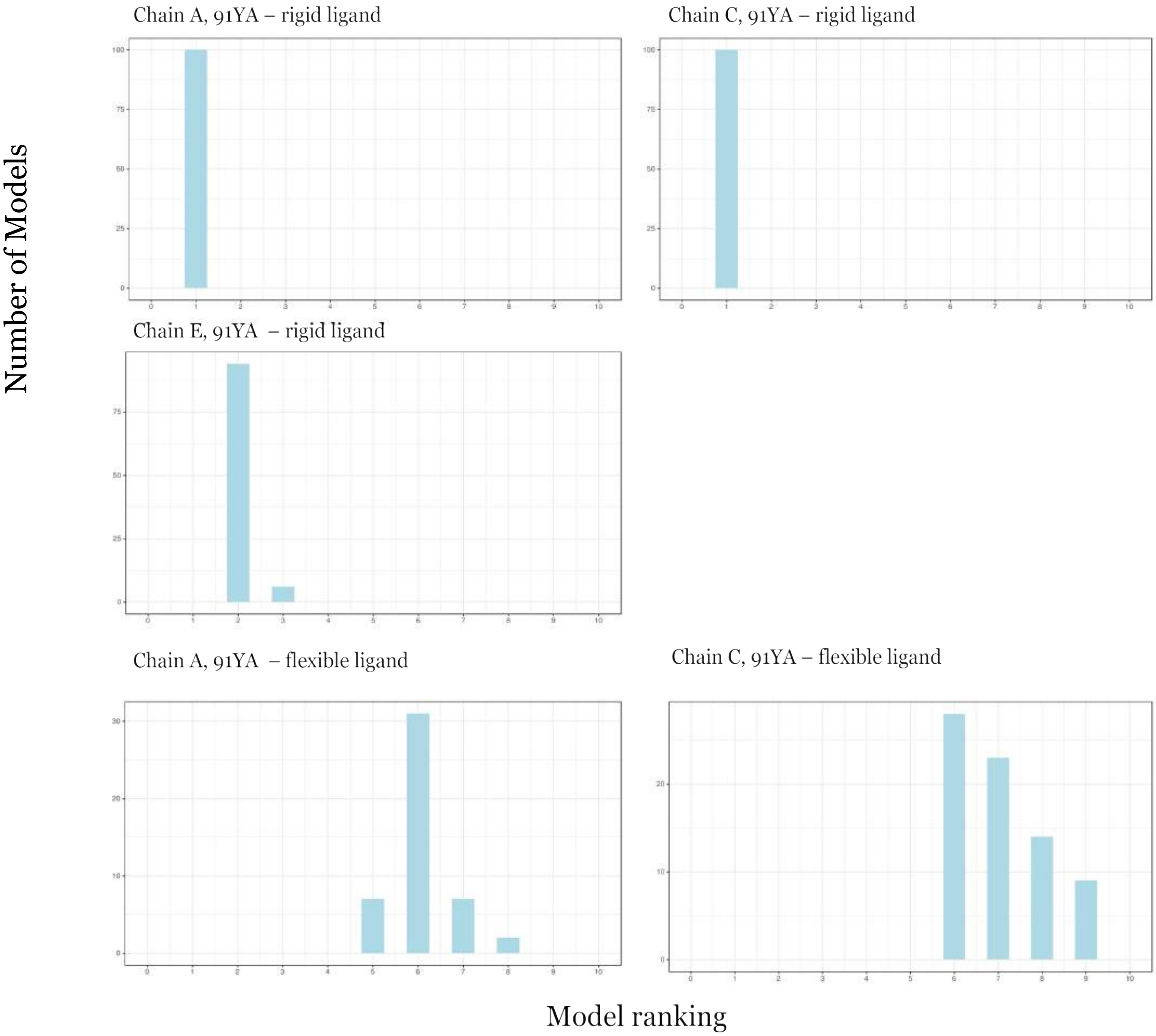


Figure 35. *The composition of the cluster that the crystal pose belongs to in each redocking simulation of hemagglutinin chains carrying the 91YA substitution to their ligands. In other words, for every one of these simulations performed, the models of the cluster that contains the crystal pose of the ligand were plotted in histograms. When compared to Figure 33 one concludes that only minor changes occurred for the rigid ligands. Since the crystal pose is clustered again with models that are ranked the same as in the control experiment for the rigid ligands, the 91YA substitution seems to not affect the orientation of the ligand in the binding pocket. However, for the flexible ligand, the lower ranked models seem to be more densely populated, compared to the orientation of the ligands in the crystal structure (Figure 33).*

Notice how the cluster the crystal pose belongs to doesn't change significantly in comparison with the control simulations when treating the ligand as rigid but does change when treating it as flexible. Since chain A in the control simulations had the most predictable results, lets focus on this one. While in the control simulation in which chain A was docked to its flexible ligand vina managed to find the crystal pose and rank the models similar to the crystal pose high (namely with lower ΔΔG) and specifically second, the crystal pose after the 91YA substitution is clustered with less probable models, indicating that this substitution affects the binding orientation of the ligand (see also Figure 37, left).

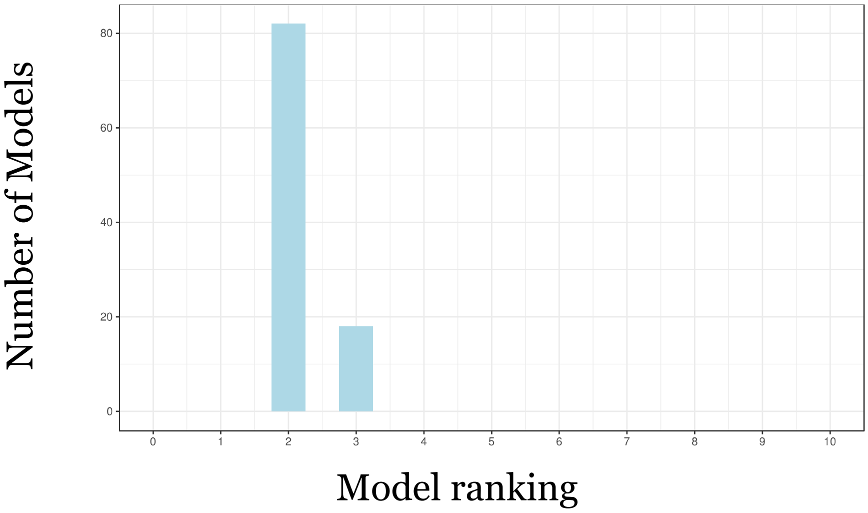


Figure 36. *The composition of the cluster (3rd cluster) that the crystal pose belonged to in the respective control experiment, namely the redocking of chain A to its flexible ligand. The mean ΔΔG of this cluster is -7.681 kcal/mol and the SD is 0.04.*

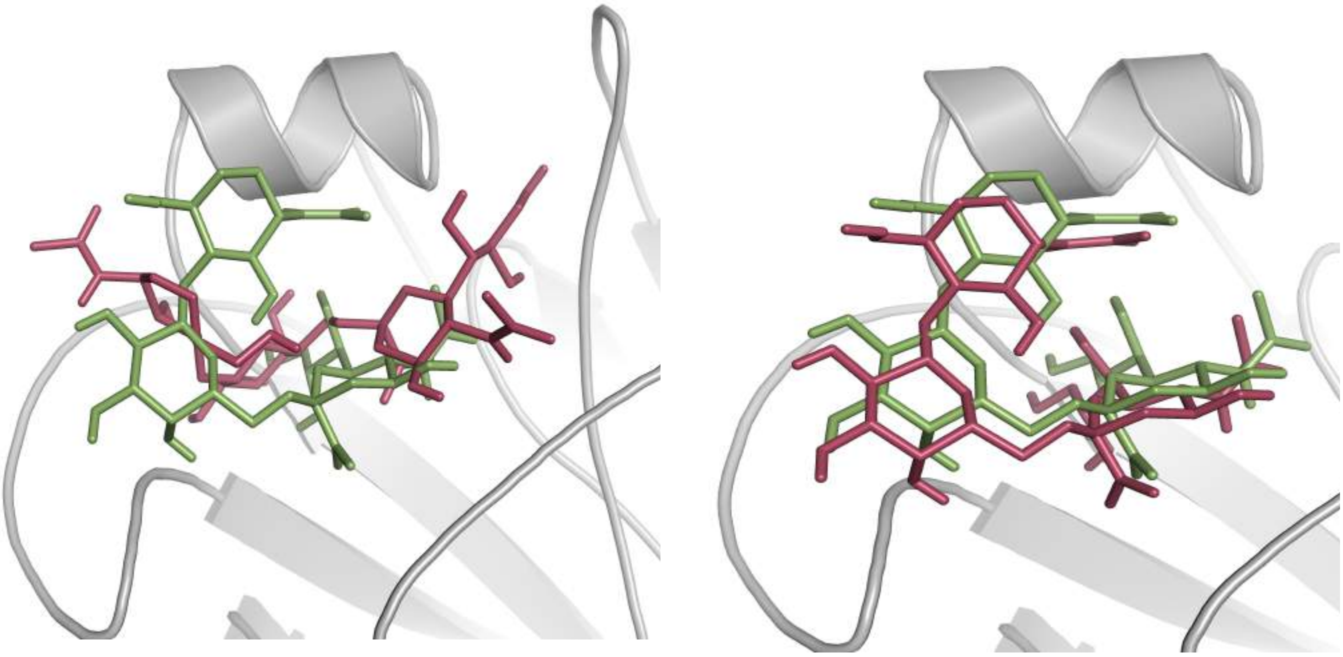


Figure 37. *The second model of the second run of the chain A(91YA) – flexible ligand docking simulation is shown in pink and the original crystal pose of the ligand is in green. The predicted ΔΔG of the model is -7.6 kcal/mol. (left)*

The fifth model of the second run of the chain A(91YA) – flexible ligand docking simulation is shown in pink and the original crystal pose of the ligand is in green. The predicted ΔΔG of the model is -7.5 kcal/mol.

However, in the same run, the 5th model was identical to the crystal pose of the ligand (Figure 37, right) posing the question of whether the second model is actually the most probable ligand pose after the 91YA mutation or the system built doesn't possess enough sensitivity to detect changes caused by a single substitution and the changes in the crystal – pose like models' ranking observed here are a result not related to a biology phenomenon.

5. Remarks

The results of the redocking simulations, especially regarding chain A, with the three – ringed ligand were encouraging, since vina managed to suggest repetitively a model similar to the crystal pose of the ligand and also rank this model among the most probable ones, namely assigning it a lower ΔΔG. Attention should be paid that when using rigid molecules, the system becomes less sensitive to minor changes in the protein, such as single mutations.

However, it is confusing how vina yielded models similar to the crystal pose of the ligand for chain C but not for chain E, although both chains had a ligand with four rings.

What's more, apart from studying the conformation of the models and their ΔΔG of binding, a more extensive study should be performed regarding the bonds between the protein and each suggested vina model.

The protocol described in this project could be used for testing the effects of single substitutions in chain A of the 4JTV PDB structure to the binding efficiency and the orientation of the bound three – ringed ligand, which is part of the α2,6 – SA receptor analogue, LSTc. However, the lack of predictability and consistency in the resulting models, regarding their ranking according to ΔΔG suggests that more suitable parameters should be used, in order to be able to trust the vina results.

Most importantly, it was surprising to me how many questions come up while working on a tiny project, expanding it and adding more experiments to be done and more problems to be solved. The hardest part of the project was to plan the project steps and cope with the – most of the times – unexpected results.

6. References

1. World Health Organization. August 2010. H1N1 in post-pandemic period. http://www.who.int/mediacentre/news/statements/2010/h1n1_vpc_20100810/en/.
2. Yang Y, Halloran ME, Sugimoto JD, Longini IM. 2007. Detecting Human-to-Human Transmission of Avian Influenza A (H5N1). Emerging Infectious Diseases 13(9):1348–1353. http://doi.org/10.3201/eid1309.07-0111.
3. Acheson NH, Fundamentals of molecular virology, 2nd ed, p 210. John Wiley & Sons, Inc., Hoboken, NJ.
4. Sayers EW, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Edgar R, Federhen S, Feolo M, Geer LY, Helmberg W, Kapustin Y, Landsman D, Lipman DJ, Madden TL, Maglott DR, Miller V, Mizrachi I, Ostell J, Pruitt KD, Schuler GD, Sequeira E, Sherry ST, Shumway M, Sirotkin K, Souvorov A, Starchenko G, Tatusova TA, Wagner L, Yaschenko E, Ye J. 2009. Database resources of the National Center for Biotechnology Information. Nucleic Acids Res. 37:D5-15.
5. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. 2009. GenBank. Nucleic Acids Res. 37:D26-31.
6. Centers for Disease Control and Prevention. September 2017. Types of Influenza viruses. https://www.cdc.gov/flu/about/viruses/types.htm.
7. Samji T. December 2009. Influenza A: Understanding the viral life cycle. Yale J Biol Med 82(4): 153-159.
8. Sieczkarski SB, Whittaker GR. 2002. Influenza Virus Can Enter and Infect Cells in the Absence of Clathrin-Mediated Endocytosis. Journal of Virology 76(20):10455–10464. http://doi.org/10.1128/JVI.76.20.10455-10464.2002.
9. Gasparini R, Amicizia D, Lai PL, Bragazzi NL, Panatto D. 2014. Compounds with anti-influenza activity: present and future of strategies for the optimal treatment and management of influenza. Part I: influenza life-cycle and currently available drugs. Journal of Preventive Medicine and Hygiene 55(3):69–85.
10. U.S. Department of Health and Human Services Centers for Disease Control and Prevention. October 2016. Workshop Proceedings Pandemic Influenza—Past, Present, Future: Communicating Today Based on the Lessons from the 1918–1919 Influenza Pandemic, Washington, DC.
11. Viboud C, Simonsen L, Fuentes R, Flores J, Miller MA, Chowell G. 2016 . Global Mortality Impact of the 1957–1959 Influenza Pandemic. The Journal of Infectious Diseases 213(5):738–745. http://doi.org/10.1093/infdis/jiv534.
12. Viboud C, Grais RF, Lafont BAP, Miller MA, Simonsen L. July 2005. Multinational Impact of the 1968 Hong Kong Influenza Pandemic: Evidence for a Smoldering Pandemic. The Journal of Infectious Diseases 192(2):233–248. https://doi.org/10.1086/431150.
13. Gregg MB, Hinman AR, Craven RB. 1978. The Russian Flu.Its History and Implications for This Year's Influenza Season. JAMA 240(21):2260–2263. doi:10.1001/jama.1978.03290210042022.
14. Dawood FS, Iuliano AD, Reed C, Meltzer MI, Shay DK, Cheng PY, Bandaranayake D, Breiman RF, Brooks WA, Buchy P, Feikin DR, Fowler KB, Gordon A, Hien NT, Horby P, Huang QS, Katz MA, Krishnan A, Lal R, Montgomery JM, Mølbak K, Pebody R, Presanis AM, Razuri H, Steens A, Tinoco YO, Wallinga J, Yu H, Vong S, Bresee J, Widdowson MA. September 2012. Estimated global mortality associated with the first 12 months of 2009 pandemic influenza A H1N1 virus circulation: a modelling study. Lancet Infect Dis. 12(9):687 – 695. http://dx.doi.org/10.1016/S1473-3099(12)70121-4.
15. Sun X, Whittaker GR. 2013. Entry of influenza virus. Adv Exp Med Biol 790:72-82. https://doi.org/10.1007/978-1-4614-7651-1_4.
16. World Health Organization. 1980. A revision of the system of nomenclature for influenza viruses: a WHO Memorandum. Bulletin of the World Health Organization, 58(4):585–591.
17. Neumann G, Noda T, Kawaoka Y. 2009. Emergence and pandemic potential of swine-origin H1N1 influenza virus. Nature 459(7249):931–939. http://doi.org/10.1038/nature08157.
18. World Health Organization. February 2014. Recommended composition of influenza virus vaccines for use in the 2014-2015 northern hemisphere influenza season. http://www.who.int/influenza/vaccines/virus/recommendations/201402_recommendation.pdf?ua%CF%AD1.
19. World Health Organization. March 2017. Recommended composition of influenza virus vaccines for use in the 2017- 2018 northern hemisphere influenza season. http://www.who.int/influenza/vaccines/virus/recommendations/201703_recommendation.pdf?ua=1
20. World Health Organization. September 2017. Recommended composition of influenza virus vaccines for use in the 2018 southern hemisphere influenza season. http://www.who.int/

influenza/vaccines/virus/recommendations/201703_recommendation.pdf?ua=1.

21. Drake JW. 1993. Rates of spontaneous mutation among RNA viruses. Proceedings of the National Academy of Sciences of the United States of America 90(9):4171–4175.
22. Connor RJ, Kawaoka Y, Webster RG, Paulson JC. 1994. Receptor specificity in human, avian, and equine H2 and H3 influenza virus isolates. Virology 205(1):17-23. https://doi.org/10.1006/viro.1994.1615.
23. Varki A, Schnaar RL, Schauer R. 2017. Sialic Acids and Other Nonulosonic Acids. *In* Varki A, Cummings RD, Esko JD, Stanley P, Hart GW, Aebi M, Darvil AG, Kinoshita T, Packer NH, Prestegard JH, Schnaar RL, Seeberger PH (ed), Essentials of Glycobiology, 3rd ed, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY. doi: 10.1101/glycobiology.3e.015.
24. Wang Q. 2010. Influenza Type B Virus Hemagglutinin: Antigenicity, Receptor binding and Membrane Fusion. *In* Wang Q, Tao YJ. 2010. Influenza: Molecular Virology, Caister Academic Press, U.K.
25. Pedersen JC. Hemagglutination – Inhibition Test for Avian Influenza Virus subtype identification and the detection and quantitation of serum antibodies to the avian Influenza virus. *In* Spackman E(ed). 2008. Avian Influenza Virus. Human Press,Totowa, NJ.
26. Koel BF, Mögling R, Chutinimitkul S, Fraaij PL, Burke DF , van der Vliet S, de Wit E, Bestebroer TM, Rimmelzwaan GF, Osterhaus AD, Smith DJ, Fouchier RA, de Graaf M. 2015. Identification of amino acid substitutions supporting antigenic change of influenza A(H1N1)pdm09 viruses. J Virol 89(7):3763-75. http://doi.org/10.1128/JVI.02962-14.
27. Zhang W1, Shi Y, Qi J, Gao F, Li Q, Fan Z, Yan J, Gao GF. 2013. Molecular basis of the receptor binding specificity switch of the hemagglutinins from both the 1918 and 2009 pandemic influenza A viruses by a D225G substitution. J Virol 87(10):5949-58. doi:10.1128/JVI.00545-13
28. Branden C, Tooze J. 2000. Introduction to protein structure, 2nd ed. Garland Publishing, Inc. New York, NY.
29. Wilson IA, Skehel JJ, Wiley DC. 1981. Structure of the hemmagglutinin membrane glycoprotein of influenza virus at 3Å resolution. Nature 289:366-373. doi:10.1038/289366a0
30. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. 2000. The Protein Data Bank. Nucleic Acids Res 28(1):235–242.
31. DuBois RM, Aguilar-Yañez JM, Mendoza-Ochoa GI, Oropeza-Almazán Y, Schultz-Cherry S, Alvarez MM, White SW, Russell CJ. 2011. The receptor-binding domain of influenza virus hemagglutinin produced in Escherichia coli folds into its native, immunogenic structure. J Virol 85(2):865-72. http://doi.org/10.1128/JVI.01412-10.
32. Skehel JJ, Wiley DC. 2000. Receptor binding and membrane fusion in virus entry: the influenza hemagglutinin. Annu Rev Biochem 69:531-69. https://doi.org/10.1146/annurev.biochem.69.1.531.
33. Gerhard W, Yewdell J, Frankel ME, Webster R. 1981. Antigenic structure of influenza virus haemagglutinin defined by hybridoma antibodies. Nature. 290(5808):713-7.
34. Koel BF, Burke DF, Bestebroer TM, van der Vliet S, Zondag GC, Vervaet G, Skepner E, Lewis NS, Spronken MI, Russell CA, Eropkin MY, Hurt AC, Barr IG, de Jong JC, Rimmelzwaan GF, Osterhaus AD, Fouchier RA, Smith DJ. 2013. Substitutions near the receptor binding site determine major antigenic change during influenza virus evolution. Science 342(6161):976-9. http://dx.doi.org/10.1126/science.1244730.
35. Wu A, Peng Y, Du X, Shu Y, Jiang T. 2010. Correlation of influenza virus excess mortality with antigenic variation: application to rapid estimation of influenza mortality burden. PLoS Computational Biology 6(8), e1000882. http://doi.org/10.1371/journal.pcbi.1000882.
36. Burke DF, Smith DJ. 2014. A Recommended Numbering Scheme for Influenza A HA Subtypes. PLoS ONE 9(11), e112302. http://doi.org/10.1371/journal.pone.0112302
37. Xu R, McBride R, Nycholat CM, Paulson JC, Wilson IA. 2012. Structural characterization of the hemagglutinin receptor specificity from the 2009 H1N1 influenza pandemic. J Virol 86(2):982-90. http://doi.org/10.1128/JVI.06322-11.
38. Beginner's Guide to the PDBbind Database. 2016. http://www.pdbbind-cn.org/download/pdbbind_2016_intro.pdf.
39. Munster VJ, de Wit E, van den Brand JM, Herfst S, Schrauwen EJ, Bestebroer TM, van de Vijver D, Boucher CA, Koopmans M, Rimmel-zwaan GF, Kuiken T, Osterhaus AD, Fouchier RA. 2009. Pathogenesis and transmission of swine-origin 2009 A(H1N1) influenza virus in ferrets. Science 325:481– 483. http://dx.doi.org/10.1126/science.1177127.
40. Edgar RC. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research 32(5):1792–1797. http://doi.org/10.1093/nar/gkh340.
41. Al-qattan MN1, Mordi MN. 2010. Docking of sialic acid analogues against influenza A

hemagglutinin: a correlational study between experimentally measured and computationally estimated affinities. J Mol Model 16(5):1047-58. doi: 10.1007/s00894-009-0618-7.

42. The Scripps Research Institute. 2007. What is the format of a PDBQT file?. http://autodock.scripps.edu/faqs-help/faq/what-is-the-format-of-a-pdbqt-file/.
43. Trott O, Olson AJ. 2010. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Compute Chem 30;31(2):455-61. http://doi.org/10.1002/jcc.21334.
44. Feinstein WP, Brylinski M. 2015. Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. Journal of Cheminformatics 7:18. http://doi.org/10.1186/s13321-015-0067-5.
45. Sanner MF. 1999. Python: A Programming Language for Software Integration and Development. J. Mol. Graphics Mod 17:pp57-61
46. Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, Olson AJ. 2009. AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility. J. Computational Chemistry 16: 2785-91. http://doi.org/10.1002/jcc.21256.
47. Forli S1, Huey R1, Pique ME1, Sanner MF1, Goodsell DS1, Olson AJ1. 2016. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. Nat. Protoc (5):905-919. http://doi.org/10.1038/nprot.2016.051.
48. R Core Team. 2017. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
49. Wickham H, Francois R, Henry L, Müller K. 2017. dplyr: A Grammar of Data Manipulation. R package version 0.7.2. https://CRAN.R-project.org/package=dplyr.
50. Wickham H. 2009. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.
51. RStudio Team. 2015. RStudio: Integrated Development for R. RStudio, Inc., Boston, MA. http://www.rstudio.com/.
52. Plotly Technologies Inc. 2015. Collaborative data science. Montréal, QC. https://plot.ly.
53. Wickham H. 2007. Reshaping Data with the reshape Package. Journal of Statistical Software 21(12):1-20. http://www.jstatsoft.org/v21/i12/.
54. The Scripps Research Institute. 2013. How-to dock a ligand with many torsions. http://autodock.scripps.edu/faqs-help/how-to/how-to-dock-a-ligand-with-many-torsions.
55. PMV: Sanner MF. 1999. Python: A Programming Language for Software Integration and Development. J. Mol. Graphics Mod. 17: 57-61.
56. UniProt Consortium. 2017. UniProt: the universal protein knowledgebase . Nucleic Acids Res. 45:D158-D169.
57. Spek AL. 2009. Acta Cryst. D65:148-155. http://www.cryst.chem.uu.nl/spek/platon/.

7. Appendix

A1. Hemagglutinin numbering conversions

Numbering conversions from H3 to H1 numbering in order to renumber HA sequences according to a cross – subtype numbering scheme proposed by Bruke and Smith. The corresponding position in 4JTV PDB Hemagglutinin is also included, to make it easier to explore this specific structure. This table is useful for reading the literature as well, as H3 numbering is conventionally used in the field for all subtypes. The representative amino acid in each position for H1N1pdm and H3 is also included(36).

H3	H1N1pdm	4JTV
92K	84S	90
98Y	91Y	97
116G	109S	115
130V	126H	132
131T	127D	133
136S	133T	139
138A	135A	141
145S	142K	148
151L	148L	154
152N	149I	155
153W	150W	156
154L	151L	157
155T	152V	158
156K	153K	159
157S	154K	160
158G	155G	161
159S	156N	162
183H	180H	186
186S	183S	189
189Q	186A	192
190E	187D	193
193S	190S	196
195Y	192Y	198

H3	H1N1pdm	4JTV
220R	217R	223
222W	219K	225
225G	222D/G	228D
226L	223Q	229
227S	224E	230
228S	225G	231
261R	258E	264

A2. **crossDCD** script

The crossDCD script was used to calculate the RMSD matrices. The commands used were of the form:
./crossDCD ligand.psf ligand.dcd ligand.dcd 1 “-atmid HEAVY”
the DCD files used as input are the same file. In contrast to what the first comments declare, the script has been modified to not perform least squares fitting between the frames.

```
#!/bin/tcsh -f

if ( $# > 5 || $# < 3 ) then
echo " "
echo "[1m[37mUsage   : [mCrossDCD <PSF_file> <DCD_1> <DCD_2> [step] [flags]"
echo "[1m[37mSummary : [mCalculate a 2D matrix containing the rms
deviations"
echo "          (after  least  squares  fitting) between  the frames"
echo "          of two DCD files. If <step> is defined, then instead"
echo "          of using each and every frame of the  two DCDs, only"
echo "          consider every <step>th frame.  DCD_1 and  DCD_2 can"
echo "          well be the one and same file. The  matrix  will  be"
echo "          written to a  file ( crossDCD.matrix ).  To plot the"
echo "          results use 'carma - < crossDCD.matrix' and view the"
echo "          resulting postscript file."
echo "          If you want to pass flags to the carma runs  (things"
echo "          line -segid or -atomid), you will have to define the"
echo "          <step> (even if it is 1)  and  then  define  carma's"
echo '          flags enclosed in double quotes (ie " ...flags...").'
echo "          THIS VERSION IS MODIFIED TO_NOT_ FIT                 "
echo " "
exit
endif

if (! -es $1 || ! -es $2 || ! -es $3) then
echo "[1m[31mMissing PSF or DCD files ? [m"
exit
endif

if ( -es "carma.fitted.dcd" ) then
echo "[1m[31mA file 'carma.fitted.dcd' already exists. Please rename it. [m"
exit
endif

if ( -es "carma.fit-rms.dat" ) then
echo "[1m[31mA file 'carma.fit-rms.dat' already exists. Please rename it.
[m"
exit
endif

if ( -es "crossDCD.matrix" ) then
echo "[1m[31mA file 'crossDCD.matrix' already exists. Please rename it. [m"
exit
endif

if ( $%4 ) then
echo "Step for frame selection set to $4"
set step = $4
else
set step = 1
endif

if ( $%5 ) then
echo "Flags to pass to carma : $5"
endif
```

```
set intra = 0
if ( $2 == $3 ) then
echo "DCD_1 and DCD_2 are identical. Will calculate intra-DCD rmsds."
set intra = 1
endif

set numframes1 = `catdcd -num $2 | grep 'Total frames' | awk '{print $3}'`
set numframes2 = `catdcd -num $3 | grep 'Total frames' | awk '{print $3}'`
echo "DCD_1 contains $numframes1 frames"
echo "DCD_2 contains $numframes2 frames"


if ( $intra ) then
########## INTRA RMSD

touch crossDCD.matrix
set points = 0
set ref = 1
while ( $ref <= $numframes1 )
eval "carma $5 -fit -nofit -ref $ref -step $step $2 $1"
awk '{print $2}' carma.fit-rms.dat > $$.1.tmp
paste crossDCD.matrix $$.1.tmp > $$.2.tmp
mv -f $$.2.tmp crossDCD.matrix
rm -rf $$.1.tmp $$.2.tmp
@ ref += $step
echo -n "."
@ points++
if ( $points % 50 == 0 ) then
echo " "
set points = 0
endif
end
rm -rf carma.fit-rms.dat
rm -rf carma.fitted.dcd
exit

else
########## CROSS RMSD
## to avoid continuous catdcd runs, we paste the two
## DCD files, and then use carma's -ref and -first flags
## to save the day.

echo -n "Merging the two DCD files ..."
catdcd -o cross$$.dcd $2 $3 >& /dev/null
echo "done."
@ firstframe = $numframes1 + 1
touch crossDCD.matrix
set points = 0
set ref = 1
while ( $ref <= $numframes1 )
eval "carma $5 -fit -nofit -ref $ref -first $firstframe -step $step cross$
$.dcd $1"
awk '{print $2}' carma.fit-rms.dat > $$.1.tmp
paste crossDCD.matrix $$.1.tmp > $$.2.tmp
mv -f $$.2.tmp crossDCD.matrix
rm -rf $$.1.tmp $$.2.tmp
@ ref += $step
echo -n "."
@ points++
if ( $points % 50 == 0 ) then
echo " "
```

```
set points = 0
endif
end
rm -rf carma.fit-rms.dat
rm -rf carma.fitted.dcd
rm -rf cross$$.dcd
exit

endif


exit
```

A3. pdb2psf script

The pdb2psf script was used to produce the psf files needed for the calclulation of the RMSD matrices.

```
#!/usr/bin/perl -w

#
# Open input-output files
#
if ( @ARGV == 1 )
  {
    if ( $ARGV[0] =~ /(\w+)\.(p|P)(d|D)(b|B)/ )
      {
        $outname = $1 . ".psf";
        open( IN , $ARGV[0] ) or die "Can not open input file\n";
        open( OUT, ">$outname" ) or die "Can not open output file\n";
      }
    else
      {
        print "Usage: pdb2psf in.pdb out.psf\n";
        exit;
      }
  }
elsif ( @ARGV == 2 )
  {
        open( IN , $ARGV[0] ) or die "Can not open input file\n";
        open( OUT, ">$ARGV[1]" ) or die "Can not open output file\n";
  }
else
  {
    print "Usage: pdb2psf in.pdb out.psf\n";
    exit;
  }

print OUT "PSF\n\n";
print OUT "       2 !NTITLE\n";
print OUT " REMARKS This is a _pseudo_ PSF file for sole use with the
program carma.\n";
print OUT " REMARKS It will not work with any other PSF-reading
program.\n\n";

$nof_atoms = 0;
while ( $line = <IN> )
{
  if ( $line =~ /^ATOM\s*(\d*)\s*(\w*)\s*(\w*).(.)\s*(\d*)/ )
    {
      $nof_atoms++;
    }
  elsif ( $line =~ /^HETATM\s*(\d*)\s*(\w*)\s*(\w*).(.)\s*(\d*)/ )
    {
      $nof_atoms++;
    }
}

printf OUT "%8d !NATOM\n", $nof_atoms;

print "Found $nof_atoms atoms. Writing ...\n";

close( IN );
open( IN , $ARGV[0] );

while ( $line = <IN> )
{
  if ( $line =~ /^ATOM\s*(\d*)\s*(\w*)\s*(\w*).(.)\s*(\d*)/ )
    {
      printf OUT "%8d %1s%5d    %-5s%-5sDUMMY  0.000000        0.0000
0\n", $1, $4, $5, $3, $2;
    }
  elsif ( $line =~ /^HETATM\s*(\d*)\s*(\w*)\s*(\w*).(.)\s*(\d*)/ )
    {
      printf OUT "%8d %1s%5d    %-5s%-5sDUMMY  0.000000        0.0000
0\n", $1, $4, $5, $3, $2;
    }

}
```