Aus der Virologischen Klinik, Universität des Saarlandes, Homburg/Saar
und dem
Luxembourg Institute of Health, Esch-sur-Alzette, Luxemburg

No escape mutants and decreased variability of viral quasispecies after immunization against long alpha helix of pandemic H1N1 in a mouse model study

Keine Fluchtmutanten und verminderte Variabilität der viralen Quasispezies nach Impfung gegen die lange Alpha-Helix von pandemischem H1N1 im Mausversuch

Dissertation

zur Erlangung des Grades eines Doktors der Medizin
der Medizinischen Fakultät
der UNIVERSITÄT DES SAARLANDES

2018

vorgelegt von
Nastasja Christine Hauck
Geb. am 06. Januar 1989 in Greifswald

I. Index of contents

II. Index of figures

III. Index of tables

IV. Index of amino acids

Amino acid	Single letter code
Alanine	A
Arginine	R
Asparagine	N
Aspartic acid	D
Cysteine	C
Glutamic acid	E
Glutamine	Q
Glycine	G
Histidine	H
Isoleucine	I
Leucine	L
Lysine	K
Methionine	M
Phenylalanine	F
Proline	P
Serine	S
Threonine	T
Tryptophan	W
Tyrosine	Y
Valine	V

V. Index of abbreviations

Abbreviation	Full name
aa	amino acids
BAM	binary alignment map
bp	base pairs
DNA	deoxyribonucleic acid
dNTP	deoxyribonucleotide triphosphate
dpi	days post infection
HA	hemagglutinin
HxNx	Hemagglutinin x, Neuraminidase x
Hz	Hertz
IAV	Influenza A virus
LAH	long alpha helix
M	matrix protein
MDCK	Madin Darby canine kidney
MID	mouse identifier
MLD	mouse lethal dose
NA	neuraminidase
NGS	next generation sequencing
NP	nucleoprotein
PCA	principle component analysis
PCR	Polymerase chain reaction
pH1N1	pandemic H1N1
RNA	ribonucleic acid
rpm	rounds per minute

SEM	standard error of the mean
TCID	Tissue culture infection dose
UID	unique identifier
V	Version
VGM	virus growth medium
VLP	virus like particle
WHO	World Health Organization

1 Abstract (English and German)

1.1 Abstract

For an increase in pandemic preparedness and to overcome yearly efforts and costs for the production of seasonal influenza vaccines, new approaches for the induction of broadly protective and long-lasting immune responses have been developed during the past decade. It is critical to understand the evolution of influenza viruses in response to Universal Influenza Vaccines. Antibody pressure on conserved domains may increase their variability and lead to the rise of escape mutants. Here we used a mouse challenge model with a vaccine construct targeting the long alpha helix of hemagglutinin stalk to identify alterations in quasispecies composition. The viral ribonucleic acid from lungs of vaccinated and non-vaccinated mice was extracted and analyzed using a next generation sequencing approach. The viral ribonucleic acid in the supernatant of infected Madin Darby Canine kidney cells was used as comparison. The vaccine used elicited significant seroconversion and protection against homologous and heterologous influenza virus strains in mice. The vaccine not only significantly reduced lung viral titers, but also induced a well-known bottleneck effect by decreasing virus diversity. In contrast to the classical bottleneck effect, here we showed a significant increase in the frequency of viruses with amino acid sequences identical to that of vaccine targeting the long alpha helix domain. No emergence of escape mutants in significant quantity was detected after vaccination. These findings support the potential of targeting the conserved long alpha helix domains and add to the hope of having a potent and safe Universal Influenza Vaccine in the near future.

1.2 Zusammenfassung

Im Sinne einer besseren Vorbereitung auf die nächste Influenza-Pandemie und um jährliche Anstrengungen und Kosten für die Impfstoffproduktion gegen saisonale Influenza zu reduzieren, werden seit den letzten etwa zehn Jahren neue Ansätze für die Induzierung breiter, schützender und langanhaltender Immunantwort unternommen. Es ist dabei entscheidend die Evolution der Influenzaviren in Reaktion auf solche Universellen Grippeimpfungen zu verstehen. Der Stress durch Antikörper, die weitgehend konservierte Domänen enthalten, könnte zu vermehrter Variabilität und der Entstehung von Fluchtmutanten führen. In einem Mausinfektionsversuch mit einem Impfstoffkonstrukt, das die lange Alpha-Helix des Hämagglutinin-Stiels angreift, wurden die Veränderungen der Quasispezies untersucht. Die virale Ribonukleinsäure wurde aus den Lungen von geimpften und nicht geimpften Mäusen extrahiert und mit einem Next-Generation-Sequenzierungsverfahren analysiert. Die virale Ribonukleinsäure aus dem Überstand von infizierten Nierenzellkulturen wurde als Vergleich herangezogen. Der Impfstoff führte in Mäusen zu signifikanter Serokonversion und Schutz gegen homologe und heterologe Influenzaviren. Die Impfung reduzierte nicht nur den Virustiter in den Lungen sondern induzierte auch den bekannten Flaschenhals-Effekt mit einer signifikanten Abnahme der Virusdiversität. Im Gegensatz zum klassischen Flaschenhals-Effekt konnte hier eine Zunahme des Virusanteils mit der gleichen Aminosäurensequenz wie der verwendete Impfstoff gezeigt werden, der die lange Alpha-Helix angreift. Nach der Impfung konnten keine Fluchtmutanten in signifikanter Menge gefunden werden. Diese Ergebnisse unterstreichen das Potential, die konservierten Regionen der langen Alpha-Helix ins Visier zu nehmen und machen Hoffnung auf eine sichere und wirkungsvolle Universelle Grippeimpfung in naher Zukunft.

2 Introduction

2.1 Influenza

2.1.1 Epidemiology

Influenza is a respiratory disease that affects humans throughout the world. About 3 to 5 million suffer from severe illness and about 250,000 to 500,000 deaths occur due to influenza every year (World Health Organization, 2014). Consequences include high financial burden due to work time loss, school absenteeism and strain of health systems. Furthermore, adding to yearly epidemics which occur usually during winter (World Health Organization, 2014), influenza pandemics are possible and have occurred four times within the last 100 years (Paules et al., 2017; Shaw et al., 2013). The so called Spanish flu pandemic from 1918/19 has caused up to 50 million deaths (Paules et al., 2017), estimated in various sources, more than World War First (Tucker, 2005). In 1957 the Asian Pandemic was caused by an H2N2 strain, the 1968 Hong Kong pandemic by an H3N2 and the most recent pandemic, in 2009, was caused by an H1N1 and has been referred to as Swine Flu pandemic (Paules et al., 2017). Predictions about when the next pandemic will strike are hard to make but researchers seem to agree that there will be a next pandemic (Barclay, 2017; Wright et al., 2013).

2.1.2 Clinical picture

Influenza is a respiratory disease and onset of symptoms is usually very sudden with high fevers, coughs, muscle and joint pains with an incubation period of two days (World Health Organization, 2014). Super infection with bacteria is possible which can cause higher mortality. In seasonal epidemics mortality lies around 0.1% which in pandemics can rise to 2.5% (Wright et al., 2013). Morbidity and mortality are highest in the old, the very young, pregnant women and immune-compromised persons (World Health Organization, 2014). In pandemics young adults can be significantly affected as well which could be shown in the 1918 pandemic (Wright et al., 2013).

2.1.3 Diagnosis

Clinical signs overlap with other respiratory diseases (Dwyer et al., 2006). Therefore, diagnosis by clinical signs alone is linked to low reliability and usually laboratory testing is needed to confirm influenza virus as causative agent (Chamberlain, 2009). Possible laboratory tests include viral culture, antigen detection and nucleic acid testing (Paules et al., 2017). There are rapid influenza diagnostic tests available which play an important role in clinical settings but reach a sensitivity of only 40% to 70% (Wright et al., 2013), and specificity of 90% to 95% ("CDC flu diagnosis clinician guidance," n.d.).

2.1.4 Therapy

Non-specific treatment includes medication against fevers and pain medication against muscle and joint pains. Targeting influenza virus specifically, there are two different approaches available, neuraminidase inhibitors and M2 proton channel blockers (World Health Organization, 2014). Both should be administered during the first 48 hours of infection to reach the best efficacy (World Health Organization, 2014). Neuraminidase inhibitors, like Oseltamivir and Zanamivir, act on one of the surface proteins, neuraminidase, that severs the connection between the new budding virions and the cell surface. Inhibition of neuraminidase hinders the liberation of new virions and slows through this the rapid growth of virus population (Mckimm-Breschkin, 2013; Wright et al., 2013). M2 proton channel blockers, Amantadine and Rimantadine, interfere with acidification of the virion that is essential for viral uncoating (Pinto et al., 2006). Anti-drug resistance has been monitored and high levels have been recorded during the last seasons for M2 channel blockers, limiting effectiveness and usage (World Health Organization, 2014). Resistance to available anti-influenza medication stresses the need for preventive measures.

2.1.5 Prevention

An effective way of prevention lies in vaccination. The first influenza vaccines were administered in the 1940s (Francis et al., 1945). Influenza vaccines since then and up to present times are seasonal. This is to say, that due to antigenic drift composition of vaccines need to be reformulated almost every season (World Health Organization, 2014) and adapted to circulating strains. Selection of vaccine strains is based on extensive WHO surveillance but some uncertainty remains (Gerdil, 2003; World Health Organization, 2014). Long production time frames slow the reaction in case of

pandemics. In the 2009 pandemic first infections were reported in February in Mexico and the first vaccination was available in September 2009 (reviewed in (Wright et al., 2013)). This is still relatively fast but a gap of half a year without vaccination remains.

People recommended to receive vaccination include pregnant women, people with an age under five or over 65, with chronic medical conditions and health-care workers (World Health Organization, 2014).

2.1.6 Pandemic preparedness

Pandemic preparedness means taking into account all we can predict about the next pandemic and were possible taking action to limit effects and damages. Governments like Germany ("Informationen zur neuen Grippe," n.d., "Nationaler Pandemieplan. Teil I," n.d.) and Great Britain (Barclay, 2017) keep stockpiles of neuraminidase inhibitors as one key factor in pandemic preparedness. Concerns remain whether the next pandemic strain might be resistant to neuraminidase inhibitors, whether at the time of the next pandemic the stockpiled medication is still within its expiration period (Fock et al., 2001) and whether the stockpiled amounts might be sufficient. Often delayed diagnosis and the above mentioned points stress the need for more preventive actions against influenza to achieve greater pandemic preparedness.

The best prevention, therefore, lies in effective vaccination (Florian Krammer et al., 2013; Paules et al., 2017; Wright et al., 2013) and is discussed below.

2.2 Influenza virus

2.2.1 Types and subtypes

Influenza virus belongs to the family of orthomyxoviridae viruses. Carrier of influenza disease are three known influenza virus types labeled A, B and C. Type A can infect humans, but also several animals like birds, swine, dogs and horses. Subtype B is known to cause infections in humans, and has been isolated in seals as well (Wright et al., 2013). Type C causes only mild symptoms and is of little relevance (Paules et al., 2017). Influenza A is further divided into subtypes according to their hemagglutinin (HA) and neuraminidase (NA). So far 18 hemagglutinin subtypes and 11 neuraminidase subtypes have been discovered (Tong et al., 2013). Naming of influenza A subtypes follows Hx and Nx, i.e. H1N1 or H3N1. The hemagglutinin subtypes fall into two groups, with H1, H2, H5, H6, H8, H9, H11, H12, H13, H16, H17 and H18 in group 1 and group 2 containing H3, H4, H7, H10, H14 and H15 (Joyce et al., 2016; Medina et

al., 2011). Influenza B comprises two main lineages, Yamagata and Victoria. Up to date only influenza A pandemics have been recorded.

2.2.2 Drift and shift

Two phenomena of influenza viruses are frequently occurring shifts and drifts. Drift describes small changes in influenza genome that occur due to high rates of mutations and antibody selective pressure (Florian Krammer et al., 2013), while shift means the new combination of surface proteins. During the packaging of new virions in a simultaneously with two strains infected host cell, exchanges of surface proteins are possible and can lead to new hemagglutinin and neuraminidase combinations. Without preexisting immunity, new combinations can lead to pandemics (Medina et al., 2011; Paules et al., 2017).

2.2.3 Structure

Influenza virions are enveloped and their genetic material is comprised of eight negative-sense RNA single-strand molecules. By alternative splicing and alternative open reading frames (Shaw et al., 2013) a number of up to 17 proteins can be folded (Paules et al., 2017). Important features on the virus surface are two glycoproteins, hemagglutinin and neuraminidase. Hemagglutinin mediates binding to host cells by recognizing sialic acid residues and enables fusion between viral and endosomal membrane (Shaw et al., 2013). Its appearance is divided into a globular head, which is the binding site for most neutralizing antibodies, and a stalk or stem region (in this work referred to as “stalk”) which is highly conserved. Only recently, neutralizing antibodies against hemagglutinin stalk have been discovered (Joyce et al., 2016; Laursen et al., 2013; Shaw et al., 2013). (Laursen et al., 2013)The second glycoprotein on the virus surface, neuraminidase, enables release of new virions from cell surface (Paules et al., 2017).

2.2.4 Reproduction cycle

The reproduction cycle begins with attachment of virus to a host cell, mediated by hemagglutinin through binding to a sialic acid residue. Entry occurs through endocytosis. Fusion of viral and endosomal membrane is facilitated through a conformational change of hemagglutinin and content of the virus can enter the host cell through the created pore. Viral ribonucleoproteins are transported into the cell’s nucleus, where transcription and replication of virus genome ensues. The newly synthesized viral proteins are assembled and transferred to the cytoplasm. Budding of

new virions follows. These new virions are actively released from the host cell through neuraminidase activity. Released virions can proceed to infect new cells, beginning the cycle anew (Shaw et al., 2013).

2.2.5 Quasispecies

An important characteristic of influenza virus is its error prone RNA polymerase and missing proof reading ability with -5x10^-4 to -8x10^-3 nucleotide substitutions per site per year (reviewed in (Wright et al., 2013)). High mutation rates lead to different variants of virus simultaneously present in the same host. A mutation can induce greater or lesser fitness and a variant with similar fitness as the "main" or originally infecting variant can exist in the quasispecies as minor variant. When circumstances change, antiviral medication application, presence of antibodies, minor variants with better adaptation to these restrictions, for example resistance to the applied antiviral drug, can expand to become the major variant within the host. This event is called a bottleneck and mutations that allow the virus to escape treatment and proliferate are called escape mutants (Domingo et al., 2012).

When talking about quasispecies and evading mechanisms the language used lets the virus appear a thinking and calculating being. This is of course not the case. Virus evolution follows simple biological patterns of niches and bottlenecks (Domingo et al., 2012; Wright et al., 2013). Words used in this work that imply thinking and planning on the virus's part are used for clarification and illustration alone.

2.3 Influenza Vaccines

2.3.1 Seasonal influenza vaccine

Vaccination against influenza started in the 1940s (Francis et al., 1945). There are inactivated and live attenuated vaccines available against seasonal influenza viruses (Wright et al., 2013). Each year the WHO decides for the Northern hemisphere in February and for the Southern hemisphere in September which strains should be included (World Health Organization, 2014). The vaccine production finishes around six months later so that vaccination can commence before the new influenza season starts (Gerdil, 2003). Long time frames between vaccine strain selection and start of influenza season carry the risk of mismatches between vaccine strain and circulating strain (Wright et al., 2013).

Vaccines are either trivalent or quadrivalent, with the former containing two influenza A subtypes and one lineage B subtype and the latter one containing in addition the other lineage B subtype. As an example, for the upcoming season 2017/2018 in the Northern hemisphere strains chosen by the WHO to be included in influenza vaccines are 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. For the quadrivalent vaccine a B/Phuket/3073/2013-like virus is recommended in addition ("WHO vaccine recommendations," n.d.).

2.3.2 Beyond seasonal influenza vaccine

Above mentioned obstacles and costs concerning the vaccine production and application have let to research efforts going beyond the seasonal vaccine. As explained in the previous section vaccine effectiveness is very narrow and strain specific. In the example of H1N1 a specific strain (in this year's vaccine A/Michigan/45/2015 (H1N1)pdm09-like virus) is chosen for vaccine production. Unfortunately, small changes in the genome through drift may result in low protection. A vaccine that would protect against all pdm09-like viruses would be an improvement and could well mean less frequent administration of vaccine needed. Researchers are aiming even beyond this and search for ways of not only protecting against one subtype but its group. That is to say, that a vaccine would protect against more than one subtype of group 1 for example (i.e. H1N1 and H2N2). In addition, the added protection against group 2 strains would be of interest and in a next step against pandemic viruses as well (Figure 1). Definition of what is a "Universal Influenza Vaccine" has not reached a clear consensus, but protection going beyond the seasonal strain has been termed "broadly protective" (Nachbagauer et al., 2017; Zhang et al., 2014). In addition, Nachbagauer et al. describe a "universal" vaccine as one that confers protection against all influenza A and B viruses (Florian Krammer, 2015; Nachbagauer et al., 2017).

All vaccines giving more protection than can be reached with the seasonal vaccine and that need to be administered less frequently signify improvement on this front and might be a step on the way to a truly universal vaccine.

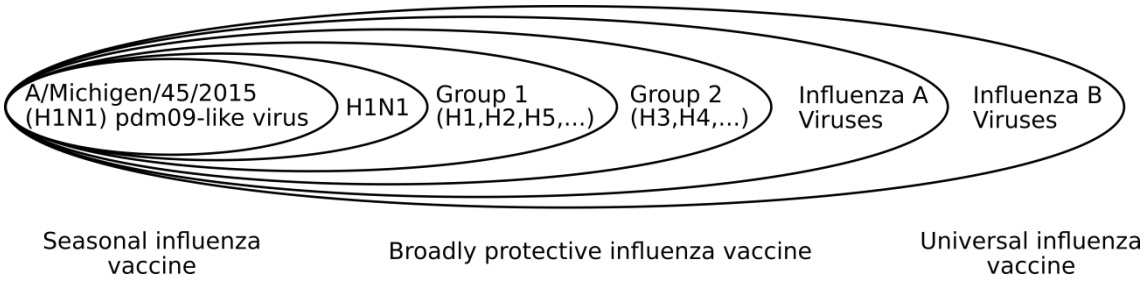

Figure 1: Beyond seasonal influenza vaccine. *Shown with the example of this year's H1N1 strain chosen by the WHO, is the concept of adding to seasonal vaccine to make influenza vaccination broader or even universal.*

2.3.3 Approaches toward a Universal Influenza Vaccine

Hemagglutinin is the main target of immune response mediated by antibodies (Shaw et al., 2013). It is very important to stress the difference between HA head and HA stalk though. The head domain of HA, where most neutralizing antibodies bind, possesses high plasticity (Heaton et al., 2013). A very effective mechanism of immune-response evasion works through antigenic drift that takes place in the head domain which retains function in spite of frequent and various mutations. The stalk region, in contrast, is highly conserved (Shaw et al., 2013; Zhang et al., 2014).

In addition to hemagglutinin, M protein and NP protein have been used in approaches toward a Universal Influenza Vaccine (M. Zheng et al., 2014). In contrast to the conserved influenza matrix protein and nucleoprotein epitopes, which provide only weak protection in human challenge studies (Florian Krammer et al., 2015), the conserved stalk portion of HA is a much more potent candidate for a universal vaccine (Florian Krammer et al., 2015; Steel et al., 2010; Zhang et al., 2014). In particular, the highly conserved long alpha helix (LAH) of the stalk domain spanning amino acid 76-130 induces broadly reactive and protective antibody responses (Hauck et al., n.d.; Wang et al., 2010).

One obstacle when intending to use HA stalk as vaccine target is the observation that most antibodies produced in an influenza infection are directed against HA head. The strains encountered during an individual's lifetime play an important role in antibody production and new antibodies tend to be directed against the same epitopes, a phenomenon described as "original antigenic sin" (Vatti et al., 2017). To bypass this phenomenon new approaches of HA presentation are under development to elicit antibodies against the stalk and not the head after vaccination. Here, headless HA and

HA with various, very diverse heads are being used (F. Krammer et al., 2013; Neu et al., 2016; Steel et al., 2010). Since the three-dimensional structure plays an important role in epitope recognition new possibilities of presenting headless HAs needed to be developed as well. One possible way is using virus like particles (VLP) as carriers (Blokhina et al., 2013; Peyret et al., 2015). The vaccine that was used in this work was designed as part of a Framework 7 European Union program and used iQur's Tandem Core™ technology of virus like particles as carriers for different antigens ("Flutcore project outline," n.d., "Flutcore tandem core," n.d.; Liu et al., 2016).

2.4 Next generation sequencing

Sequencing means determining the order of nucleic acids making up an RNA or DNA sequence (Heather et al., 2016). Sanger sequencing, that used to be the predominant method of DNA decoding, is being more and more replaced by next generation sequencing (Heather et al., 2016). The term is being used for sequencing machines that can decode longer sequences in shorter a time. One example and the platform used in our work is the IonTorrent sequencer. IonTorrent uses instead of luminescently or fluorescently labeled dNTPs the fact that in dNTP incorporation into a newly synthesized strand a proton is being released causing a change in pH. In flooding the sample with each dNTP individually and registering the change in pH, it is possible to determine the order of nucleic acids in a sequence (Bragg et al., 2013; Quail et al., 2012; Rothberg et al., 2011).

At the time this work was performed IonTorrent was able to sequence 400 bp sequences with around 5 million reads per run taking around 5 to 7 hours per run ("Ion PGM™ System Specifications," n.d.). By now, according to various sources (Greenleaf et al., 2014; Heather et al., 2016) market leader is Illumina. Illumina uses fluorescently labeled nucleotides for paired-end sequencing (Heather et al., 2016). Various new approaches have been developed or are in the pipeline (Heather et al., 2016).

Another aspect of sequencing is the follow up, including data processing and analysis. In addition to more precise sequencing techniques new ways of adding to correctness of data have been developed. Among them is the method of barcoding (Figure 2). These barcodes can be used to label the DNA or RNA of samples and allow this way for pooling samples together. Sequencing more samples on the same chip leads to increased comparability between samples. More complex and with a key role in error correction, barcoding can be used as unique identifiers (UID) labeling RNA or DNA molecules.

The UID consists of a certain number of random nucleotides attached to the beginning or the end and in some cases to both sides of a sequence. After amplification and sequencing, sequences originating from the same RNA can be traced back and used for error correction (Kinde et al., 2011; Vollmers et al., 2013).

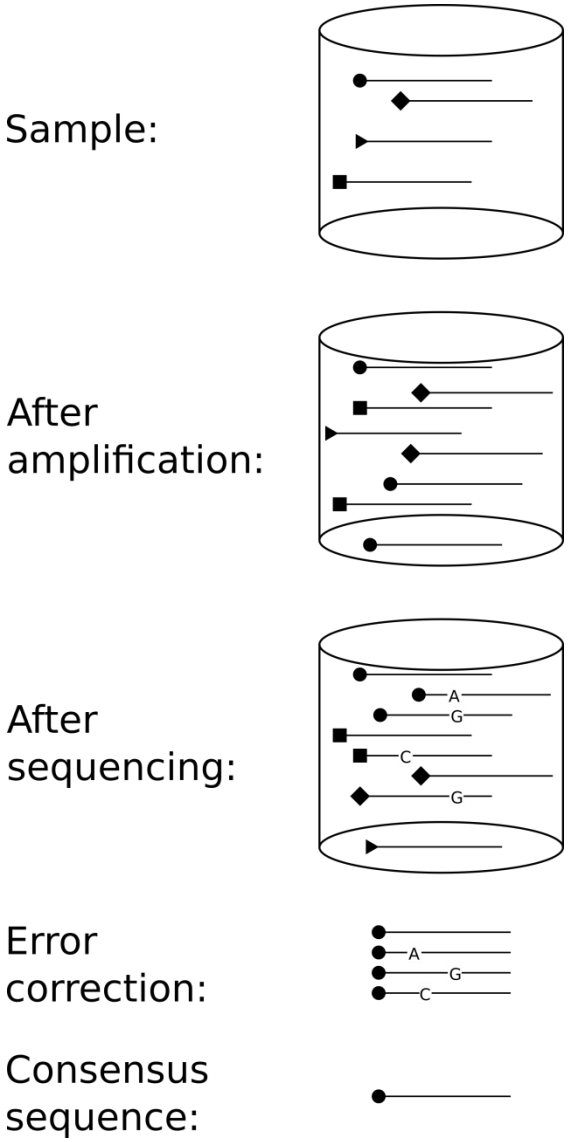

Figure 2: Barcoding principle. *Through barcoding (14 random nucleotides attached to the beginning of sequence, here represented as different symbols) each RNA molecule in the sample is labeled uniquely. After amplification different RNAs are present in different copy numbers. During amplification and sequencing errors are introduced into sequences. In grouping the same UID together (here schematically put as a circle), it is possible to build a consensus sequence that resembles the original sequence with high fidelity.*

Next generation sequencing produces large amounts of data that need to be analyzed effectively to make full use of its potential. Various platforms have been developed and for specific purposes new data processing approaches are being generated.

2.5 Aim

With new, broadly protective influenza vaccines on the horizon, questions regarding not only efficacy but safety as well arise. Will the presence of antibodies targeting conserved regions in a wider population trigger the creation of escape mutations? The aim of this work was to gauge the changes in virus population after vaccination targeting a specific conserved region. Furthermore, existence of escape mutations was to be investigated. So this work, within its boundaries of being performed in mice and with one specific vaccine construct, aims at being a proof of concept that a broadly protective vaccine against influenza targeting the conserved long alpha helix region of HA can be a safe strategy in combatting influenza.

3 Materials and Methods

3.1 Materials

3.1.1 Animals

- BALB/c mice (Harlan Laboratories, Inc., Horst, The Netherlands)

3.1.2 Cells

- Madin-Darby canine kidney cells (American Type Culture Collection)

3.1.3 Viruses

- pH1N1(A/Luxembourg/46/2009)

3.1.4 Solutions, chemicals, reagents

- Virus growth medium: 500 ml EMEM with Glutamine
  16.5 ml 7.5% BSA
  12.5 ml Hepes
  5 ml Pen-Strep 100x
  500 µl TPCK/ Trypsin
- TPCK (Sigma-Aldrich, Diegem, Belgium)
- Agencourt® AMPure® XP beads (Beckman Coulter, Suarlée, Belgium)
- MF59/Addavax® (Invivogen, Toulouse, France)
- DTT (ThermoFisher Scientific)
- Q5® High GC Enhancer (NewEngland BioLabs, Ipswich, Massachusetts)
- MDCK medium: 500 ml EMEM with Glutamine
  50 ml FBS
  5 ml Pen-Strep
  12.5 ml HEPES

3.1.5 Enzymes

- Superscript® IV (ThermoFisher Scientific)
- Phusion® (NewEngland BioLabs®, Ipswich, Massachusetts)
- Q5® (NewEngland BioLabs®, Ipswich, Massachusetts)
- RNAse OUT™ (ThermoFisher Scientific)

3.1.6 Commercial kits

- QIAamp® Viral RNA Mini kit (Qiagen, Hilden, Germany)
- Ion PGM™ Template OT2 400 kit (ThermoFisher Scientific)
- Ion PGM™ Sequencing 400 kit (ThermoFisher Scientific)
- Ion 316™ and 318™ chip kits v2 (ThermoFisher Scientific)

3.1.7 Buffers

- Superscript® IV First Strand Buffer (ThermoFisher Scientific)
- Phusion® Buffer (NewEngland BioLabs®, Ipswich, Massachusetts)
- Q5® Reaction Buffer Pack (NewEngland BioLabs®, Ipswich, Massachusetts)

3.1.8 Primers

- Forward primer: 5’ CACAGTTCACAGCAGTAGGTAAAGA 3’
- Reverse primer: 5’ ATTGCCCCCAGGGAGACTAC 3’
- MID07: TTCGTGATTC
- MID11: TCCTCGAATC
- MID19: TTAGTCGGAC
- MID30: CGAGGTTATC
- A Adapter: 5’ CCATCTCATCCCTGCGTGTCT 3’
- P1 Adapter: 5’ CCTCTCTATGGGCAGTCGGT 3’

3.1.9 Instruments

- Agilent Bioanalyzer 2100 (Agilent Genomics, Santa Clara, California)
- IonTorrent PGM™ (ThermoFisher Scientific)
- Ion OneTouch™ 2 System (ThermoFisher Scientific)
- Qubit® 2.0 Fluorometer (ThermoFisher Scientific)
- IonTorrent ES™ (ThermoFisher Scientific)
- NanoDrop™ (Thermos Scientific™)
- TissueLyser II (Qiagen, Hilden, Germany)

- Thermocycler UNO96 (VWR®)
- PURELAB® flex 3 and 4 Water Purification Systems (ELGA LabWater)

3.1.10 Software

- Pymol (The PyMOL Molecular Graphics System, Version 1.7, Schrödinger, LLC.)
- ID-50 5.0 program (http://www.ncbi.nlm.nih.gov/CBBresearch/Spouge/html_ncbi/html/index/software.html#1)
- BioEdit V7.2.5 (Hall, T.A. 1999. BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT. Nucl. Acids. Symp. Ser. 41:95-98)
- GraphPad Prism 5 (San Diego, CA, USA)
- Biopython V 1.66 (http://biopython.org/)
- Python V 3.5 (https://www.python.org/)
- MAFFT V 7.273 (http://mafft.cbrc.jp/alignment/server/)
- FASTX-Toolkit V 0.0.13 (http://hannonlab.cshl.edu/fastx_toolkit/)
- Geneious V R8.1 (http://www.geneious.com, Kearse et al., 2012)
- RStudio V 1.0.153 (RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/.)
- Mendeley V 1.17.10 (https://www.mendeley.com/reference-management/reference-manager)

3.2 Methods

3.2.1 Mouse work

Animals were housed in an ASL3 animal facility. Feeding and cleaning of cages was performed following a regular scheme. Weighing before challenge was done once a week and after challenge every day to monitor weight loss and to determine end point of the experiment for individual mice and their sacrifice time.

All mouse experiments were performed in accordance with protocols approved by the Animal Welfare Structure of Luxembourg Institute of Health and by the Minister of Agriculture, Viticulture and the Consumer Protection of the Grand Duchy of Luxembourg (Ref. LNSI-2014-02) (Hauck et al., n.d.).

3.2.2 Virus culture

MDCK cells were cultured to be used in lung titer determination. To this end trypsin and medium were warmed in a water bath and confluency of cells was checked under a microscope. The old medium was removed with a pipette and 1 ml trypsin was added to remove cells from surface of the flask. The flask was then put in a 37°C incubator and left there for 5 to 10 minutes depending on the cells still attached to the surface, which was checked again by microscope. 9 ml of fresh medium were added to the flask and cells now resuspended in medium and trypsin were transferred to a Falcon tube and centrifuged for 5 minutes at 1200 rpm and room temperature. The supernatant was removed and cells resuspended in 1 ml of fresh medium. For a dilution of 1:6 another 5 ml of medium were added and then 1ml of suspended cells transferred to a new flask with 20 ml medium.

3.2.3 Virus challenge

Eight week old BALB/c mice received three times immunization with LAH-HBc chimeric proteins containing long alpha helix of pH1N1 (A/Luxembourg/46/2009, pH1N1) and MF59 (Invivogen, Toulouse, France) in two weeks intervals. Mock group mice received adjuvant only. Two weeks after the final immunization, mice were challenged with 5 x 50% Mouse Lethal Dose of pH1N1 intranasally, after anesthesia with isoflurane (Lu et al., 2017). Mice were sacrificed 5 days post infection (n=3 for mock, n=6 for immunized) or 7 days post infection (n=2 for mock, n=3 for immunized) for organ removal (Hauck et al., n.d.) (Figure 3).

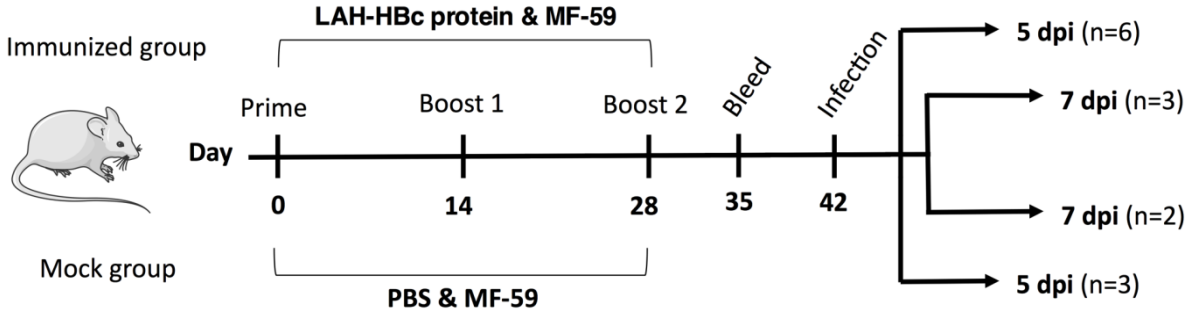

Figure 3: Scheme of mouse immunization and virus challenge. *Two groups ("immunized" and "mock") were immunized days 0, 14 and 28. Two weeks later animals were infected and sacrificed 5/ 7 days post infection (Hauck et al., n.d.).*

3.2.4 Organ extraction

Mice were sacrificed by cervical dislocation and directly prepared for organ removal by spraying them with ethanol. A pair of scissors was used to separate the skin from the peritoneal wall. At the sternum the ribcage was cut open until the neck. Cutting off the ribcage to the sides of the incision was followed by carefully lifting lungs and heart together with forceps out of the chest cavity. Lungs were disconnected from heart and remaining tissue. The lungs were put into a 2 ml safe-lock tube with a shredding bead and 0.9 ml VGM. Lungs were not stored but directly processed for RNA extraction and virus titer determination (Figure 4).

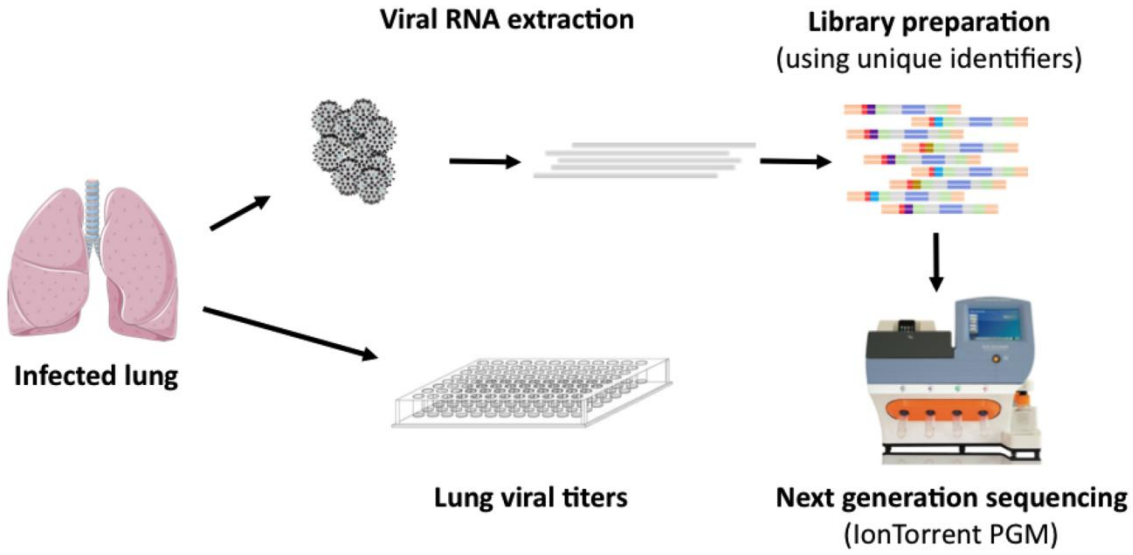

Figure 4: Assays performed on lung samples from immunized and mock immunized mice. *After extraction lungs were used for lung viral titers and viral RNA extraction. Viral RNA was sequenced on an IonTorrent PGM platform (Hauck et al., n.d.).*

3.2.5 Lung titer

Explanted lungs (n=14) were homogenized in 900 µl virus growth medium for 12 minutes at 25 Hz and centrifuged for 10 minutes at 11,000 rpm (Theisen et al., 2012). TCID50 in the supernatant was determined on Madin Darby canine kidney cells. Wildtype pH1N1/09 virus was cultured on MDCK cells in serum free growth medium that contained 2mg/ml L-1-tosylamido-2-phenylethylchloromethylketone trypsin. 50% TCID determinations of virus were performed on MDCK cells by incubating them in

quadruplicates for 20 hours with 8-fold serial dilutions of virus-containing supernatant at 37°C and 5% CO2, and were calculated by the ID-50 5.0 program (Hauck et al., n.d.).

3.2.6 RNA extraction

RNA was extracted from wildtype virus and from the supernatant of the homogenized lungs using the QIAamp® Viral RNA Mini kit. First, lung tissue was homogenized in a TissueLyser at 25 Hz for 12 minutes. Afterwards tubes were centrifuged at 11,000 rpm for 10 minutes and supernatant was pipetted off for RNA extraction. The following steps were performed according to the manufacturer's protocol. 560 µl AVL Buffer (included in kit) containing carrier RNA were pipetted into a 1.5 ml tube and heated at 80°C for 5 minutes to dissolve any precipitates. 140 µl of sample were added to the buffer and pulse-vortexed for 15 seconds. The mixture was left to incubate at room temperature for 10 minutes and afterwards briefly centrifuged to remove any droplets from lid. 560 µl of 100% ethanol were added, mixed by pulse-vortexing for 15 seconds and again briefly centrifuged. Half of this mixture (630 µl) was transferred to a QIAamp® mini column (included in kit) that sat on a 2 ml collection tube and centrifuged at 8,000 rpm for 1 minute. The 630 µl still left of the sample mixture were likewise transferred to the column and centrifuged. The filtrate was discarded and the column placed on a new collection tube. Afterwards, 500 µl of AW1 Buffer (also included in kit) were added and the column with collection tube centrifuged at 8,000 rpm for 1 minute. The filtrate was again discarded and the column placed on a new collection tube. This time 500 µl of buffer AW2 (included in kit) were added and the centrifuge set to 14,000 rpm for 3 minutes. The filtrate was discarded, the column placed on a 1.5 ml tube and 60 µl AVE buffer as final eluent added. Final centrifugation was done at 8,000 rpm for 1 minute ("QIAamp® Viral RNA Mini Handbook," n.d.). The quality of extracted RNA was analyzed employing an Agilent Bioanalyzer 2100 RNA kit.

RNA extraction and handling of virus were performed under Biosafety level 3 conditions.

3.2.7 Library preparation

For sequencing on the IonTorrent platform samples needed to be prepared in a specific way. The libraries for sequencing were prepared using an adapted protocol from Bürckert et al. (paper under preparation (Bürckert, n.d.)) and (Kinde et al., 2011; Loman et al., 2012; Vollmers et al., 2013).

Reverse transcription was performed on 100- 200 ng virus RNA:

1-10 µl of template

1 µl dNTPs

1 µl forward primer

Adding water to reach 13 µl of mixture in total

Cycler: 5’ 65°C, incubation on ice for 1 min

Adding:

4 µl FirstStrand Buffer

1 µl DTT

1 µl RNase OUT™

1 µl Superscript® IV

Cycler: 10’ 55°C, 10° 80°C

The primer for reverse transcription was linked to a unique identifier (UID) consisting of 14 random nucleotides which enables the recognition of every original mRNA strand after amplification (Figure 5). The primer was connected to four different mouse identifiers (MID) to allow pooling samples from different mice on a single sequencing chip. Furthermore, the primer contained a short sequence of IonTorrent A-Adaptor. After reverse transcription the second strand synthesis was performed on half the outcome of reverse transcription:

5 µl Phusion® Buffer

1 µl dNTPs

1 µl reverse primer

7 µl water

1 µl Phusion®

10 µl template

Cycler: 2’ 98°C, 2’ 50°C, 10’ 72°C

The reverse primer was connected to a short sequence of IonTorrent P1-Adaptor. After second strand synthesis and before amplification, the samples were purified twice with Agencourt® AMPure® XP beads at a ratio of 1:1.

For the final amplification step of the library Q5® enzyme in combination with a primer mix of A-Adaptor and P1-Adaptor was used:

5 µl Q5® Buffer

5 µl GC enhancer

1 µl dNTPs

0.5 µl primer mix

3 µl water

0.5 µl Q5®

10 µl of template

Cycler: 5’ 98°, 20x (10’’ 98°, 20’’ 65°, 30’’ 72°), 2’ 72°

The finished library was purified once with Agencourt® AMPure® XPbeads before being analyzed for both quality and quantity on Agilent Genomic’s Bioanalyzer 2100 before deep sequencing (Hauck et al., n.d.).

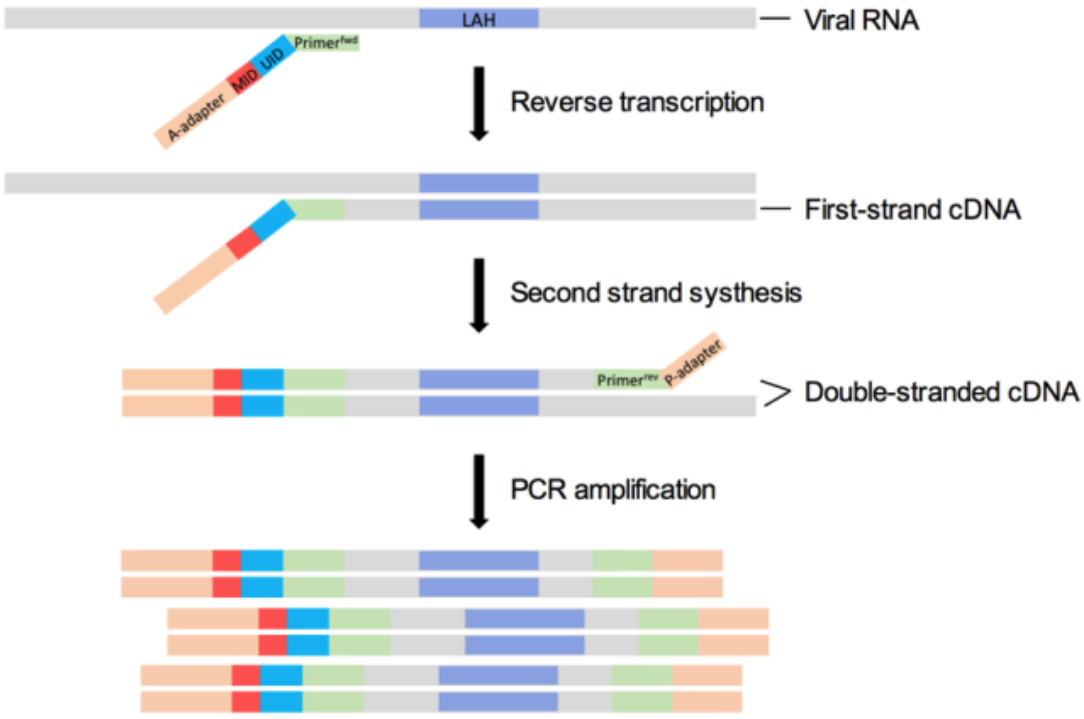

Figure 5: UID method in library preparation. *Shown is the UID method as applied in library preparation (Hauck et al., n.d.).*

3.2.8 Sequencing preparation

Libraries of two immunized and one mock mouse were pooled to be sequenced on the same chip. Wildtype virus libraries were done in triplicates and pooled together (Hauck et al., n.d.). Pooling was done to minimize discrepancy between samples because of differences in loading or other sequencing steps and in a way that the same concentration of RNA from each sample would be used. Template preparation was done using the Ion PGM™ Template OT2 400 kit, according to manufacturer's instructions, using the Ion OneTouch™ 2 System, Qubit® 2.0 Fluorometer and IonTorrent ES™ for enrichment.

3.2.9 High-throughput next generation sequencing

For sequencing the Ion PGM™ Sequencing 400 kit and Ion 316/318 chip kits v2 were used.

For all steps related to IonTorrent or preparation for IonTorrent sequencing, protocols were followed as indicated by LifeTech (Paisley, UK).

3.2.10 Data processing

An in house pipeline was built to process and analyze IonTorrent data that we obtained (Figure 6).

A few preparatory steps were needed before running the main part of the pipeline ("variants.py"). In order to normalize for differences between sequencing runs, similar to the principle of quantile normalization (Bullard et al., 2010; Rapaport et al., 2013), we compared the percentage of reads with a minimum coverage of three (three was chosen because consensus building requires at least three copies) for every chip. A minimum UID copy number for a sequence to be included into data analysis per chip was calculated, so that the same percentage of reads (33%) from every chip would be used. The threshold of 33% was determined by checking for every chip the percentage of reads with a copy number greater or equal to 3 and then selecting the lowest percentage of all chips. This task was performed by Python script "distribution_threshold.py" (see Annex (7.3.1)).

To overcome the heterogeneity in copy numbers among different UIDs, we calculated how many sequences needed the same nucleotide at the given position, for the nucleotide to be the consensus character for this position. For example, to build a reliable consensus sequence from only 3 copies for a given UID, a minimum of 2 need

to be the same to define a reliable nucleotide read. The thresholds were calculated individually for every coverage using the cumulative binomial distribution, with q=0.015625, and α=1-q. Cumulative binomial distribution was used to create a list that assigns the copy number needed to always reach the same probability to every possible UID copy number. This list was used in the main script ("variants.py") for consensus building.

In the main script ("variants.py") (see 7.3.2) trimmed BAM files exported from IonTorrent platform were imported. The pipeline was used to filter out poor quality reads and to correct for homopolymer errors. In a first quality filtering step, reads with 20% less than the expected length were filtered out and only reads which had more than 95% of nucleotide positions with a minimum quality score of 20 were kept for further analysis. After splitting the reads coming from the different samples according to their MIDs, they were grouped into reads originally coming from the same mRNA molecule by using the UIDs. Errors in poly-A-sequences were corrected to forestall the most commonly reported errors in IonTorrent sequencing (Bragg et al., 2013). Since the three poly-A-sites within our sequence were composed of repeats of 6 nucleotides, we uniformed all the sequences by correcting those with 5 or more than 6 repeating nucleotides at these positions to ones with 6 nucleotides. Sequences were cut to include only the epitope that the vaccine construct targets. Within each selected UID family we aligned reads using MAFFT and built a consensus sequence using Biopython (gap_consensus) with the thresholds from our list obtained with cumulative binomial distribution (Hauck et al., n.d.).

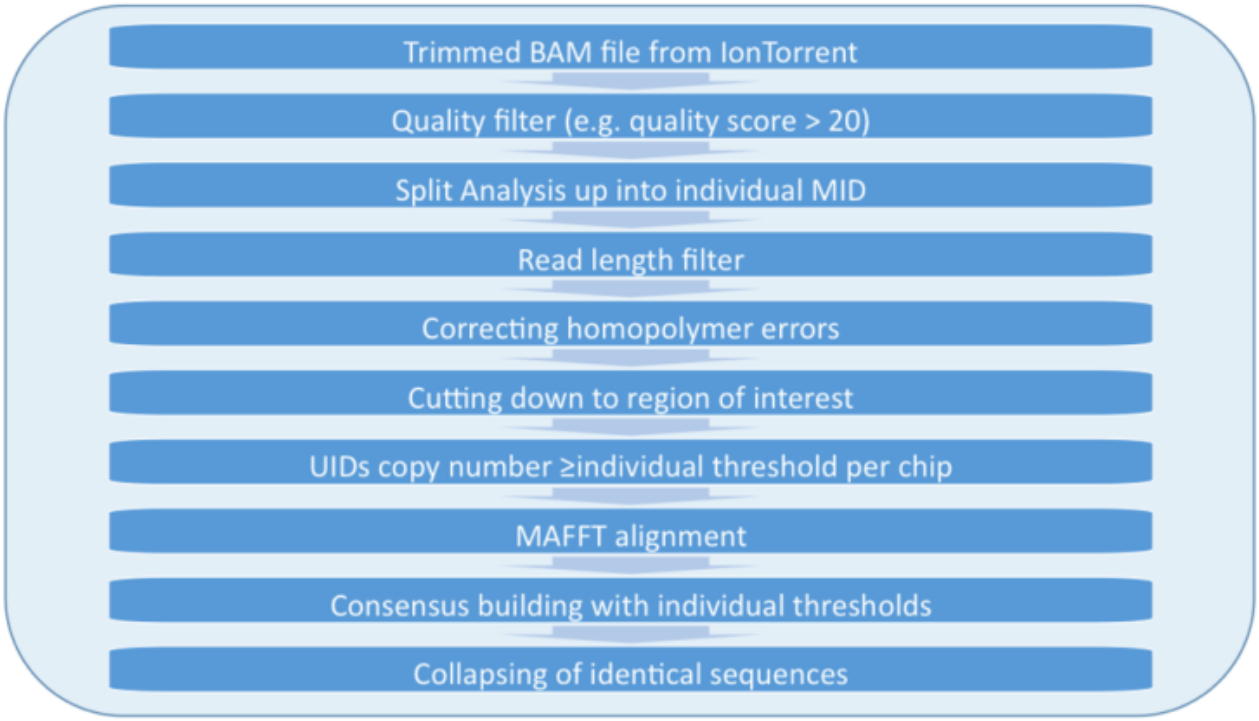

Figure 6: Data processing steps. *The different steps our data processing pipeline was comprised of are listed (Hauck et al., n.d.).*

The main script ("variants.py") was called individually for every pool. The variables to be set included the input file (BAM), a prefix to recognize data from the specific input file (this prefix was added to all files generated in this run of the script), the forward and backward "primer" used to recognize the region of interest, the UID threshold (all reads with a coverage below threshold were discarded) calculated in "distribution_thresholds.py". Furthermore it was possible to decide whether quality correction or poly-A-site correction should be performed (these steps were used in all pools) and the list of how many copies of a certain UID needed to be the same to build a consensus sequence were given as tsv file. Some variables were set to a default value, like the minimum and maximum required sequence lengths (270 and 500, respectively), the minimum quality a base should have (20) and the percentage of bases within a sequence which should have the minimum quality (95%) (as described above).

For pipeline see Annex (7.3 Pipeline for data processing).

3.2.11 Data analysis

Geneious and BioEdit were used to convert the nucleotide sequences into amino acid sequences and to study mutations. The Shannon diversity index was calculated (Luo et

al., 2012) to determine diversity of quasispecies population. All sequences, including the ones having frameshifts or stop codons (1-2% of final sequences) were included in the diversity calculations. The heatmap (Figure 12) showing each diversified amino acid position on LAH epitope was generated using R program. Values for each mutation have first been normalized by determining the percentage of total consensus sequences they make up in the individual samples before a clustering analysis (Hauck et al., n.d.).

3.2.12 Statistical analysis

In GraphPad Prism 5 multiple t-tests, unpaired t-test, and one-way ANOVA followed by Tukey's as post-hoc test were used to determine statistical significance. P value less than 0.05 was considered as significant (Hauck et al., n.d.).

4 Results

4.1 Protection against influenza A virus challenge

Female BALB/c mice were challenged with H1N1 and weight was controlled every day. In previous experiments in our lab, performed by I-Na Lu and Sophie Farinelle, extensive protection could be shown against both strain specific challenge and various other strains, including even protection against challenges with strains from other IAV group (Figure 7)(Hauck et al., n.d.).

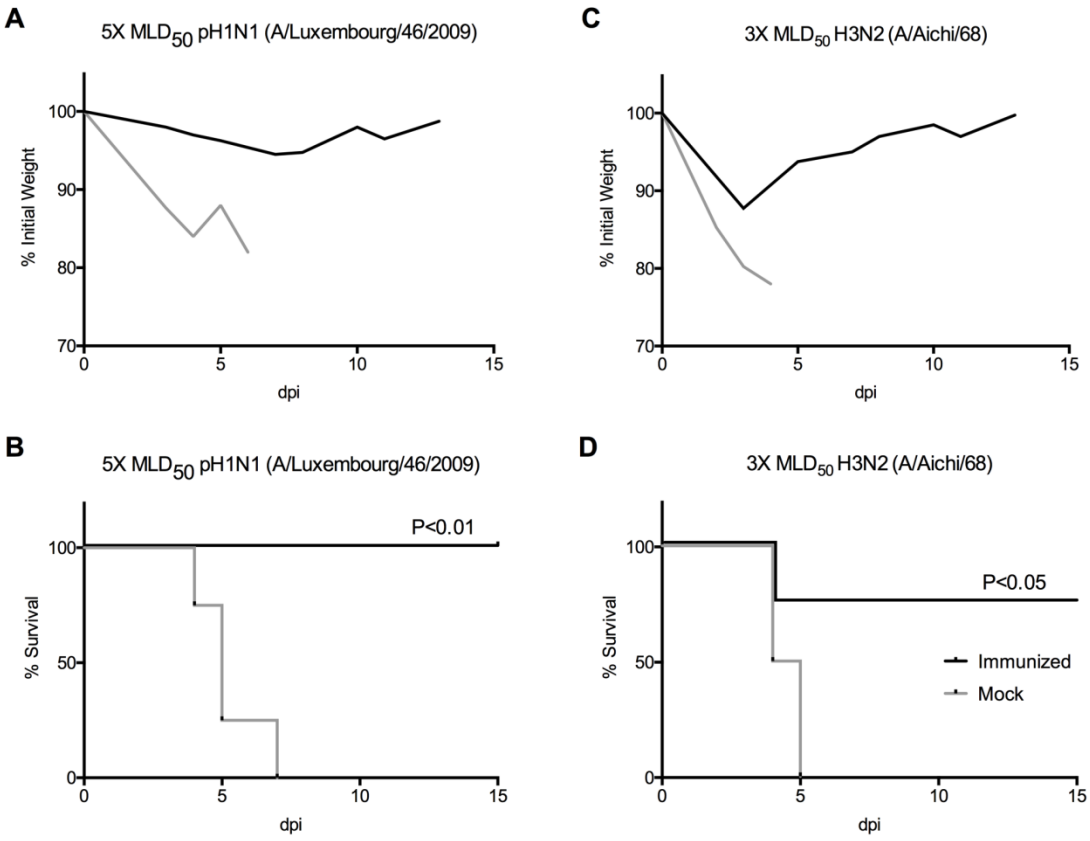

Figure 7: Protection against different IAV strains. *Protection conferred against lethal challenge of 5x MLD50 of either pH1N1 (A/Luxembourg/46/2009) or H3N2 (A/Aichi/68) virus. Mock mice were vaccinated with adjuvant only. Mice vaccinated with LAH-HBc chimeric proteins (A and B) showed 100% survival after pH1N1 challenge and 75% survival after H3N2 challenge (C and D), while the mock mice all died (Hauck et al., n.d.).*

4.2 Reduced lung virus titers in immunized mice

After immunization animals were challenged intranasally with 5 x MLD50 of influenza A virus and the viral load was measured in the lung 5 and 7 days post infection. A significant difference in virus lung titers emerged on 7 days post infection between immunized and mock animals (Figure 8) (Hauck et al., n.d.).

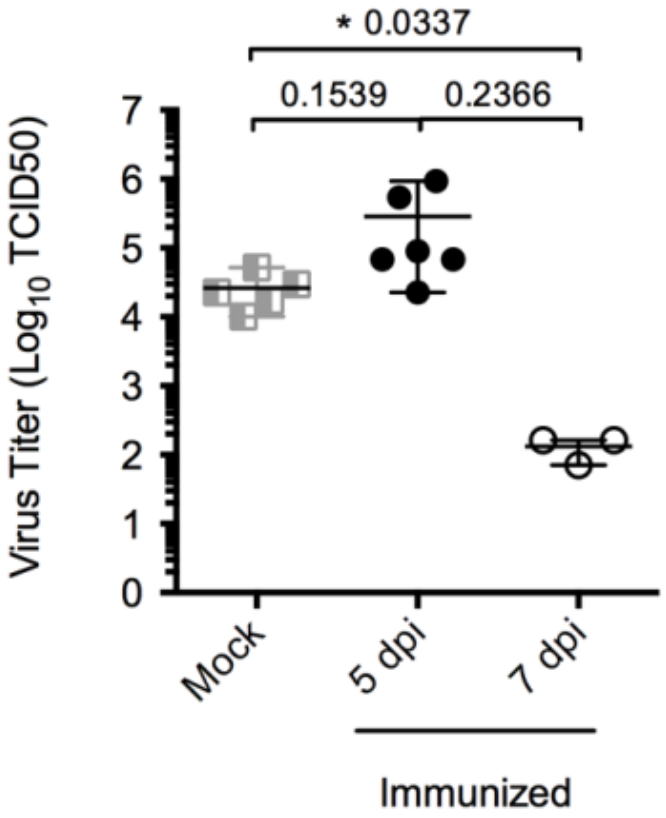

Figure 8: Lung titer. *Lung virus titers of mice 5 days post infection and 7 days post infection. *P < 0.05. Error bars represent standard error of the mean (SEM) (Hauck et al., n.d.).*

4.3 Presence of influenza RNA and sample input determination for libraries

Lungs were explanted after sacrifice and virus RNA was extracted. The extracted RNA was tested by PCR analysis by Aurélie Sausy to certify the presence of influenza virus RNA. This way, the presence of influenza RNA could be confirmed in all samples.

Extracted RNA was quantified on NanoDrop™ and the best amount of starting material had to be determined. In the end about 100 ng of wildtype virus RNA and between 100 ng and 1750 ng of mouse sample RNA were used to prepare at least three libraries for each sample. The library used for sequencing was chosen by library quality determined on Agilent's Bioanalyzer 2100 (Figure 9).

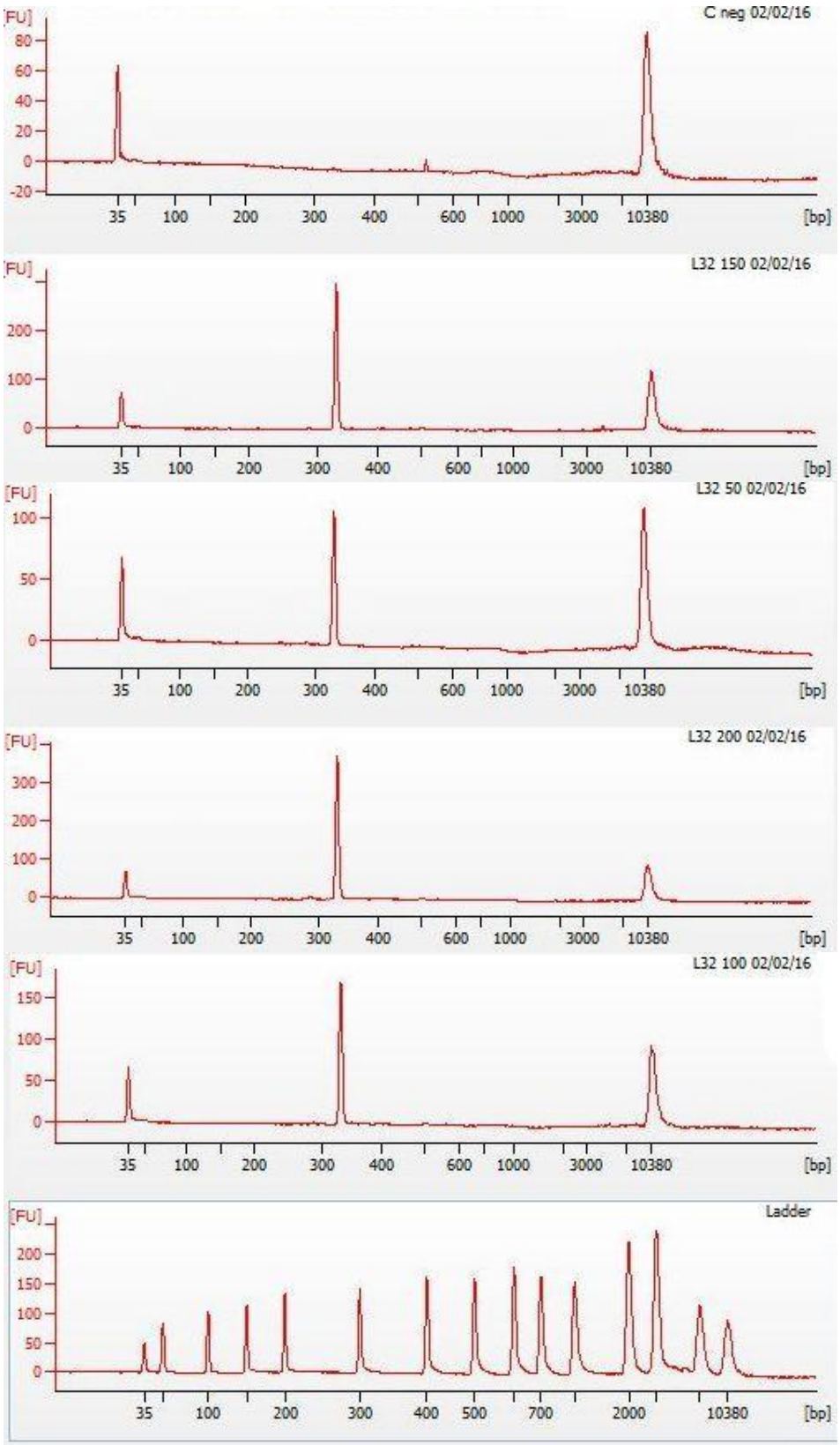

Figure 9: Quality control on Bioanalyzer. *On the example of lung 32 are shown the Bioanalyzer results after library preparation (C neg means negative Control).*

4.4 Sequencing quality

Chip loading influences the number of raw reads that can be achieved in sequencing (Figure 10).

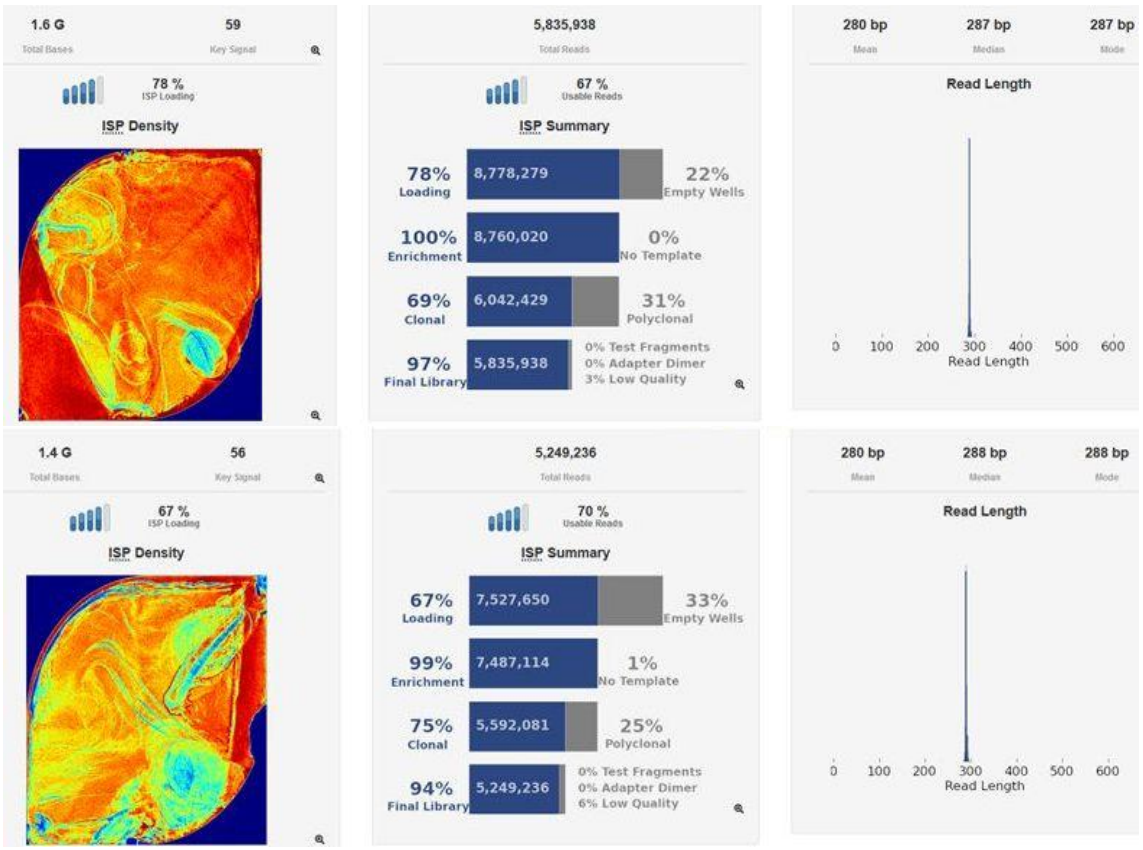

Figure 10: Chip loading. *Two examples of chip loading and raw read output (at the top Pool 22 and at the bottom Pool 28).*

For number of raw reads, reads for consensus building after clean-up, consensus sequences and number of unique consensus sequences see Table 1: Sample properties.

4.5 Decreased diversity of long alpha helix epitopes

The virus on a population level was further characterized on 5 dpi and 7 dpi using next generation sequencing. The overall diversity of viral quasispecies in the LAH domain in the different groups was calculated using the Shannon Diversity Index. The wildtype virus expanded in vitro on MDCK cells showed the highest variability within the tested region. On 5 dpi, viruses from the lung of immunized mice showed significantly less variability than those of the mock group. This was no longer true at 7 dpi. These results

demonstrate that in the animals the virus loses diversity and that this effect is more dramatic in immunized mice than in the control group. Inversely, when comparing the diversity of nucleotides with amino acids, the ratio was significantly increased on 5 dpi in the immunized mice compared to the wildtype virus and mock animals. Thus, in the immunized animals the virus has more synonymous mutations compared to the one from the wildtype virus and the mock group (Hauck et al., n.d.) (Figure 11).

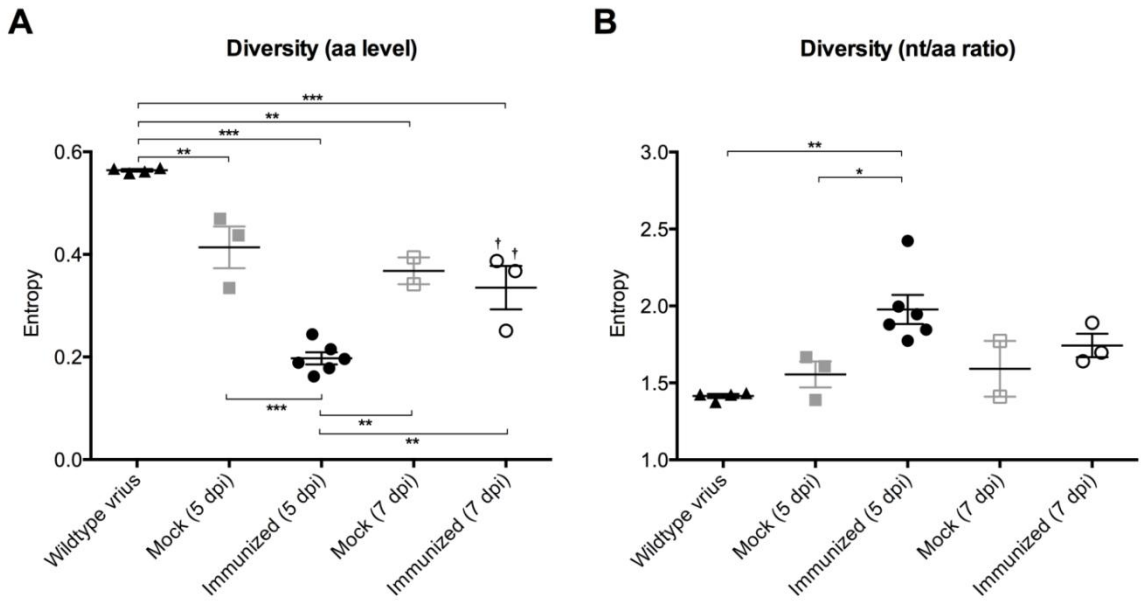

Figure 11: Levels of diversity in comparison between different groups. *Lung viruses of vaccinated mice showed a reduction in LAH sequence diversity. (A) Shannon diversity index of LAH amino acids. †: corresponding to the samples with the same labeling in Fig. 12B. (B) Nucleotide to amino acid entropy ratios of the Shannon diversity calculated for each sample. *P < 0.05, **P < 0.01, and ***P < 0.001. Error bars represent SEM (Hauck et al., n.d.).*

Sample type	Library input RNA (in ng)	Sacrifice dpi	Lung viral load (TCID50/ml)	Raw reads	UIDs	Reads for Consensus building	Consensus sequences	Unique Consensus sequences	Pool n°	MID	Sample ID
Virus stock (Cell line)	100	n.a.	n.a.	249062	28895	207057	19306	319	24	7	1.1
Virus stock (Cell line)	100	n.a.	n.a.	413227	62346	377514	46755	539	24	11	2.1
Virus stock (Cell line)	100	n.a.	n.a.	384512	51099	351825	38826	501	24	19	2.2
Virus stock (Cell line)	100	n.a.	n.a.	478384	75522	434946	51322	572	24	30	2.3
Lung (non-immunized)	275	7	1.17E+04	389408	26892	218449	8102	147	21	11	1
Lung (non-immunized)	1375	7	2.97E+04	968277	249635	418190	55685	537	22	19	2
Lung (non-immunized)	145	5	5.18E+04	520478	17578	468695	11755	298	26	7	35
Lung (non-immunized)	100	5	2.26E+04	729269	34345	697904	13131	199	28	30	36
Lung (non-immunized)	200	5	1.01E+04	504915	18283	434872	1852	69	30	30	37
Lung (immunized)	1750	7	1.62E+02	359130	12425	327526	5014	128	21	30	79
Lung (immunized)	640	7	1.62E+02	558652	13859	539592	9040	201	22	11	80
Lung (immunized)	600	7	7.12E+01	405625	18700	391768	14014	284	22	7	11
Lung (immunized)	110	5	8.97E+04	604422	15195	583742	10267	137	26	19	27
Lung (immunized)	100	5	6.82E+04	848061	95931	334904	21357	239	26	11	28
Lung (immunized)	100	5	6.82E+04	397063	27641	386600	23130	253	28	7	29
Lung (immunized)	100	5	5.40E+05	620031	40259	601034	31871	303	28	19	30
Lung (immunized)	100	5	2.26E+04	780964	8348	742711	3718	83	30	11	31
Lung (immunized)	100	5	9.36E+05	678494	19510	512667	7991	153	30	19	32

Table 1: Sample properties

4.6 Absence of escape mutants following vaccination

To understand the observed differences in quasispecies diversity, the missense mutations in the different groups were examined. In order to select for relevant mutations only, we focused on those that appeared in at least 3 of the 18 samples. With this approach 212 possible amino acid exchanges within the analyzed epitope, which covered all 55 positions were found. A heatmap was generated to compare the occurrence of these missense mutations between the groups (Figure 12). Hierarchical clustering based on these occurrence levels resulted in three major clusters, one composed only of wildtype virus samples, a second one composed only of samples from immunized mice and a third one containing all mock samples and two of the 7dpi samples from the immunized group. This observation could be further confirmed by principle component analysis (PCA) to the frequencies of mutants among different groups. The marked two 7dpi samples showed a similar level of diversity as in the mock mice. An overall gradual decline in mutated virus sequences can be observed when comparing the wildtype virus, mock and immunized samples, confirming again the differences in diversity observed between the groups (Hauck et al., n.d.).

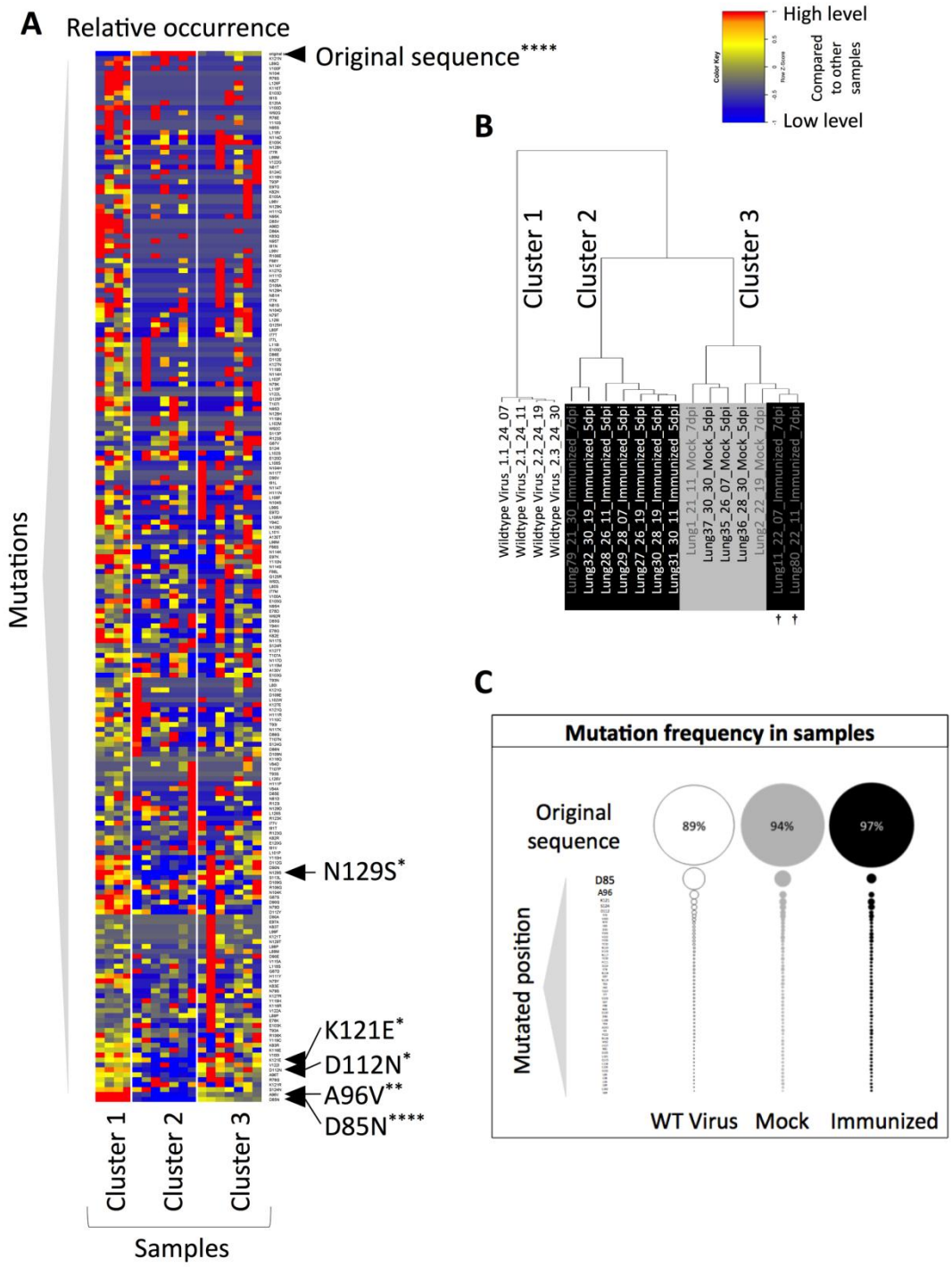

Figure 12: Constrained viral quasispecies evolution under immune pressure. *(A) Heatmap of missense mutations detected in at least three out of the eighteen samples analyzed. Values from each row (i.e. mutation) have been normalized to have a mean of 0 and SD of 1 and relative expression levels of the same mutation within the individual mice are represented by the color code indicated in the legend at the top right. (B) Hierarchical clustering analysis performed using complete linkage and Euclidean distance measurement. †: corresponding to the samples with the same labeling in Fig. 11A. (C) Mean sequence composition of samples per group. The sizes of the bubbles are proportional to the mean percentage sample consensus sequences of each group. *P < 0.05, **P < 0.01, and ****P < 0.0001. Error bars represent SEM (Hauck et al., n.d.).*

4.7 Identification of diversified positions on long alpha helix epitope

In order to extract the most significantly mutated positions of the analyzed epitope, the Shannon diversity index was calculated for every amino acid position across the LAH epitope (**Fehler! Ungültiger Eigenverweis auf Textmarke.**). This way, 11 different positions were identified to reach statistical significance when comparing the groups using a two-tailed student's t-test. At each of these positions, except for position 123, the immunized group showed a lower or similar aa diversity as compared to the mock mice. Amino acid positions 85 and 96 were particularly variable in the cultured virus, but this was lost in the in vivo virus populations. Variability in these two positions was the major contributor to the differences in diversity observed between the groups as confirmed by PCA analysis. Positions 121 and 124 seem to be generally more diverse but with little difference between groups. The PCA analysis showed very different diversities among in vitro and in vivo samples. However, the differences between Mock and Immunized mice were not clearly distinguishable by using the Shannon diversity index alone, indicating the need of more detailed analysis on the amino acid substitutions (Hauck et al., n.d.).

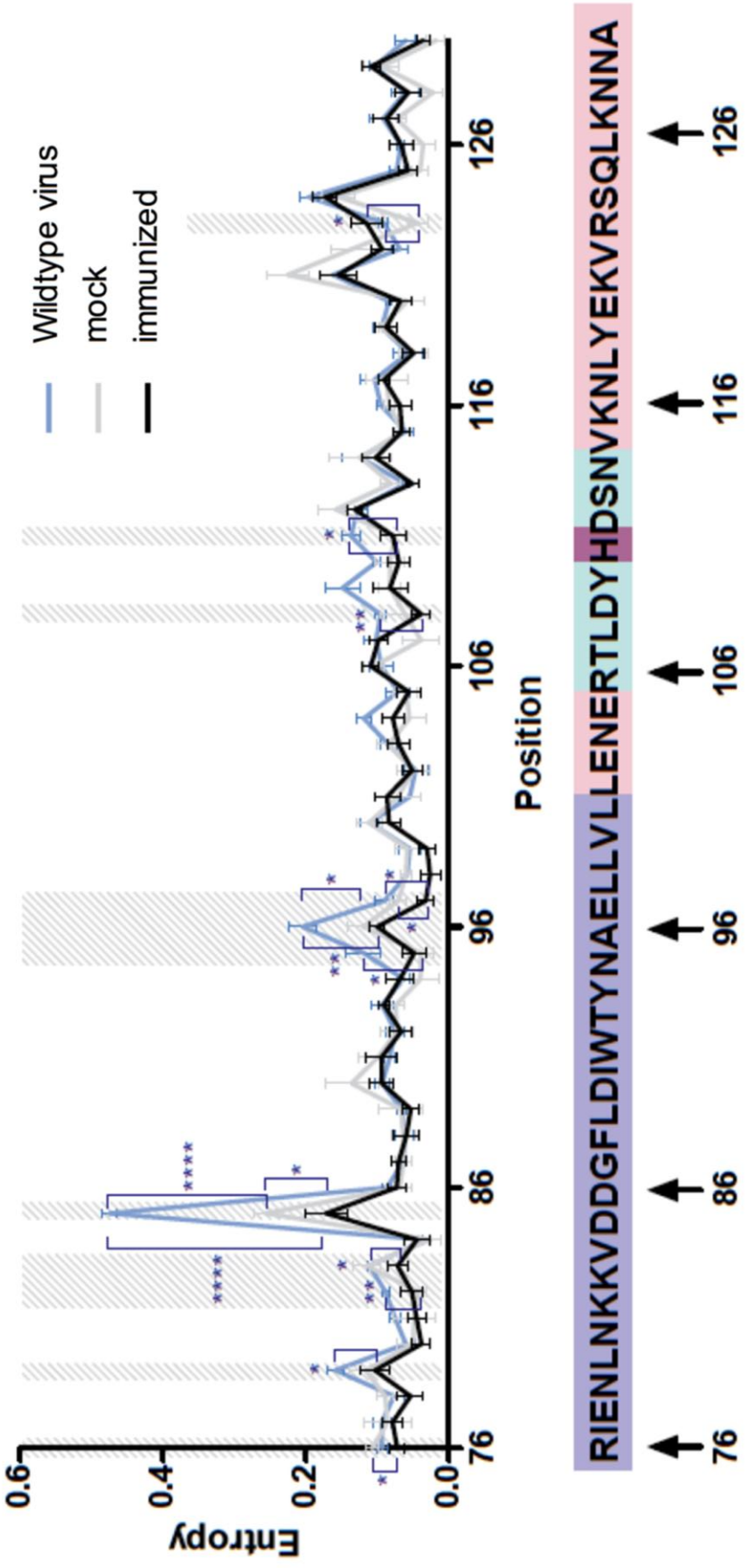

Figure 13: Identification of variable amino acid positions. *Entropy at individual positions. Significant differences of mutation frequencies among groups are highlighted (Hauck et al., n.d.).*

4.8 Analysis of amino acid substitutions

To identify the mutations that were selected by the vaccine-induced immune pressure, all the identified 212 amino acid exchanges were compared between the Immunized and the Mock groups. Five mutations were found with significantly different frequencies between the groups: D85N, A96V, D112N, K121E, and N129S (Figure 14). Each of these amino acids showed a significantly reduced frequency in the immunized animals. No mutant variant emerged that replaced the dominant viral sequence in any of the samples. There were, however, significant differences in the occurrence of the dominant viral sequence between groups (Figure 15), suggesting a constrained viral quasispecies evolution in the Immunized group (Hauck et al., n.d.).

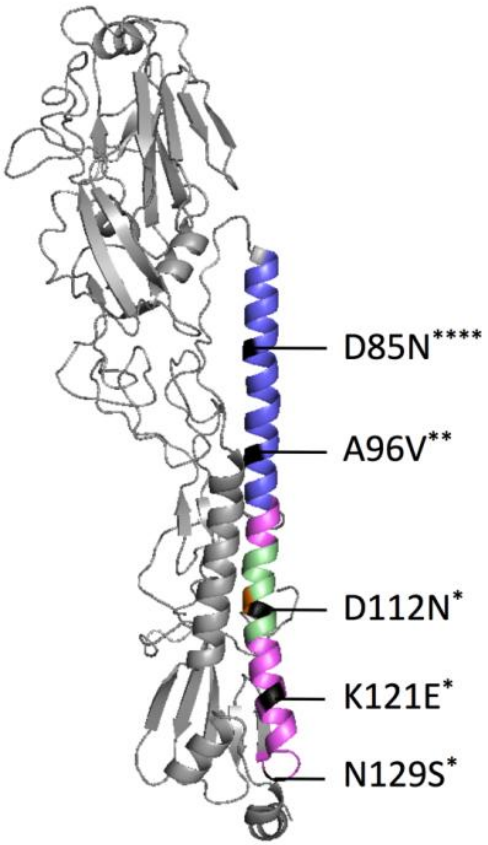

Figure 14: Crystal structure of hemagglutinin. *The crystal structure of H1 subtype (PDB: 3AL4) was used to map the positions of significant missense mutations on the HA monomer by PyMOL software. The colored region represents the LAH epitope. Slate blue denotes the heptad repeat region, pale green represents conformational change region with the H contributing to histidine-rich patch labeled in orange, and the remaining of the epitope is colored in pink (Sriwilaijaroen et al., 2012). The mutations denoted in black are mutations for which a statistically significant difference in occurrence was observed when comparing immunized with mock samples (D85N, A96V, D112N, K121E, and N129S) (Hauck et al., n.d.).*

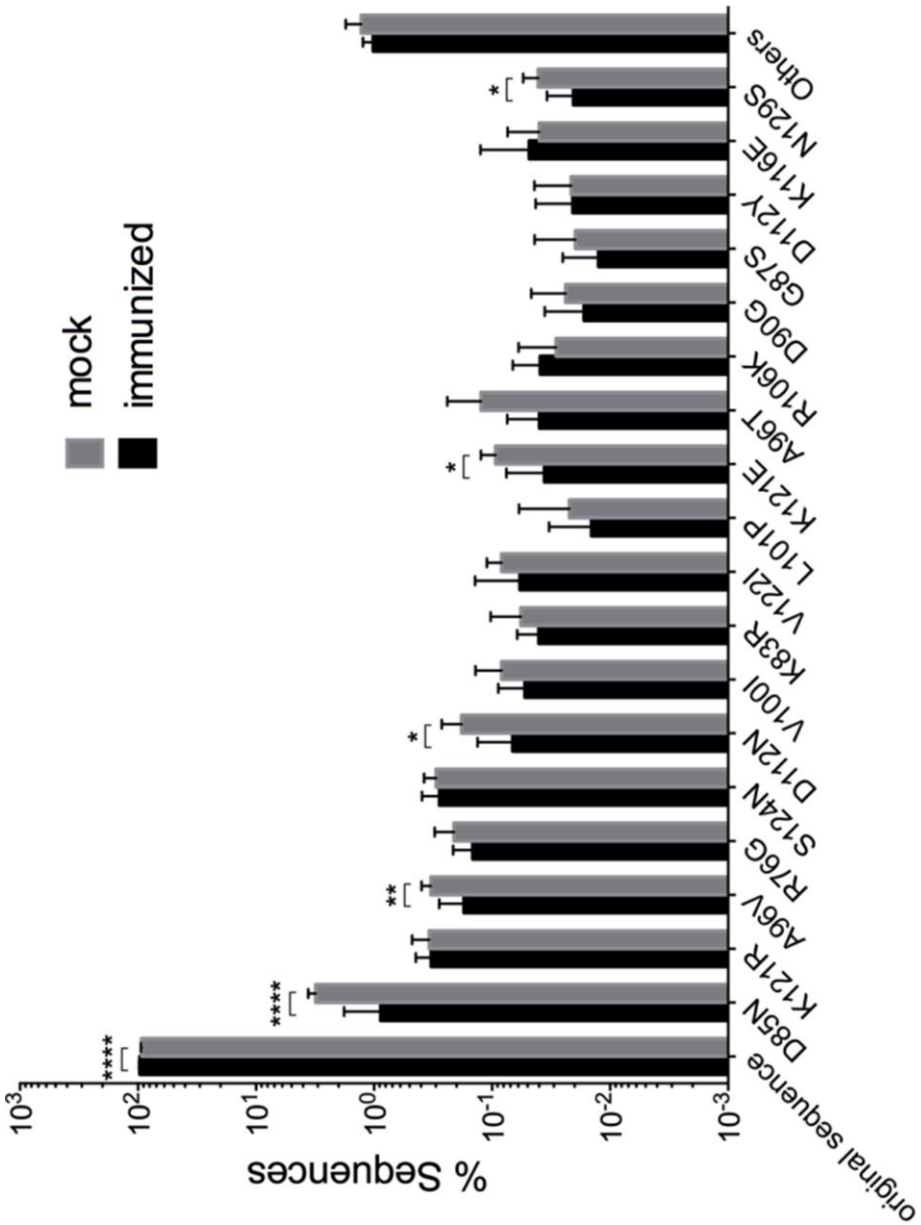

Figure 15: Frequencies of top 18 mutations. *Frequencies of top 18 mutations in Mock and Immunized groups.* $*P < 0.05$, $**P < 0.01$, *and* $****P < 0.0001$. *Error bars represent SEM (Hauck et al., n.d.).*

5 Discussion

In the present work, NGS was applied to directly follow the *in vivo* IAV quasispecies evolution in response to humoral immunity to long alpha helix, a conserved hemagglutinin stalk epitope. In agreement with previous studies conducted by Wang et al. and Zheng et al., our LAH targeting vaccine elicited significant seroconversion and protection against homologous and heterologous IAV strains in mice (Wang et al., 2010; D. Zheng et al., 2016). The virus quasispecies found after vaccination showed a significant reduction in amino acid variability in the LAH domain and no escape mutant emerged. These findings provide further evidence for LAH as a potential vaccine candidate and as proof of concept for a Universal Influenza Vaccine that can control influenza A virus without enhanced propensity for escape mutants within the investigated region (Hauck et al., n.d.).

5.1 Summary

Influenza is a serious threat to global health and economy. It is unlikely humankind could eliminate influenza but we can certainly strive to lower the burden and improve influenza preparedness. Going new ways in vaccination techniques, there is a need to know what our work might entail. Part of this is being aware and taking responsibility for how we influence the evolution of influenza. When putting pressure on a virus, in form of antiviral drugs or antibodies targeting the virus, evading mechanisms arise (Florian Krammer et al., 2013). Antiviral drugs like M2 channel blockers have let to a growing population of resistant viruses (World Health Organization, 2014). Seasonal influenza vaccines targeting the head region of HA lead to similar changes in population which the virus population can circumvent through antigenic drift and shift. It would be naïve to not consider similar evolution after vaccination against conserved regions. The question following out of this observation is whether new vaccination approaches targeting conserved regions might serve as the pressure that will give the incentive for the virus to escape. The hemagglutinin might be more conserved because of very specific functionality or conserved regions might be relatively stable because of little

need to change so far because conserved regions were not targeted frequently by the immune system. Already Chai et al. have shown that evading mechanisms are possible though (Chai et al., 2016). The evading mechanisms discovered so far were observed in *in vitro* studies and meant less fitness for the virus. It remains to be shown what the evolution *in vivo* and in the long run might be. The mouse model is a sound depiction (Frise et al., 2016) but an interesting further approach would be an in vivo study with a ferret model which allows for inter-individual infection cycles similar to humans (Frise et al., 2016). In a study like that it would be interesting and possible to find out which mutants might keep fitness to infect effectively. Furthermore, some discoveries as to the virus's reaction might only be made after vaccination against conserved regions becomes widely available and used.

5.2 Quasispecies and virus growth dynamics

The population dynamics of multiple IAV strains have been well documented in both humans and mice (Grenfell, 2004). Chai et al. used antibodies against conserved regions of HA to select for escape mutants and tested them *in vivo* (Chai et al., 2016). In contrast to our findings, Chai et al. could identify two possible escape mechanisms, although with reduced fitness for the escape mutations. Here, we used a murine infection model to investigate the changes of lung virus quasispecies under the pressure of antibodies targeting LAH. Unvaccinated animals failed to control IAV replication in the lungs and died at 7 dpi due to serious pulmonary damage. In response to the vaccination, viral kinetics followed a course similar to the one described in previous studies (Grenfell, 2004). Viral replication in the lungs became detectable at 3 dpi, peaked at 5 dpi, and began to decrease at 7 dpi. For this reason, we examined lung virus quasispecies at both peak (5 dpi) and reduced (7 dpi) viral replication (Hauck et al., n.d.).

We observed an increased proliferation of influenza virus at 5 dpi and a rapid viral clearance at 7 dpi in lungs of the immunized mice, while viremia persisted in the lungs of the mock animals at both time points (Figure 8). These results are in accordance with previous observations that a short but rapid proliferation of IAV in mice with a stronger immunity leads to enhanced activation of the acquired immune system, which ultimately results in fast viral clearance (Hernandez-Vargas et al., 2014; Pawelek et al., 2012). A well-known bottleneck effect (Bull et al., 2011; Domingo et al., 2012) that is defined by a reduction in viral diversity accompanied with escape mutants is often

observed in viral evolution studies under host immune selection. In this study, a similar bottleneck effect was also observed, which reduced amino acid diversity of virus quasispecies in the LAH domain. However, an unexpected preferential proliferation of viruses with non-mutated LAH in immunized mice was shown in the analysis. These findings demonstrate that single or few mutations on LAH do not allow influenza virus to survive or to become resistant, and the vaccine did not favor any expansion of escape mutants (Hauck et al., n.d.).

Apart from antibodies other reasons play into the way that the quasispecies changes. Certain characteristics of the individual mouse might influence viral clearance and severity of infection. Those factors were not controlled for in this study. In addition, in the present study design it was not possible to control the outcome since mice that survive would have cleared lungs with no RNA for sequencing.

RNA viruses have a high mutation rate which in influenza A in particular has been calculated to be -5x10^-4 to -8x10^-3 nucleotide substitutions per site per year (reviewed in (Wright et al., 2013)). Thus, we would expect to find escape mutations within one cycle of infection in mice, if escape mutations should arise. Of course, a longer observation time frame might lead to additional findings and might be needed to definitely rule out any appearance of fit escape mutants.

The Shannon diversity index has been used to follow quasispecies evolution (Gregori et al., 2016). Other possibilities of describing the evolution of quasispecies include various other indices (e.g. minimum mutation frequency or population nucleotide diversity) (Gregori et al., 2016). The methods of quasispecies' description remain limited as long as sequencing the whole genome of all virions present at the same time is not feasible (Gregori et al., 2016). This remains a field of further studies and might render new insights into quasispecies in the future.

5.3 Next generation sequencing and data processing

Next generation sequencing is a fast evolving field, applied for various and increasing numbers of studies. Likewise, NGS is on its way of becoming the gold standard for viral quasispecies observations (Barzon et al., 2011). Here, development of techniques and their respective evaluation is still in progress. This includes error rates and processing pipelines. IonTorrent, as one of different next generation sequencing approaches, holds some challenges but with careful error correction it was useful for this study. In contrast

to other NGS platforms IonTorrent is prone to indels but has a low propensity for substitution errors (Goodwin et al., 2016). Indel errors could be reliably removed by comparison to the IAV reference sequence. When deciding how to best build consensus sequences IonTorrent data have the advantage and disadvantage of specific errors in specific situations (Bragg et al., 2013). When errors are not random but occurring in the same positions consensus building has limitations. On the other hand knowing what errors to expect three potentially challenging regions with homopolymer A sequence could be identified and corrected specifically to facilitate an efficient consensus building. Since low levels of mutations were of primary interest, substitution errors were further reduced by applying a barcoding technique that allows tracing back of individual RNA strands (Kinde et al., 2011; Loman et al., 2012). This technique also allowed pooling of samples from different groups on a single sequencing chip to minimize differences caused by chip loading. Inter-chip normalization enabled the removal of any further factor that could influence reproducibility (Rapaport et al., 2013) (Hauck et al., n.d.).

An in-house pipeline was established so that meeting the exact requirements of this work would be possible. As a small drawback the pipeline was not validated on other data sets for cross checks. With multiple tests on smaller parts of our data between pipeline steps and careful building of the pipeline, we added to a reliable processing procedure.

As another drawback of using IonTorrent in the set-up of this work, sequencing length was limited to a few hundred base pairs which made monitoring of all mutations throughout the whole genome impossible. This study therefor focused on the target epitope. In this region of greatest interest due to the antibodies targeting the region, no escape mutations could be shown. It is however possible that escape mutations could occur in other regions than the targeted one, so for the future it would be desirable to sequence the whole influenza genome in order to control all possible mutation sites. Here, further studies are needed to confirm and extend our promising findings (Hauck et al., n.d.).

5.4 Mutations and variability

According to Brooke et al. a large number of virions in an infected host are not assembled completely (less than the normal 8 RNA strands) (Brooke et al., 2013). In our samples 1-2% of final sequences had stop codons or frameshift mutations that

would make a proper assembly of HA protein impossible. These variants were included in quasispecies diversity calculation because they could be seen as "attempts of escape" but not in showing mutants since virions without a functioning HA would not be able to infect a new cell and thus pose no threat of replicating and making up large portions of the quasispecies as escape mutants (Hauck et al., n.d.).

With the set-up of this work, previous discoveries could be confirmed. For example, a mutation S124N was identified that was present in all samples analyzed with relatively high prevalence (0.37%-0.41% in "wildtype virus"; 0.16%-0.35% in "Mock"; and 0.04%-0.36% in "Immunized"). This mutation was also sporadically found in humans during the 2009-2010 season and became dominant during the 2011/2012 influenza season (Lee et al., 2015), which further supports the robustness of our approach (Hauck et al., n.d.).

When comparing the Immunized and the Mock group, five mutations showed a significantly different prevalence. Each of these mutations was less frequent in the immunized than in the mock animals. These mutations were scattered across three different regions (Sriwilaijaroen et al., 2012) of the LAH domain: D85N and A96V were found in the heptad repeat region, D112N in the conformational change region, and K121E and N129S in a non-functional region. The latter two mutations were also situated on a previously identified CD4 T-cell epitope of BALB/c mice (Lu et al., 2017). Whether or not the immunization had an impact on the reduction of these mutations will require further experiments for illustration. Altogether, neither was an increase in diversity as a result of the virus escaping the immune response observed, nor was a replacement of the dominant viral sequence by a mutant strain seen. This suggests that viruses failed to generate functional mutated LAH proteins. The absence of major non-synonymous substitutions within this region indicates that the amino acid composition is highly restricted. This is in line with the extensive conformational changes that the LAH undergoes during viral growth (Hauck et al., n.d.; Sriwilaijaroen et al., 2012).

5.5 Medical relevance and outlook

Universal Influenza Vaccines are still at the beginning. Careful planning on what is needed (broadly protective versus universal), further investigation of conserved epitopes and effectiveness and safety have to be performed in the future.

One control that might be of interest but not conceivable in the set-up of this study is a control group vaccinated with seasonal influenza vaccine. Seasonal vaccines have been shown to neither induce stalk specific antibodies (Florian Krammer et al., 2015) nor mutation on stalk domain in human (Lee et al., 2015). Therefore, the finding of escape mutations in the stalk region of hemagglutinin following seasonal influenza vaccination was not to be expected. In addition, at the chosen time point for viral quasispecies investigation, antibodies induced by seasonal vaccine have cleared too much virus from the lungs to effectively extract and analyze virus RNA (Hauck et al., n.d.). In this specific set-up a comparison with a control group receiving seasonal vaccine would not have given interpretable results. Nevertheless, universal vaccines need to be judged on their additional benefit compared with seasonal vaccines and further investigation into the appearance of escape mutants might have to be included in new study set-ups. Now that it is becoming conceivable to sequence the whole genome it would be very promising to repeat experiments looking not only at the region of greatest interest but also at escape mutations that may arise by heightening the efficacy of other viral proteins. Another interesting aspect would be the study of simultaneous mutations that together, as permissive mutations, might confer protection from vaccine induced antibodies. Here, obstacles have to be overcome because simultaneousness of mutations would have to be shown on the same virion or population of viruses.

In summary, reaching beyond the vaccination we have at our disposal is a step of financial and health care interest. New questions and concerns arise and call for further investigation. This work was done as one step in this direction to find out whether vaccination against a conserved epitope region will induce escape mutations. It might only be one small part of what needs to be done before we can rule out any harmful outcomes of this new vaccination program. But in its context this work is a promising proof of concept for an effective and safe Universal Influenza Vaccine.

6 References

1. Barclay, W. S. (2017). Influenza: a world of discoveries, outbreaks and controversy. *Journal of General Virology*, *98*, 892–894. https://doi.org/10.1099/jgv.0.000812

2. Barzon, L., Lavezzo, E., Militello, V., Toppo, S. & Palù, G. (2011). Applications of next-generation sequencing technologies to diagnostic virology. *International Journal of Molecular Sciences*. https://doi.org/10.3390/ijms12117861

3. Blokhina, E. A., Kuprianov, V. V., Stepanova, L. A., Tsybalova, L. M., Kiselev, O. I., Ravin, N. V. & Skryabin, K. G. (2013). A molecular assembly system for presentation of antigens on the surface of HBc virus-like particles. *Virology*, *435*(2), 293–300. https://doi.org/10.1016/j.virol.2012.09.014

4. Bragg, L. M., Stone, G., Butler, M. K., Hugenholtz, P. & Tyson, G. W. (2013). Shining a Light on Dark Sequencing: Characterising Errors in Ion Torrent PGM Data. *PLoS Computational Biology*, *9*(4). https://doi.org/10.1371/journal.pcbi.1003031

5. Brooke, C. B., Ince, W. L., Wrammert, J., Ahmed, R., Wilson, P. C., Bennink, J. R. & Yewdell, J. W. (2013). Most Influenza A Virions Fail To Express at Least One Essential Viral Protein. *Journal of Virology*, *87*(6), 3155–3162. https://doi.org/10.1128/JVI.02284-12

6. Bull, R. A., Luciani, F., McElroy, K., Gaudieri, S., Pham, S. T., Chopra, A., Cameron, B., Maher, L., Dore, G. J., White, P. A. & Lloyd, A. R. (2011). Sequential bottlenecks drive viral evolution in early acute hepatitis c virus infection. *PLoS Pathogens*, *7*(9). https://doi.org/10.1371/journal.ppat.1002243

7. Bullard, J. H., Purdom, E., Hansen, K. D. & Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, *11*(1), 94. https://doi.org/10.1186/1471-2105-11-94

8. Bürckert, J.-P. (n.d.). *No Title*.

9. CDC flu diagnosis clinician guidance. (n.d.). Retrieved August 31, 2017, from https://www.cdc.gov/flu/professionals/diagnosis/clinician_guidance_ridt.htm

10. Chai, N., Swem, L. R., Reichelt, M., Chen-Harris, H., Luis, E., Park, S., Fouts, A., Lupardus, P., Wu, T. D., Li, O., McBride, J., Lawrence, M., Xu, M. & Tan, M. W. (2016). Two Escape Mechanisms of Influenza A Virus to a Broadly Neutralizing Stalk-Binding Antibody. *PLoS Pathogens*, *12*(6). https://doi.org/10.1371/journal.ppat.1005702

11. Chamberlain, N. R. (2009). *The big picture. Medical microbiology*. New York and others: McGraw Hill.

12. Domingo, E., Sheldon, J. & Perales, C. (2012). Viral Quasispecies Evolution. *Microbiology and Molecular Biology Reviews*, *76*(2), 159–216. https://doi.org/10.1128/MMBR.05023-11

13. Dwyer, D. E., Smith, D. W., Catton, M. G. & Barr, I. G. (2006). Laboratory diagnosis of human seasonal and pandemic influenza virus infection. *The Medical Journal of Australia*, *185*(10 Suppl), S48-53. Retrieved from http://www.mja.com.au/public/issues/185_10_201106/dwy10867_fm.pdf%5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/17115952

14. Flutcore project outline. (n.d.). Retrieved September 5, 2017, from http://www.flutcore.eu/index.php/project-outline

15. Flutcore tandem core. (n.d.). Retrieved September 5, 2017, from http://www.flutcore.eu/index.php/tandem-core

16. Fock, R., Bergmann, H., Bußmann, H., Fell, G., Finke, E.-J., Koch, U., Niedrig, M., Peters, M., Scholz, D. & Wirtz, A. (2001). Management und Kontrolle einer Influenzapandemie. *Bundesgesundheitsbl- Gesundheitsforsch-Gesundheitsschutz*, *44*, 969–980.

17. Francis, T., Salk, J. E., Pearson, H. E. & Brown, P. N. (1945). Protective effect of vaccination against induced influenza a. *The Journal of Clinical Investigation*, *24*(4), 536–546. https://doi.org/10.1172/JCI101633

18. Frise, R., Bradley, K., van Doremalen, N., Galiano, M., Elderfield, R. A., Stilwell, P., Ashcroft, J. W., Fernandez-Alonso, M., Miah, S., Lackenby, A., Roberts, K. L., Donnelly, C. A. & Barclay, W. S. (2016). Contact transmission of influenza virus between ferrets imposes a looser bottleneck than respiratory droplet transmission allowing propagation of antiviral resistance. *Scientific Reports*, *6*(1), 29793. https://doi.org/10.1038/srep29793

19. Gerdil, C. (2003). The annual production cycle for influenza vaccine. *Vaccine*. https://doi.org/10.1016/S0264-410X(03)00071-9

20. Goodwin, S., McPherson, J. D. & McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet*, *17*(6), 333–351. https://doi.org/10.1038/nrg.2016.49

21. Greenleaf, W. J. & Sidow, A. (2014). The future of sequencing: convergence of intelligent design and market Darwinism. *Genome Biology*, *15*(3), 303. https://doi.org/10.1186/gb4168

22. Gregori, J., Perales, C., Rodriguez-Frias, F., Esteban, J. I., Quer, J. & Domingo, E. (2016). Viral quasispecies complexity measures. *Virology*. https://doi.org/10.1016/j.virol.2016.03.017

23. Grenfell, B. T. (2004). Unifying the Epidemiological and Evolutionary Dynamics of Pathogens. *Science*, *303*(5656), 327–332. https://doi.org/10.1126/science.1090727

24. Hauck, N. C., Kirpach, J., Kiefer, C., Farinelle, S., Maucourant, S., Morris, S., Rosenberg, W., He, F., Muller, C. P. & Lu, I.-N. (n.d.). *Next generation sequencing reveals a constrained viral quasispecies evolution under crossreactive antibody pressure targeting long alpha helix of hemagglutinin*.

25. Heather, J. M. & Chain, B. (2016). The sequence of sequencers: The history of sequencing DNA. *Genomics*. https://doi.org/10.1016/j.ygeno.2015.11.003

26. Heaton, N. S., Sachs, D., Chen, C.-J., Hai, R. & Palese, P. (2013). Genome-wide mutagenesis of influenza virus reveals unique plasticity of the hemagglutinin and NS1 proteins. *Proceedings of the National Academy of Sciences*, *110*(50), 20248–20253. https://doi.org/10.1073/pnas.1320524110

27. Hernandez-Vargas, E. a., Wilk, E., Canini, L., Toapanta, F. R., Binder, S. C., Uvarovskii, A., Ross, T. M., Guzmán, C. a., Perelson, A. S. & Meyer-Hermann, M. (2014). Effects of Aging on Influenza Virus Infection Dynamics. *J. Virol.*, *88*(8), 4123–4131. https://doi.org/10.1128/JVI.03644-13

28. Informationen zur neuen Grippe. (n.d.). Retrieved September 2, 2017, from https://www.bundesregierung.de/Content/DE/Magazine/MagazinSozialesFamilieBildung/080/t3-informationen-zur-neuen-grippe.html

29. Ion PGM™ System Specifications. (n.d.). Retrieved September 6, 2017, from https://www.thermofisher.com/de/de/home/life-science/sequencing/next-generation-sequencing/ion-torrent-next-generation-sequencing-workflow/ion-torrent-next-generation-sequencing-run-sequence/ion-pgm-system-for-next-generation-sequencing/ion-pgm-system-spe

30. Joyce, M. G., Wheatley, A. K., Thomas, P. V., Chuang, G. Y., Soto, C., Bailer, R. T., Druz, A., Georgiev, I. S., Gillespie, R. A., Kanekiyo, M., Kong, W. P., Leung, K., Narpala, S. N., Prabhakaran, M. S., Yang, E. S., Zhang, B., Zhang, Y., Asokan, M., Boyington, J. C., Bylund, T., Darko, S., Lees, C. R., Ransier, A., Shen, C. H., Wang, L., Whittle, J. R., Wu, X., Yassine, H. M., Santos, C., Matsuoka, Y., Tsybovsky, Y., Baxa, U., Mullikin, J. C., Subbarao, K., Douek, D. C., Graham, B. S., Koup, R. A., Ledgerwood, J. E., Roederer, M., Shapiro, L., Kwong, P. D., Mascola, J. R. & McDermott, A. B. (2016). Vaccine-Induced Antibodies that Neutralize Group 1 and Group 2 Influenza A Viruses. *Cell*, *166*(3), 609–623. https://doi.org/10.1016/j.cell.2016.06.043

31. Kinde, I., Wu, J., Papadopoulos, N., Kinzler, K. W. & Vogelstein, B. (2011). Detection and quantification of rare mutations with massively parallel sequencing. *Proceedings of the National Academy of Sciences*, *108*(23), 9530–9535. https://doi.org/10.1073/pnas.1105422108

32. Krammer, F. (2015). Emerging influenza viruses and the prospect of a universal influenza virus vaccine. *Biotechnology Journal*. https://doi.org/10.1002/biot.201400393

33. Krammer, F. & Palese, P. (2013). Influenza virus hemagglutinin stalk-based antibodies and vaccines. *Current Opinion in Virology*, *3*(5), 521–530. https://doi.org/10.1016/j.coviro.2013.07.007

34. Krammer, F. & Palese, P. (2015). Advances in the development of influenza virus vaccines. *Nature Reviews Drug Discovery*, *14*(3), 167–182. https://doi.org/10.1038/nrd4529

35. Krammer, F., Pica, N., Hai, R., Margine, I. & Palese, P. (2013). Chimeric Hemagglutinin Influenza Virus Vaccine Constructs Elicit Broadly Protective Stalk-Specific Antibodies. *Journal of Virology*, *87*(12), 6542–6550. https://doi.org/10.1128/JVI.00641-13

36. Laursen, N. S. & Wilson, I. A. (2013). Broadly neutralizing antibodies against influenza viruses. *Antiviral Research*. https://doi.org/10.1016/j.antiviral.2013.03.021

37. Lee, A. J., Das, S. R., Wang, W., Fitzgerald, T., Pickett, B. E., Aevermann, B. D., Topham, D. J., Falsey, A. R. & Scheuermann, R. H. (2015). Diversifying Selection Analysis Predicts Antigenic Evolution of 2009 Pandemic H1N1 Influenza A Virus in Humans. *Journal of Virology*, *89*(10), 5427–5440. https://doi.org/10.1128/JVI.03636-14

38. Liu, H., Frijlink, H. W., Huckriede, A., van Doorn, E., Schmidt, E., Leroy, O., Rimmelzwaan, G., McCullough, K., Whelan, M. & Hak, E. (2016). Influenza Vaccine Research funded by the European Commission FP7-Health-2013-Innovation-1 project. *Vaccine*. https://doi.org/10.1016/j.vaccine.2016.10.040

39. Loman, N. J., Misra, R. V, Dallman, T. J., Constantinidou, C., Gharbia, S. E., Wain, J. & Pallen, M. J. (2012). Performance comparison of benchtop high-throughput sequencing platforms. *Nature Biotechnology*, *30*(5), 434–439. https://doi.org/10.1038/nbt.2198

40. Lu, I.-N., Farinelle, S., Sausy, A. & Muller, C. P. (2017). Identification of a CD4 T-cell epitope in the hemagglutinin stalk domain of pandemic H1N1 influenza virus and its antigen-driven TCR usage signature in BALB/c mice. *Cellular & Molecular Immunology*, (14), 511–520. https://doi.org/10.1038/cmi.2016.20

41. Luo, C., Hirsch, H. H., Kant, J. & Randhawa, P. (2012). VP-1 quasispecies in human infection with polyomavirus BK. *Journal of Medical Virology*, *84*(1), 152–161. https://doi.org/10.1002/jmv.22147

42. Mckimm-Breschkin, J. L. (2013). Influenza neuraminidase inhibitors: Antiviral action and mechanisms of resistance. *Influenza and Other Respiratory Viruses*. https://doi.org/10.1111/irv.12047

43. Medina, R. A. & García-Sastre, A. (2011). Influenza A viruses: new research developments. *Nature Reviews Microbiology*, *9*(8), 590–603. https://doi.org/10.1038/nrmicro2613

44. Nachbagauer, R. & Krammer, F. (2017). Universal influenza virus vaccines and therapeutic antibodies. *Clinical Microbiology and Infection*. https://doi.org/10.1016/j.cmi.2017.02.009

45. Nationaler Pandemieplan. Teil I. (n.d.). Retrieved September 2, 2017, from https://www.gmkonline.de/documents/Pandemieplan_Teil-I.pdf

46. Neu, K. E., Henry Dunand, C. J. & Wilson, P. C. (2016). Heads, stalks and everything else: how can antibodies eradicate influenza as a human disease? *Current Opinion in Immunology*. https://doi.org/10.1016/j.coi.2016.05.012

47. Paules, C. & Subbarao, K. (2017). Influenza. *Lancet*. https://doi.org/10.1016/S0140-6736(17)30129-0

48. Pawelek, K. A., Huynh, G. T., Quinlivan, M., Cullinane, A., Rong, L. & Perelson, A. S. (2012). Modeling within-host dynamics of influenza virus infection including immune responses. *PLoS Computational Biology*, *8*(6). https://doi.org/10.1371/journal.pcbi.1002588

49. Peyret, H., Gehin, A., Thuenemann, E. C., Blond, D., El Turabi, A., Beales, L., Clarke, D., Gilbert, R. J. C., Fry, E. E., Stuart, D. I., Holmes, K., Stonehouse, N. J., Whelan, M., Rosenberg, W., Lomonossoff, G. P. & Rowlands, D. J. (2015). Tandem fusion of hepatitis B core antigen allows assembly of virus-like particles in bacteria and plants with enhanced capacity to accommodate foreign proteins. *PLoS ONE*, *10*(4). https://doi.org/10.1371/journal.pone.0120751

50. Pinto, L. H. & Lamb, R. A. (2006). The M2 proton channels of influenza A and B viruses. *Journal of Biological Chemistry*. https://doi.org/10.1074/jbc.R500020200

51. QIAamp® Viral RNA Mini Handbook. (n.d.). Retrieved January 8, 2016, from https://www.qiagen.com/fi/resources/resourcedetail?id=c80685c0-4103-49ea-aa72-8989420e3018&lang=en

52. Quail, M., Smith, M. E., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., Bertoni, A., Swerdlow, H. P. & Gu, Y. (2012). A tale of three next generation sequencing platforms: comparison of Ion torrent, pacific biosciences and illumina MiSeq sequencers. *BMC Genomics*, *13*(1), 341. https://doi.org/10.1186/1471-2164-13-341

53. Rapaport, F., Khanin, R., Liang, Y., Pirun, M., Krek, A., Zumbo, P., Mason, C. E., Socci, N. D. & Betel, D. (2013). Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome Biology*, *14*(9), R95. https://doi.org/10.1186/gb-2013-14-9-r95

54. Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., Leamon, J. H., Johnson, K., Milgrew, M. J., Edwards, M., Hoon, J., Simons, J. F., Marran, D., Myers, J. W., Davidson, J. F., Branting, A., Nobile, J. R., Puc, B. P., Light, D., Clark, T. A., Huber, M., Branciforte, J. T., Stoner, I. B., Cawley, S. E., Lyons, M., Fu, Y., Homer, N., Sedova, M., Miao, X., Reed, B., Sabina, J., Feierstein, E., Schorn, M., Alanjary, M., Dimalanta, E., Dressman, D., Kasinskas, R., Sokolsky, T., Fidanza, J. A., Namsaraev, E., McKernan, K. J., Williams, A., Roth, G. T. & Bustillo, J. (2011). An integrated semiconductor

device enabling non-optical genome sequencing. *Nature*, *475*(7356), 348–352. https://doi.org/10.1038/nature10242

55. Shaw, M. L. & Palese, P. (2013). Orthomyxoviridae. In *Fields Virology* (pp. 1151–1185). https://doi.org/9781451105636

56. Sriwilaijaroen, N. & Suzuki, Y. (2012). Molecular basis of the structure and function of H1 hemagglutinin of influenza virus. *Proceedings of the Japan Academy, Series B*, *88*(6), 226–249. https://doi.org/10.2183/pjab.88.226

57. Steel, J., Lowen, A. C., Wang, T. T., Yondola, M., Gao, Q., Haye, K., García-Sastre, A. & Palesea, P. (2010). Influenza virus vaccine based on the conserved hemagglutinin stalk domain. *mBio*, *1*(1). https://doi.org/10.1128/mBio.00018-10

58. Theisen, L. L. & Muller, C. P. (2012). EPs® 7630 (Umckaloabo®), an extract from Pelargonium sidoides roots, exerts anti-influenza virus activity in vitro and in vivo. *Antiviral Research*, *94*(2), 147–156. https://doi.org/10.1016/j.antiviral.2012.03.006

59. Tong, S., Zhu, X., Li, Y., Shi, M., Zhang, J., Bourgeois, M., Yang, H., Chen, X., Recuenco, S., Gomez, J., Chen, L. M., Johnson, A., Tao, Y., Dreyfus, C., Yu, W., McBride, R., Carney, P. J., Gilbert, A. T., Chang, J., Guo, Z., Davis, C. T., Paulson, J. C., Stevens, J., Rupprecht, C. E., Holmes, E. C., Wilson, I. A. & Donis, R. O. (2013). New World Bats Harbor Diverse Influenza A Viruses. *PLoS Pathogens*, *9*(10). https://doi.org/10.1371/journal.ppat.1003657

60. Tucker, S. C. (ed. . (2005). *The Encyclopedia of World War I. A Political, Social and Military History*. Santa Barbara, California: ABC Clio.

61. Vatti, A., Monsalve, D. M., Pacheco, Y., Chang, C., Anaya, J. M. & Gershwin, M. E. (2017). Original antigenic sin: A comprehensive review. *Journal of Autoimmunity*. https://doi.org/10.1016/j.jaut.2017.04.008

62. Vollmers, C., Sit, R. V., Weinstein, J. A., Dekker, C. L. & Quake, S. R. (2013). Genetic measurement of memory B-cell recall using antibody repertoire sequencing. *Proceedings of the National Academy of Sciences*, *110*(33), 13463–13468. https://doi.org/10.1073/pnas.1312146110

63. Wang, T. T., Tan, G. S., Hai, R., Pica, N., Ngai, L., Ekiert, D. C., Wilson, I. A., Garcia-Sastre, A., Moran, T. M. & Palese, P. (2010). Vaccination with a synthetic peptide from the influenza virus hemagglutinin provides protection against distinct viral subtypes. *Proceedings of the National Academy of Sciences*, *107*(44), 18979–18984. https://doi.org/10.1073/pnas.1013387107

64. WHO vaccine recommendations. (n.d.). Retrieved September 3, 2017, from http://www.who.int/influenza/vaccines/virus/recommendations/2017_18_north/en/

65. World Health Organization. (2014). Influenza (Seasonal): fact sheet. https://doi.org/D

66. Wright, P. F., Neumann, G. & Kawaoka, Y. (2013). Chapter 41 Orthomyxoviruses. In *Fields Virology, 6th Edition* (Vol. 1, pp. 1186–1243). https://doi.org/1-4511-0563-0

67. Zhang, H., Wang, L., Compans, R. W. & Wang, B. Z. (2014). Universal influenza vaccines, a dream to be realized soon. *Viruses*. https://doi.org/10.3390/v6051974

68. Zheng, D., Chen, S., Qu, D., Chen, J., Wang, F., Zhang, R. & Chen, Z. (2016). Influenza H7N9 LAH-HBc virus-like particle vaccine with adjuvant protects mice against homologous and heterologous influenza viruses. *Vaccine*, *34*(51), 6464–6471. https://doi.org/10.1016/j.vaccine.2016.11.026

69. Zheng, M., Luo, J. & Chen, Z. (2014). Development of universal influenza vaccines based on influenza virus M and NP genes. *Infection*. https://doi.org/10.1007/s15010-013-0546-4

7 Annex

7.1 Publications

- "Next generation sequencing reveals a constrained viral quasispecies evolution under cross-reactive antibody pressure targeting long alpha helix of hemagglutinin" Nastasja C. Hauck[1¶], Josiane Kirpach[1¶], Christina Kiefer[1], Sophie Farinelle[1], Sophie Maucourant[2], Steven A. Morris[2], William Rosenberg[2], Feng He[1], Claude P. Muller[1], I-Na Lu[1*]

  [1] Department of Infection and Immunity, Luxembourg Institute of Health, 29, rue Henri Koch, L-4354 Esch-sur-Alzette, Luxembourg
  [2] iQur Ltd, London, United Kingdom
  [¶] These authors contributed equally to this work
  *Corresponding author

  Under revision

  Parts of the paper (including figures) have been used for this thesis.

7.2 Conference participations

- "B cell response and escape mutants", Nastasja Hauck, Josiane Kirpach, Sophie Farinelle, Aurélie Sausy, Regina Sinner, Jean-Philippe Bürckert, William Faison, Dr I-Na Lu, Prof Dr Claude P Muller, FLUTCORE meeting, London, Great Britain, 23rd September 2015, Oral presentation

- "Antibodies against hemagglutinin long alpha helix of pandemic H1N1 do not increase variability of viral quasispecies", Nastasja Hauck, Josiane Kirpach, Christina Kiefer, Sophie Farinelle, Dr Feng He, Dr I-Na Lu, Prof Dr Claude P Muller, School of Influenza, Siena, Italy, 11/04/2016- 15/04/2016, Oral presentation

7.3 Pipeline for data processing

7.3.1 Distribution_thresholds.py

```
#!/bin/Python

##############################################################################
#                                                                            #
#                                  imports                                   #
#                                   input                                    #
#                                 functions                                  #
#                                                                            #
##############################################################################

import sys
import os

#input distribution files
distribution_folder = sys.argv[1]
min_read_coverage = 3

given_reads = {}
#given_reads["18"] = 3682546
given_reads["21"] = 1731069
given_reads["22"] = 5835938
given_reads["24"] = 4293287
given_reads["26"] = 4458459
given_reads["28"] = 5249236
given_reads["30"] = 5316885

#This function reads in a file and saves everything into the two given
#dictionaries.
#One contains the number of reads for every pool
#One contains the distribution (How many UIDs have a certain coverage)
def read_file(filename, distribution_dict, number_of_reads):
    pool_name = file.split("_")[1]

    #gets the dictionary for the pool if it exits or creates a new one
    if pool_name in distribution_dict:
        pool_dict = distribution_dict[pool_name]
    else:
        pool_dict = {}
    #to count the number of reads for this file
    file_read_number = 0

    #read the file
    with open(distribution_folder + "/" + file) as f:
        content = f.readlines()

    #for every line in the file
    for line in content:
        #we get the uid and how often it occurs
        unique_uid = line.strip().split("\t")[0]
        num_of_occ = line.strip().split("\t")[1]
        #the sum of occurences is the number of reads for this pool
        #because every uid is unique
        file_read_number += int(num_of_occ)
        #at least 5 different UIDs exist
        if len(content) > 5:
            #then it is not a chance match and we want to use it
            #we save the uid in a list with all uids which occured the same
            #number of times
            if num_of_occ in pool_dict:
                pool_dict[num_of_occ].append(unique_uid)
            else:
                pool_dict[num_of_occ] = [unique_uid]
    #update the dictionary
    distribution_dict[pool_name] = pool_dict

    #save number of reads for this pool
    if pool_name in number_of_reads:
        number_of_reads[pool_name] = number_of_reads[pool_name] + file_read_number
    else:
        number_of_reads[pool_name] = file_read_number
    #return the values so we can use them later
    return (distribution_dict, number_of_reads)

#for all pools
```

```
#returns a dictionary containing the number of reads smaller
#a given threshold
def get_reads_smaller_threshold(dist_dict, threshold):
    read_num = {}
    for pool in dist_dict:
        pool_dict = dist_dict[pool]
        num_of_reads = 0
        for num in pool_dict:
            if int(num) < threshold:
                #number of reads is the coverage times the number of UIDs
                #which have this coverage
                num_of_reads += (int(num)*len(pool_dict[num]))
        read_num[pool] = num_of_reads
    return read_num

#for only one pool
#returns the number of reads which are greater or equal a given
#coverage threshold
def get_reads_greater_eq_threshold(pool_dict, threshold):
    num_of_reads = 0
    for num in pool_dict:
        if int(num) >= threshold:
            #number of reads is the coverage times the number of UIDs
            #which have this coverage
            num_of_reads += (int(num)*len(pool_dict[num]))
    return num_of_reads

#function that returns the max coverage of the pool
def get_max_coverage(pool_dict):
    cov_list = []
    for cov in pool_dict:
        cov_list.append(int(cov))
    return max(cov_list)

#function that returns a threshold value for the different pools
#so that approximatly the same percentage of reads in every pool
#is regarded
def read_percent_to_threshold(given_reads, dist_dict, percent, threshold_dictionary):
    for pool in dist_dict:
        pool_dict = dist_dict[pool]
        max_cov = get_max_coverage(pool_dict)
        #print "MAX cov for pool %s is %i" %(pool, max_cov)
        num_reads = 0
        all_reads = given_reads[pool]
        for cov in range(1,max_cov):
            num_reads = get_reads_greater_eq_threshold(pool_dict, cov)
            new_percent = float(num_reads) / float(all_reads) * 100.0
            if new_percent <= percent:
                threshold_dictionary[pool] = cov
                break
    return threshold_dictionary

################################################################################
#                                                                              #
#                                                                              #
#                                  Main Script                                 #
#                                                                              #
#                                                                              #
################################################################################



dist_dict = {}
initial_read_num = {}

for file in os.listdir(str(distribution_folder+"/")):
    if file.endswith(".txt"):
        #call our function so that we get a dictionary which contains a
        #dictionary for every pool, containing the read distribution
        #and a dictionary which knows how many reads are in the pool
        dist_dict, initial_read_num = read_file(file, dist_dict, initial_read_num)

#get overall number of reads for every pool, a threshold we will never reach is
#choosen to get all reads
read_num = get_reads_smaller_threshold(dist_dict, 100000)
reads_smaller_three = get_reads_smaller_threshold(dist_dict, min_read_coverage)

#get the smallest percent of used reads for all pools
```

```
min_pool = ""
min_percent = 100.0
for pool in read_num:
    percent_used = float(read_num[pool]) / given_reads[pool] * 100.0
    percent_used_cov = (float(read_num[pool]) - float(reads_smaller_three[pool]) ) /
given_reads[pool] * 100.0
    if percent_used_cov < min_percent:
        min_pool = pool
        min_percent = percent_used_cov

print "The min percent of reads used is %.2f%% in pool %s" %(min_percent, min_pool)

threshold_dict = {}
#get the best matching threshold for this percentage
threshold_dict = read_percent_to_threshold(given_reads, dist_dict, min_percent,
threshold_dict)
threshold_dict[min_pool] = min_read_coverage

#print the percentage and matching threshold for every pool
for pool in threshold_dict:
    pool_dict = dist_dict[pool]
    reads_for_pool = get_reads_greater_eq_threshold(pool_dict, threshold_dict[pool])
    percent_reads = float(reads_for_pool) / float(given_reads[pool]) * 100.0
    print "The min coverage for pool %s should be %i covering %.2f%% of reads" %(pool,
threshold_dict[pool], percent_reads)
    reads_for_pool = get_reads_greater_eq_threshold(pool_dict, threshold_dict[pool]-1)
    percent_reads = float(reads_for_pool) / float(given_reads[pool]) * 100.0
    print "The min coverage minus 1 for pool %s should be %i covering %.2f%% of reads"
%(pool, threshold_dict[pool]-1, percent_reads)
    reads_for_pool = get_reads_greater_eq_threshold(pool_dict, threshold_dict[pool]+1)
    percent_reads = float(reads_for_pool) / float(given_reads[pool]) * 100.0
    print "The min coverage plus 1 for pool %s should be %i covering %.2f%% of reads"
%(pool, threshold_dict[pool]+1, percent_reads)


#This is for the percentage table from the presenentation
#read percent used = [20, 25, 30, 35, 40, 45, 50, 55, 60]
#for p in read_percent_used:
#    thres_dict = {}
#    thres dict = read percent_to_threshold(given_reads, dist_dict, p, thres_dict)
#    for pool in thres dict:
#        pool dict = dist dict[pool]
#        print " %.2f%% pool %s coverage %i" %(p, pool, thres_dict[pool])
```

7.3.2 Variants.py

```
#!/usr/bin/python

#Bsp. Aufruf:

#python variants.py --infile R_2015_11_10_13_38_01_user_SN2-355-NH_FlutCORE_PGM-pool-
2015-021_Auto_user_SN2-355-NH_FlutCORE_PGM-pool-2015-021_448.basecaller.bam --fileprefix
pool_21 --primer_forward CACAGTTCACAGCAGTAGGTAAAGA --primer_reverse ACAAATGCGATAACACGTGC
--uid threshold 19 --poly A --quality --threshold_consens
consens_threshold_dependant_on_coverage.tsv

#############################################################################
#                                                                           #
#                                imports                                    #
#                                                                           #
#############################################################################

#here all modules we need are imported

#this means we tell python that we want to use a

#functionality provided by this import

import sys

import os

import argparse

import subprocess
```

```
import shlex

#from io import StringIO

from StringIO import StringIO

from Bio import AlignIO

from Bio.Align import AlignInfo

from Bio.Align.Applications import MafftCommandline

from collections import defaultdict

################################################################################
#                                                                              #
#                            commandline parameters                            #
#                                                                              #
################################################################################

parser = argparse.ArgumentParser(description='Processing ionTorrent reads from a bam
file to get information about how often a virus variant occurs by creating consensus
sequences for every unique identifier. The analysis is split up for every MID.')

#required arguments

parser.add_argument('--infile',required=True ,help='The input bam file from the
ionTorrent')

parser.add_argument('--fileprefix',required=True, action='store',
dest='fileprefix',help='Prefix used in intermediate filenames') # input gtf file

parser.add_argument('--primer_forward', action='store', dest='primer_forward', type=str,
required=True, help='A sequene which stands before the beginning of the regarded
sequence.')

parser.add_argument('--primer_reverse', action='store', dest='primer_reverse', type=str,
required=True, help='A sequence following the last character of the regarded sequence.')

#forward='CACAGTTCACAGCAGTAGGTAAAGA'

#backward='ACAAATGCGATAACACGTGC'

#optional arguments

#max_seq_length=300  #no longer needed

parser.add_argument('--quality', action='store_true', default=False, help='If quality
filterng using fastx tools should be used set this flag.')

parser.add_argument('--quality_min', action='store', dest='quality_min',type=int,
default=20, help='Min base quality to keep, default value is 20.')

parser.add_argument('--quality_percent', action='store',
dest='quality_percent',type=int, default=95, help='Percent of bases which should have
min quality, default is 95 Percent.')

parser.add_argument('--mask', action='store_true', default=False, help='If masking of
bases below a minimum quality is required, set this flag.')

parser.add_argument('--filter_length', action='store_true', dest='filter_length',
default=False, help='If sequences below a certain length should be excluded, set this
flag.')

parser.add_argument('--min_seq_length', action='store', dest='min_seq_length',
default=270, help='The min length a sequence needs to have.')

parser.add_argument('--max_seq_length', action='store', dest='max_seq_length',
default=500, help='The max length a sequence can to have.')

parser.add_argument('--poly_A', action='store_true', dest='poly_A', default=False,
help='If this flag is set a correction for the three known poly a stretches is
performed.')

#parser.add argument('--fastq', action='store_true', default=False, help='fastq flies
are kept as far as possible')
```

```
parser.add_argument('--uid_threshold', action='store', dest='uid_threshold', type=int,
default=3, help='Minimum coverage a uid has to have.')

parser.add_argument('--threshold_consens', action='store', dest='threshold_consens',
help='A file containing the minimum number of sequences which have to be the same for
the letter to be called in the consensus ssequence.')

arguments = parser.parse_args()

assert os.path.isfile(arguments.infile), "Bam input file not found '%s' " %
arguments.infile

#create folders to save the files

main_folder = arguments.fileprefix + "/"

if not os.path.exists(main_folder):

    os.mkdir(main_folder)

#all_files = main_folder + "intermediatefiles_"+ arguments.fileprefix + "/"

#if not os.path.exists(all_files):

#    os.mkdir(all_files)

intermed = main_folder + "MIDS/"

if not os.path.exists(intermed):

    os.mkdir(intermed)


results_folder = main_folder + "RESULTS/"

if not os.path.exists(results_folder):

    os.mkdir(results_folder)

#print(arguments)

print("")

print("")

print("")

print("                          Infile: %s" %(arguments.infile))

print("        Results can be found in: %s" %(results_folder))

print("   All in files can be found in: %s" %(main_folder))

print("             The file prefix is: %s" %(arguments.fileprefix))

print("The Parameters for this run are:")

if arguments.quality:

    print("Filtering for qulatity using the parameters:")

    print("                          Min quality to keep: %i" %(arguments.quality_min))

    print("Percent of bases that must have min qualtity: %i"
%(arguments.quality_percent))

else:

    print("No quality filtering is performed.")

if arguments.mask:

    print("Bases with a quality score below %i will be masked."
%(arguments.quality_min))

else:
```

```
    print("No masking for bad phred quality score is performed.")

if arguments.filter_length:

    print("Sequences will have to be longer than %i bases and shorter %i bases."
%(arguments.min_seq_length, arguments.max_seq_length))

else:

    print("No filtering for sequences which are to long or to short.")

if arguments.poly_A:

    print("The three poly A stretches are shortend if they are longer than 6")

else:

    print("No change is performed on Poly A stretches.")

print("A min number of %i seqences starting with the same UID is needed to consider the
UID for creating a consensus sequence" %(arguments.uid_threshold))

print("")

print("")

print("")

################################################################################
#                                                                              #
#                                    functions                                 #
#                                                                              #
################################################################################

#needs the sequence and the matching score line

#returns the sequence shortend to between primers

def find_seq_between_primer(seq, score, forward, reverse):

    res=""

    primer_found=False

    try:

        primer_forward_index = seq.index(forward)

        primer_found=True

    except:

        res="NA"

    #gets the starting position of the reverse primer

    try:

        primer_reverse_index = seq.index(reverse)

    except:

        primer_found=False

        res="NA"

    if primer_found:

        #the sequence becomes the sequence starting from the primer_forward

        #assert(len(seq) == len(score))

        new_seq = seq[ (primer_forward_index + len(forward)) : primer_reverse_index ]

        new_score = score[ (primer_forward_index + len(forward)) : primer_reverse_index
]

        res = [new_seq, new_score]
```

```
    return res
#reads in the fastq file are sorted by MIDs
#each MID is associated with a dictionary using UIDs as keys
#every UID entry has a list with lists of the sequence entry
def read_fastq(fastq_file_name, mid_dict):
    counter = 0
    for line in open(fastq_file_name):
        counter += 1
        #if it is a fastq file a read consitst of 4 lines
        #if the counter is 1 the line is the sequence header
        if counter == 1:
            header_seq = line
            #print header_seq.strip('\n')
        #if the counter is 2 the line is the sequence
        if counter == 2:
            seq = line
        #if the counter is 3 the line is the header of the score
        if counter == 3:
            header_score = line
        #if the counter is 4 the line is the line of scores
        #everything we want to do with the lines should be done in this step
        if counter == 4:
            score = line
            counter=0
            #take mid as first dict key
            mid = seq[0:10]
            uid = seq[10:24]
            if mid in mid_dict:
                dict_of_mid = mid_dict[mid]
                if uid in dict_of_mid:
                    uid_list = dict_of_mid[uid]
                    uid_list.append([header_seq, seq, header_score, score])
                    dict_of_mid[uid] = uid_list
                else:
                    dict_of_mid[uid] = [[header_seq, seq, header_score, score]]
                mid_dict[mid] = dict_of_mid
    return mid_dict
#this function filters for length of the sequences
def filter_for_length(mid_dict, min_length, max_length):
```

```
    new_mid_dict = {}
    for mid in mid_dict:
        m_dict = mid_dict[mid]
        new_uid_dict = {}
        for uid in m_dict:
            #list of entries for this uid
            u = m_dict[uid]
            u_list = []
            for entry in u:
                #if sequence is longer min and shorter max
                if (len(entry[1]) > min_length) and (len(entry[1]) < max_length):
                    #then add to the list
                    u_list.append(entry)
            #if sequences with this uid are left
            if len(u_list) > 0:
                new_uid_dict[uid] = u_list
        new_mid_dict[mid] = new_uid_dict
    return new_mid_dict
#prints the uid and how often it occures in one file
def creating_distribution_files(mid_dict, mid_to_seq):
    for mid in mid_dict:
        #folder = mid_folder[mid]
        distribution_file = intermed + mid_to_seq[mid] + "_unique_uids.csv"
        dist_file = open(distribution_file, 'w')
        for uid in mid_dict[mid]:
            cov = len(mid_dict[mid][uid])
            dist_file.write(uid + "," + str(cov) + "\n")
        dist_file.close()
def count_reads(mid_dict):
    read_counter = 0
    for mid in mid_dict:
        for uid in mid_dict[mid]:
            read_counter += len(mid_dict[mid][uid])
    return read_counter
def count_reads_per_mid(mid_dict):
    read_num_dict = {}
    for mid in mid_dict:
        read_counter = 0
        for uid in mid_dict[mid]:
```

```
            read_counter += len(mid_dict[mid][uid])
        read_num_dict[mid] = read_counter
    return read_num_dict
def correct_for_poly_A(mid_dict, mid_folder, mid_to_seq):
    positions=[[60,73],[84,97],[218,231]]
    ploy_A_seven="AAAAAAA"
    new_mid_dict = {}
    for mid in mid_dict:
        modifications = mid_folder + mid_to_seq[mid] + "_poly_A_modifications.txt"
        mod_file = open(modifications, 'w')
        uid_dict = mid_dict[mid]
        new_uid_dict = {}
        for uid in uid_dict:
            new_entry_list = []
            for entry in uid_dict[uid]:
                #print "enty for uid %i " %(len(entry))
                new_entry = [entry[0],entry[1],entry[2],entry[3]]
                ind = 0
                seq = entry[1]
                score = entry[3]
                pos_to_delete = []
                for pos in positions:
                    try:
                        #if a ploy A stretch of seven or longer is detected the starting
index is saved
                        ind = seq.index(ploy_A_seven, pos[0], pos[1])
                        #in the list of positions which should be deleted
                        pos_to_delete.append(ind+6) #because we want to delete the last
A
                        #and every A in the poly a stretch following the sixth A
                        #the loop is performed as long as a is True
                        a=True
                        #index in the sequence of the seventh A
                        i=ind+6
                        while a:
                            #the index in the string is counted up
                            i+=1
                            if (seq[i] == 'A'):
                                #if it is A, the position needs to be deleted
                                #for that it is added to the pos_to_delete list
```

```
                                pos_to_delete.append(i)
                            else:
                                #the poly A stretch is over ass soon as the position
                                #contains an other character than A, so a is set to
                                #false which meens the loop is not performed any more
                                a=False
                        except:
                            #everything is normal, the poly A stretch is not to long
                            ind = 0
                    if pos_to_delete != []:
                        #reverse the order so that the highest position is deleted first and
                        #no problems with a moving index is created
                        for i in reversed(pos_to_delete):
                            assert(len(seq) == len(score))
                            #remove according to index
                            #largest index is removed first
                            #because the seq and the score lines have the same length
                            #the position is deleted from both
                            if(i < len(seq)):
                                phred=str(ord(score[i])-33)
                                mod = entry[0].strip("\n") + "\t" + seq[1:24] + "\t" +
seq[i] + "\t" + phred + "\n"
                                #print mod
                                mod_file.write(mod)
                                seq = seq[:i] + seq[i+1:]
                                score = score[:i] + score[i+1:]
                                new_entry = [entry[0], seq, entry[2], score]
                                #print "Corrected for a poly A stretch"
                        new_entry_list.append(new_entry)
                    new_uid_dict[uid] = new_entry_list
                    #print "new entry length %i " %(len(new_entry_list))
                new_mid_dict[mid] = new_uid_dict
                #print "length mid dict %i " %(len(new_mid_dict))
                mod_file.close()
        return new_mid_dict
    def read_consensus_threshold_file(threshold_file, threshold_dict):
        for line in open(threshold_file):
            content = line.strip().split('\t')
            coverage = content[0]
```

```
            percentage = content[1]
            percentage = percentage.replace(',','.')
            threshold_dict[coverage] = percentage
        return threshold_dict
    def create_align(fasta_file_name, uid_reads, threshold_dict):
        number_of_reads = 0
        new_fasta = open(fasta_file_name, 'w')
        for entry in uid_reads:
            number_of_reads +=1
            header = entry[0].strip()
            sequence = entry[1].strip()
            new_fasta.write(">" + header[1:] + "\n" + sequence + "\n")
        new_fasta.close()
        #the alighnment is created using mafft
        mafft_cline = MafftCommandline(input=fasta_file_name)
        stdout, stderr = mafft_cline()
        #here the alighnment is read from the mafft output
        align = AlignIO.read(StringIO(stdout), "fasta")
        #get the consensus sequence from the alignment
        summary_align = AlignInfo.SummaryInfo(align)
        threshold_consens = threshold_dict[str(number_of_reads)]
        float_thres = float(threshold_consens) / 100.0
        consensus = summary_align.gap_consensus(ambiguous="N", threshold=(float_thres-0.1))
        #starts from end of primer_forward to beginning of primer_reverse
        #immediatley use consensus sequence without gaps
        #table = {ord('-') : None}
        #consensus_without_gap = str(consensus).translate(table)
        consensus_without_gap = str(consensus).translate(None, '-')
        return [header, consensus_without_gap]
    def count_consensus_seq(consens_dict, result_file):
        num_sequences = len(consens_dict)
        count = dict()
        for consensus in consens_dict:
            consensus_seq = consens_dict[consensus][0]
            num_of_reads = consens_dict[consensus][1]
            if consensus_seq in count:
                count[consensus_seq][0] += 1
                count[consensus_seq][1] += num_of_reads
            else:
```

```
            count[consensus_seq] = [1 , num_of_reads]

    stats = open(result_file, 'w')

    for seq in count:

        p = float(count[seq][0])/float(num_sequences)

        line = "%s,%i,%.3f,%i\n" %(seq, count[seq][0], p, count[seq][1] )

        #line = seq + ',' + str(count[seq]) + ',' +
str(float(count[seq])/float(num_sequences) + "\n")

        stats.write(line)

    stats.close()

################################################################################
#                                                                              #
#                                    analysis                                  #
#                                                                              #
################################################################################

#read in individual threshold

threshold_dict = {}

threshold_dict = read_consensus_threshold_file(arguments.threshold_consens,
threshold_dict)

#mid_to_seq = {"CGAGGTTATC":"MID30"  }

mid_to_seq = {"TTCGTGATTC":"MID07" , "TCCTCGAATC":"MID11" , "TTAGTCGGAC":"MID19" ,
"CGAGGTTATC":"MID30" }

#mid_to_seq = {"TTCGTGATTC":"MID07"}

#mid_seq = ["TTCGTGATTC", "TCCTCGAATC", "TTAGTCGGAC", "CGAGGTTATC" ]

#mids = ["MID07", "MID11", "MID19", "MID30"]

mid_dict = {"TTCGTGATTC": {}, "TCCTCGAATC": {}, "TTAGTCGGAC": {}, "CGAGGTTATC": {} }


mid_folder = {}

for mid in mid_to_seq:

    mid_folder_name = intermed + mid_to_seq[mid] + "/"

    if not os.path.exists(mid_folder_name):

        os.mkdir(mid_folder_name)

    mid_folder[mid] = mid_folder_name

#convert the bam file to a fastq file

fastq_file_name = main_folder + arguments.fileprefix + ".fastq"

#bam_call = "samtools fastq %s" %(arguments.infile)

bam_call = "bam2fastx -q %s" %(arguments.infile)

args = shlex.split(bam_call)

fastq_file = open(fastq_file_name, "w+")

p = subprocess.Popen(args, stdout=fastq_file)

p.wait()

fastq_file.close()

#for read counting

#mid_dict_for_counting = read_fastq(fastq_file_name, mid_dict)
```

```
#read_num = count_reads(mid_dict_for_counting)

#print("+++++++ Number of Reads at the beginning %i" %(read_num))

if arguments.quality:

    print("Filtering for quality:")

    print("%i%% of bases need to have min quality score of %i."
%(arguments.quality_percent, arguments.quality_min))

    qualtity_file = "%s%s_quality_filterd_%i_%i.fastq" %(main_folder,
arguments.fileprefix, arguments.quality_min, arguments.quality_percent)

    quality_line = "fastq_quality_filter -q %i -p %i -i %s -o %s"
%(arguments.quality_min, arguments.quality_percent, fastq_file_name, qualtity_file)

    args = shlex.split(quality_line)

    p = subprocess.Popen(args)

    p.wait()

    fastq_file_name = qualtity_file

    #mid_dict_for_counting = read_fastq(fastq_file_name, mid_dict)

    #read_num = count_reads(mid_dict_for_counting)

    #print("+++++++ Number of Reads after quality filter %i" %(read_num))

if arguments.mask:

    print("Masking bases below a qualtiy score of %i" %(arguments.quality_min))

    #mask all nucleotides with N wich have a quality smaller the min_quality

    masked_file = "%s%s_masked_below_%i.fastq" %(main_folder, arguments.fileprefix,
arguments.quality_min)

    mask_line = "fastq_masker -q %i -i %s -o %s" %(arguments.quality_min,
fastq_file_name, masked_file)

    args = shlex.split(mask_line)

    p = subprocess.Popen(args)

    p.wait()

    fastq_file_name = masked_file

    #mid_dict_for_counting = read_fastq(fastq_file_name, mid_dict)

    #read_num = count_reads(mid_dict_for_counting)

    #print("+++++++ Number of Reads after masking %i" %(read_num))

print("Reading fastq file for processing.")


mid_dict = read_fastq(fastq_file_name, mid_dict)

print("Num of MIDs %i" %(len(mid_dict)))

read_num = count_reads(mid_dict)

print("+++++++ Number of Reads for processing %i" %(read_num))

read_num_dict = count_reads_per_mid(mid_dict)

for mid in read_num_dict:

    print("%s Num of UIDs %i" %(mid_to_seq[mid],len(mid_dict[mid])))

    print("+++++++ %s has %i reads" %(mid_to_seq[mid], read_num_dict[mid]))
```

```
if arguments.filter_length:

    print("Filter reads for length.")

    mid_dict = filter_for_length(mid_dict, arguments.min_seq_length,
arguments.max_seq_length)

    read_num = count_reads(mid_dict)

    print("+++++++ Number of Reads after length filter %i" %(read_num))

    read_num_dict = count_reads_per_mid(mid_dict)

    for mid in read_num_dict:

        print("%s Num of UIDs %i" %(mid_to_seq[mid],len(mid_dict[mid])))

        print("+++++++ %s has %i reads" %(mid_to_seq[mid], read_num_dict[mid]))

if arguments.poly_A:

    print("Correcting Poly A stretches")

    mid_dict = correct_for_poly_A(mid_dict, intermed, mid_to_seq)

    read_num = count_reads(mid_dict)

    print("+++++++ Number of Reads after poly A correction %i" %(read_num))

    read_num_dict = count_reads_per_mid(mid_dict)

    for mid in read_num_dict:

        print("%s Num of UIDs %i" %(mid_to_seq[mid],len(mid_dict[mid])))

        print("+++++++ %s has %i reads" %(mid_to_seq[mid], read_num_dict[mid]))

print("Searching for the sequences between primers")

for mid in mid_dict:

    m_dict = mid_dict[mid]

    new_uid_dict = {}

    for uid in m_dict:

        #new list of entries for this uid

        u_list = []

        #print len(m_dict[uid])

        count = 0

        for entry in m_dict[uid]:

            #print "entry"

            #print entry

            #[seq, score]

            count += 1

            #print "Searching for seq %i %i" %(count, len(entry))

            res = find_seq_between_primer(entry[1], entry[3], arguments.primer_forward,
arguments.primer_reverse)

            if res != "NA":

                new_entry = [entry[0], res[0], entry[2], res[1]]

                u_list.append(new_entry)

        if u_list != []:
```

```
            new_uid_dict[uid] = u_list
    mid_dict[mid] = new_uid_dict


read_num = count_reads(mid_dict)
print("+++++++ Number of Reads after primer filtering %i" %(read_num))
read_num_dict = count_reads_per_mid(mid_dict)
for mid in read_num_dict:
    print("%s Num of UIDs %i" %(mid_to_seq[mid],len(mid_dict[mid])))
    print("+++++++ %s has %i reads" %(mid_to_seq[mid], read_num_dict[mid]))
print("Creating a distribution file for every MID")
creating_distribution_files(mid_dict, mid_to_seq)


print("Creating a file for every UID with a coverage greater %i and creating a consensus
sequence" %(arguments.uid_threshold))
#go over all mids seperatly
used_reads = 0
for mid in mid_dict:
    used_reads_mid = 0
    print("currently on MID %s" %(mid))
    result_file = results_folder + arguments.fileprefix + "_" + mid_to_seq[mid] +
"_stats.csv"
    consensus_file = intermed + arguments.fileprefix + "_" + mid_to_seq[mid] +
"_consensus.fa"
    consensus_seq = open(consensus_file, 'w')
    consens_dict = {}
    #results_files[mid] = result_file
    for uid in mid_dict[mid]:
        #print(uid)
        new_uid_fasta = mid_folder[mid] + uid + "_seq_between_primer.fa"
        uid_enties = mid_dict[mid][uid]
        if len(uid_enties) >= arguments.uid_threshold:
            used_reads += len(uid_enties)
            used_reads_mid += len(uid_enties)
            consens = create_align(new_uid_fasta, uid_enties, threshold_dict)
            consens_header = consens[0] + "occured: " + str(len(uid_enties))
            consens_seq = consens[1]
            #starts from end of primer_forward to beginning of primer_reverse
            #immediatley use consensus sequence without gaps
            consensus_seq.write(consens_header + "\n" + consens_seq + "\n")
            consens_dict[consens_header] = [consens_seq, len(uid_enties)]
    print("+++++++ %s has %i reads" %(mid_to_seq[mid], used_reads_mid))
```

```
        print("Create the statistic results for mid %s" %(mid_to_seq[mid]))
        count_consensus_seq(consens_dict, result_file)
        consensus_seq.close()
print("+++++++ Number of Reads used for consensus sequences %i" %(used_reads))
print("Done!")
```

7.4 Acknowledgements

First, I would like to thank Prof Claude P. Muller for entrusting the project to me and for inestimable scientific guidance and opportunities.

Dr I-Na Lu, as my supervisor, I want to thank for her continuous input and advice, for patience and hints.

I especially want to thank Josiane Kirpach for an open ear and open door, unwavering support, advice and friendship during my time in Luxembourg and ever since. In particular I also want to thank her for producing the heatmap.

Sophie Farinelle I would like to thank for introducing me to mouse work, cell culturing and numerous other techniques and for her help and support with the project.

Christina Kiefer I would like to thank for help getting through the mountain of data in writing the pipeline to my specific ideas and demands and a fortunate introduction.

Dr Feng He I would like to thank for his advice and active help with data processing and analysis.

I would like to thank the technicians I could turn to for advice and help with the laboratory work. Special thanks go to Emilie Charpentier and Aurélie Sausy for their support with RNA extraction and quantification and Regina Sinner for the introduction into using the IonTorrent and Bioanalyzer.

Jean-Phillipe Bürckert I want to thank for leaving his protocol on NGS library preparation to me and advice on changing it.

I also would like to thank Martha, Alessia, Wibke, Ni and all my other friends and colleagues for nice chats, after hour fun and enlightening discussions during lunch or coffee break.

I would like to thank the Luxembourg Institute of Health for this opportunity and financial support.

This work has received funding from the European research project FLUTCORE, supported by the European Union 7th Framework Program (FP7) for research, technological development, and demonstration under grant agreement No. 602437.

Danke, Wolfgang, für Deine Unterstützung besonders in der Zeit des Schreibens und dass Du an meiner Seite bist.

Schließlich möchte ich mich noch bei meinen Eltern, Schwestern und Großeltern bedanken. Danke, dass Ihr mich während des Studiums und bei dieser Arbeit unterstützt und immer an mich geglaubt habt!