Sources of HIV-1 Mutations During Infection

Jimmy Chan

Advisor: Doctor Manel Camps

Collaborator: Doctoral Candidate Jay W. Kim

University of California, Santa Cruz

Department of Biomolecular Engineering

Abstract

Human Immunodeficiency Virus (HIV) infections are extremely efficacious because of the multitude of mutations from known (and possibly unknown) sources involved, such as the viral HIV reverse transcriptase (RT) enzyme, and selection due to immune response and antiretroviral drug therapy. The goal of this project is to generate a neutral HIV-1 RT mutation spectrum by restricting the mutation spectrum to only premature stop codon-causing mutations (PSCMs). Given that PSCMs are deleterious, they must have occurred pre-selection; thus a mutation spectrum consisting of only PSCMs is selection-free. Assuming that the main source of non-RT mutations during viral infection is APOBEC, a family of cytidine deaminase enzymes, APOBEC-induced mutations were filtered out by utilizing certain APOBEC mutation signatures. A Python script was written for identifying PSCMs from publicly available HIV genome sequences from the Los Alamos National Laboratory HIV Database. The script ran on three HIV genome studies so far, and of the three studies, study ERP001266 had the most PSCMs. The mutation spectrum of the HIV-1 Pol gene across the three studies and the intergenic mutation spectrum of ERP001266 were both fairly even, as expected. The in vitro and in vivo HIV-1 mutation spectra do not share any noticeable patterns with each other, as expected given the effects of selection and other mutation sources.

Contents

1 Problem Statement

The goal of my design project is to identify distinct mutation patterns in Human Immunodeficiency Virus Type 1 (HIV-1), the most common strain of HIV, through the bioinformatic analysis of HIV genome sequences obtained from publicly available clinical samples. These mutation patterns (or mutation signatures) represent unique mutation sources that contribute to genomic variation in vivo in HIV infected patients. Distinguishing the individual mutation sources should provide significant new insights into how HIV viruses evolve in the presence of an immune response and therapeutic intervention.

Genetic variation is vital for the maintenance of HIV infections by facilitating adaptation to the immune response of the host or to exposure to antiretroviral drugs. Mutations in HIV can be caused by diverse mechanisms, which include replication errors caused by the viral reverse transcriptase (RT) enzyme, host DNA polymerases or host RNA polymerase as well as a variety of genotoxic host defense factors. The relative contribution of each of these sources of genomic variation is unknown. Therefore I aim to identify these different contribution rates by using the unique pattern of mutations created by individual sources of mutation signatures.

My project has three components: (1) bioinformatic handling of 454 HIV genome sequencing reads, (2) large-scale computations involving genome sequence manipulation, and (3) using pre-existing statistical packages to perform blind source separation (BSS), an unsupervised clustering algorithm, to identify mutation patterns corresponding to distinct sources of HIV mutagenesis.

For the first time, this research project aims to deconvolute individual sources of mutation in HIV. Given the connection between genetic instability and the virus' ability to cause disease and develop resistance to antiretroviral drug treatment, identifying sources of mutation has direct clinical applications. These mutation sources will help identify potential targets for inhibition to prevent adaptation or for enhancement to drive viral populations to error catastrophe (death due to the presence of too many mutations). Successfully identifying mutation signatures in HIV will provide new insights into the life cycle of the virus in vivo and into the impact of the genotoxic response of the host. Therefore, a better understanding of the sources of genetic variation offers opportunities for newfound therapeutic intervention and for improved epidemiological monitoring.

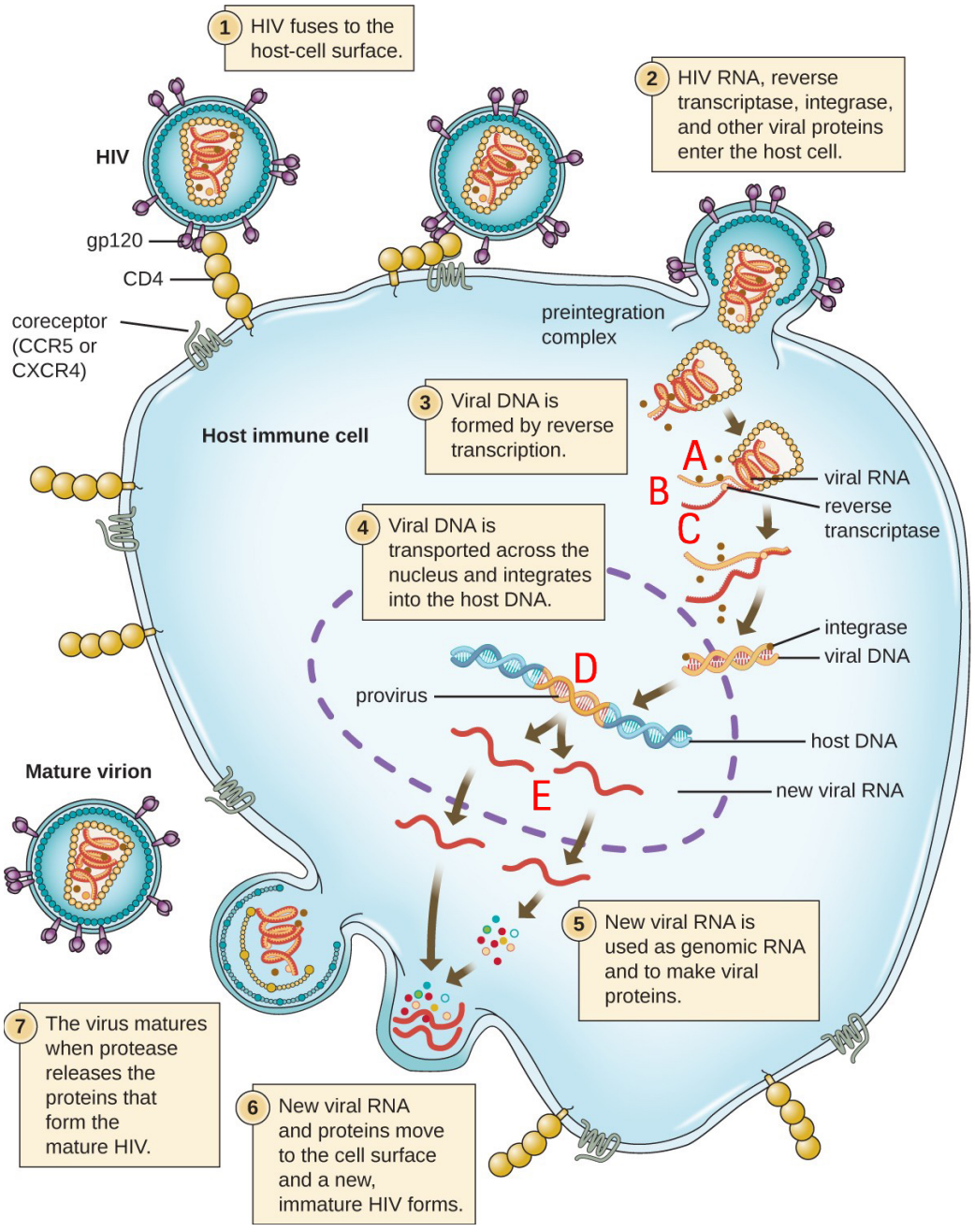
My neutral spectrum will exclude the effects of both positive and negative selection that obscure the unbiased contribution of mutations induced by the viral RT enzyme. RT is responsible for reverse transcribing the viral RNA genomes of HIV viruses into DNA copies. Positive selection occurs when underrepresented mutations conferring drug resistance are selected for, which means certain RT mutations are inaccurately amplified (Leal, Holmes & Zanotto, 2004). Negative (purifying) selection occurs when underrepresented mutations are selected against, which means certain RT mutations are inaccurately diminished (Leal et al., 2004). Selection for or against certain mutations will naturally occur in favor of the survival of the infecting HIV viruses.

Furthermore, selection is also caused by the introduction of antiretroviral drugs, namely non/nucleoside reverse transcriptase inhibitors (NNRTIs/NRTIs) that target and inhibit RT, thereby preventing HIV from making new copies. However, nonsynonymous drug resistance mutations will randomly occur naturally in response to targeted drug therapy. These mutations cause the substitution of amino acids at certain positions on RT, thus changing the shape of the protein. This is evolutionarily advantageous for the virus because the targeted RT enzyme becomes insensitive to the applied drugs, meaning the drugs cannot properly bind to RT given its structural change. These drug resistance mutations are therefore positively selected for because they enable the virus to counter and evade the effects of antiretroviral drug therapy implemented against them.

2 Abbreviations

A	Adenosine
AA	Amino Acid
A3	APOBEC3
AIDS	Acquired Immunodeficiency Syndrome
APOBEC	Apolipoprotein B mRNA Editing Enzyme, Catalytic Polypeptide-like
BSS	Blind Source Separation
C	Cytosine
cDNA	Complementary Deoxyribonucleic Acid
DBM	Double Base Mutation
dC	Deoxycytidine
DNA	Deoxyribonucleic Acid
dU	Deoxyuridine
Env	Envelope
G	Guanine
gDNA	Genomic Deoxyribonucleic Acid
HAXAT	Homopolymer Aware Cross Alignment Tool
HIV	Human Immunodeficiency Virus
IN	Integrase
LANL	Los Alamos National Laboratory
mRNA	Messenger Ribonucleic Acid
ND	Normalization Dividend
NF	Normalization Factor
NNRTI	Non-Nucleoside Reverse Transcriptase Inhibitor
NRTI	Nucleoside Reverse Transcriptase Inhibitor
PR	Protease
PSC	Premature Stop Codon
PSCM	Premature Stop Codon-Causing Mutation
RNA	Ribonucleic Acid
RT	Reverse Transcriptase
SBM	Single Base Mutation
T	Thymine
TBM	Triple Base Mutation
tRNA	Transfer Ribonucleic Acid

3 Background



https://courses.lumenlearning.com/microbiology/chapter/the-viral-life-cycle/

Figure 1. The HIV life cycle begins with a HIV virion attaching to a cell surface receptor of an immune cell and fusing with the cell membrane. Viral contents are released into the cell, where viral enzymes convert the single-stranded RNA genome into DNA and incorporate it into the host genome. The biological host mechanisms then synthesize viral proteins and new copies of the viral genome. Points of mutagenesis have been noted in red letter symbols in the figure, where 1) A represents the mutations caused by RT during the first of the two reverse transcription steps where viral RNA is reverse transcribed to viral single-stranded cDNA, 2) B points represents the mutations on the viral cDNA strand induced by APOBEC, a family of cytidine deaminase enzymes, 3) C represents the mutations caused by RT during the second of the two reverse transcription steps where viral single-stranded cDNA is reverse transcribed to viral single-stranded gDNA, 4) D represents the mutations caused by the host DNA polymerase enzyme, and 5) E represents the mutations caused by the host RNA polymerase enzyme.

3.1 HIV Biology

HIV virions, the infectious form of viruses, consist of the following viral elements required for infectivity: a pair of identical single-stranded RNAs, $tRNA^{Lys,3}$ primers, and various viral proteins (Sundquist & Kräusslich, 2012). The $tRNA^{Lys,3}$ primers are annealed to the primer binding site of the viral RNAs and are responsible for initiating reverse transcription of the RNA strands (Mak & Kleiman, 1997; Sundquist et al., 2012). The viral proteins necessary for HIV virions to effectively infect host cells are the Gag polyprotein, the envelope (Env) protein, and the Gag-Pro-Pol polyprotein consisting of three enzymes: reverse transcriptase (RT), integrase (IN), and protease (PR). Gag plays an integral role in mediating the assembly and budding of new HIV particles (Sundquist et al., 2012). Env proteins provide the lipid envelope for containing the viral contents of HIV virions as well as binding to host cells for initiating HIV infection. RT, IN, and PR each play an essential role in the successful infectivity of HIV virions.

3.2 HIV Life Cycle

HIV requires the biological machinery of a host cell to replicate its viral genome and proteins. The HIV life cycle represents the process whereby new HIV virions are synthesized through the infection of host cells, typically $CD4^+$ T cells (Figure 1). The life cycle begins with a HIV virion fusing to the surface of a host cell via glycoprotein-coreceptor binding. The viral contents then enter into the cytoplasm of the host cell. Two viral RNA strands are reverse transcribed by the viral reverse transcriptase enzyme. The subsequent viral DNA strands are translocated from the cytoplasm to the nucleus and is integrated into the host genome by the viral integrase enzyme. Now called the provirus, the integrated viral DNA is transcribed by the host RNA polymerase. The viral mRNA is either translated into viral polyproteins or used as genomic RNA strands during budding. Two untranslated viral RNA strands and the viral polyproteins begin to assemble near the cell surface and a new HIV particle buds out of the host. This newly budded particle does not reach maturation until the viral protease enzyme cleaves the polyproteins into functioning proteins. It is important to note that errors in various processes of replication and transcription throughout the viral life cycle represent sources of mutations.

3.2.1 Infection

HIV infections begin with a HIV particle's glycoproteins (gp41 and gp120 trans-membrane subunits encoded by the Env gene) binding to corresponding primary receptors located on a host cell, typically a CD4+ helper T cell (Figure 1, step 1) (Wilen, Tilton, & Doms, 2012). Once the HIV virion has successfully attached to a host cell, the viral envelope begins to merge with the host cell membrane, allowing the virion's viral contents to enter the host (Figure 1, step 2). After successful infection, the virion contents remain in the cytoplasmic region of the host cell.

3.2.2 Reverse Transcription

The next immediate stage after HIV infection is the reverse transcription of the viral RNAs into viral double-stranded DNA copies (Figure 1, step 3). Reverse transcription is a two-

step process carried out by the viral RT enzyme and it starts out with RT using the positive-sense, single-stranded RNA to synthesize a negative-sense, complementary DNA (cDNA) strand (Leal et al., 2004). The $tRNA^{Lys,3}$ primers packaged into the HIV virion is necessary for initiating the reverse transcription of the viral RNAs (Mak et al., 1997). The first round of reverse transcription is followed by a second round where the viral cDNA strand is reverse transcribed to create a complementary, positive-sense genomic DNA (gDNA) strand (Leal et al., 2004). This back-to-back reverse transcription process results in a viral double-stranded DNA copy of the viral single-stranded RNA. Extensive mutagenesis occurs at this stage of the HIV life cycle due to the low fidelity of RT (Figure 1, letters A and C).

3.2.3 Integration Into Host Genome

The viral DNA then enters the nucleus and the viral IN enzyme integrates the viral DNA into the host cell's DNA (Figure 1, step 4). The incorporated viral DNA is henceforth referred to as the provirus. From this point on, the provirus is a part of and virtually indistinguishable from the host genome so the provirus can therefore utilize the biological host machinery to synthesize its genome via DNA synthesis. Given the high fidelity of the host cell's DNA polymerase enzyme, it is highly unlikely that the DNA polymerase contributes much to HIV mutagenesis (Figure 1, letter D) (Albertson & Preston, 2006).

3.2.4 Synthesis of Viral Proteins

The provirus is now able to be transcribed into new viral mRNA by the host RNA polymerase, which itself is susceptible to making errors and thus contributes to HIV mutagenesis like the host DNA polymerase. The viral mRNA translocates from the nucleus to the cytoplasm where ribosomes translate the mRNA into viral Gag and Gag-Pro-Pol polyproteins using tRNA anticodons. The virus has achieved all that it needs to produce new copies of itself so it is ready to make its way out of the host cell.

3.2.5 Budding of New HIV Particles

The viral synthesis process has reached its end so the new HIV particles begin to assemble and exit the host cell (Figure 1, step 6). The viral RNA copies synthesized from transcription in the nucleus and the viral polyproteins translated in the cytoplasm assemble at the cell surface. Once assembled, the viral components form a lipid envelope by accumulating viral Env proteins that are within the plasma membrane via the cellular secretory pathway upon the infection stage (Sundquist et al., 2012). The viral components are then ready to bud out of the host cell to form new HIV virion copies. The new virions that bud out are noninfectious (have yet to reach maturation) until the viral PR enzyme processes the Gag and Gag-Pro-Pol polyproteins into individual, functioning proteins (Sundquist et al., 2012). Protease initiates processing during or immediately after the budding stage (Figure 1, step 7). This thereby concludes the life cycle of an average HIV virion. Over time, viral infection leads to AIDS in HIV-infected patients.

3.3 Relationship Between Virulence and Mutagenesis in HIV

The virulence of HIV is highly dependent on the genetic variation that is involved with the HIV genome. There is a need for mutagenesis to a certain extent for HIV to successfully ensure survival and infectivity without changing the genome of the virus so much that there is a loss of function for the viral proteins as a result of hypermutation. Genetic variation allows for HIV to constantly change its genetic makeup so that it is challenging for the immune system or antiretroviral therapeutic intervention to target any part of the virus. The irreversible genomic damage that APOBEC inflicts on HIV is evident by analyzing the plasma viral RNA loads in HIV infected patients and attributing those numbers with HIV RNA virulence (Girerd-Genessay et al., 2016). Given that transcription errors get passed on to future HIV generations like DNA replication errors do, RNA polymerase also plays a role in HIV virulence.

3.4 Sources of Mutations in HIV

3.4.1 Reverse Transcriptase

Unlike most DNA polymerase enzymes, RT lacks proofreading mechanisms so it is highly error prone when performing the two-step reverse transcription of viral RNA (Bakhanashvili, Novitsky, Levy & Rahav, 2005). DNA polymerase associated proofreading mechanisms are intrinsic exonuclease activities that detect misincorporated nucleotide pairings and excise them (Kunkel, 2004). Additional proofreading mechanisms in mammalian cells typically include mechanisms for repairing mismatches such as mismatch repair and nucleotide excision repair. However, given that RT lacks such proofreading mechanisms, its fidelity is significantly lower than that of other DNA polymerases.

3.4.2 APOBEC

Apolipoprotein B mRNA editing enzyme, catalytic polypeptide-like (APOBEC) is a family of DNA cytidine deaminase enzymes found in humans that plays an important role in innate antiviral immunity against exogenous viruses (such as HIV) as well as endogenous retroelements (such as mouse mammary tumour virus) within host cells (Nelson et al., 2013). APOBEC mutagenesis plays a big role in inducing mutation clusters in human cancers such as bladder, breast, and lung cancer (Roberts et al., 2013). Pertaining to HIV specifically, the APOBEC3 subfamily plays a direct role in fighting HIV through induced PSCMs.

3.4.2.1 APOBEC Biology

APOBEC3 (A3) is a subfamily of APOBEC that has been shown to be directly involved with restricting HIV activity (Cuevas, Geller, Garijo, López-Aldeguer, & Sanjuán, 2015). The A3 subfamily consists of seven proteins (A3A, A3B, A3C, A3D, A3F, A3G, and A3H) located on chromosome 22, but only four (A3D/F/G/H) pertain to HIV (Goila-Gaur & Strebel, 2008; Rebhandl, Huemer, Greil & Geisberger, 2015)**.** This subgroup of A3 enzymes converts cytidine residues to uridine residues on the negative-sense cDNA strand reverse transcribed by RT (Gillick et al., 2013). The hydrolytic deamination of deoxycytidine (dC) to deoxyuridine (dU) on the cDNA strand occurs at sequence-context specific target sites.

NSMT	Resulting stop codon[a]		
	TAG	TGA	TAA
TGG A	A3G	A3D/F/H	A3G + A3D/F/H
TGG C	A3G	RT	A3G + RT
TGG G	A3G	A3G	A3G + A3G
TGG T	A3G	RT	A3G + RT
Other (17 codons)	RT	RT	RT

[a]In all cases the indicated mutations can also be produced by the RT in addition to A3 enzymes.

Table 1. This table shows the assignment of mutations in NSMTs (Nonsense Mutation Targets) to A3G, A3D/F/H, or RT. Note: NSMT stands for nonsense mutation target (Cuevas et al., 2015).

Patient	Sex	Age	Infection time (years)	CD4count (cell/μL)	Per-year CD4 count decay rate (cell/μL)[a]	$\log_{10}$ set-point viral load (copies/mL)[b]	Progression rate[c]	HIV-1 mutation rate per base per cell
R3	M	22	1.31	380	321 ± 82	5.4 ± 0.2	Rapid	1.9×10^{-3}
R5	M	37	1.82	432	183 ± 41	5.1 ± 0.1	Rapid	1.6×10^{-3}
R6	M	42	3.10	218	243 ± 39	4.6 ± 0.3	Rapid	3.7×10^{-3}
R7	M	54	1.63	291	459 ± 33	5.1 ± 0.1	Rapid	2.8×10^{-3}
R8	M	25	1.00	338	160	4.4	Rapid	2.4×10^{-3}
R11	M	52	0.96	689	87 ± 25	4.0 ± 0.2	Normal	3.4×10^{-3}
R14	M	31	NA	637	76 ± 22	4.1 ± 0.2	Normal	3.7×10^{-3}
R15	M	25	1.24	446	68 ± 25	4.2 ± 0.1	Normal	12.6×10^{-3}
R4	M	26	2.16	439	48 ± 57	4.3 ± 0.1	Normal	3.2×10^{-3}
R9	M	35	0.65	1146	124 ± 111	3.9 ± 0.1	Normal	4.8×10^{-3}
P6	F	49	20.94	431	37 ± 4	4.3± 0.1	Normal	4.5×10^{-3}

[a] Estimated by linear regression as the slope of CD4 counts against infection time. For R8, although there were few data points for reliably estimating the CD4 decay rate, rapid progression was supported by the observation that the CD4 count dropped to 338 cells/μL within the first year of infection. Full data are shown in S1 Fig, the S1 Data file, and the S2 Data file.

[b] Average log viral load obtained from all available patient samples taken at least one year postinfection but before the onset of treatment and/or symptoms. R8 had only one available viral load determination. For R9, a viral load determination taken 329 d after infection was included in the calculation to have at least three data points. Full data are shown in S2 Fig, the S1 Data file, and the S2 Data file.

[c] Rapid progressors were defined as patients showing a CD4 count decay rate greater than 150 cell/μL per y.

doi:10.1371/journal.pbio.1002251.t001

Table 2. This table contains patient clinical data and mutation rate summaries for 11 HIV-1 infected patients (Cuevas et al., 2015).

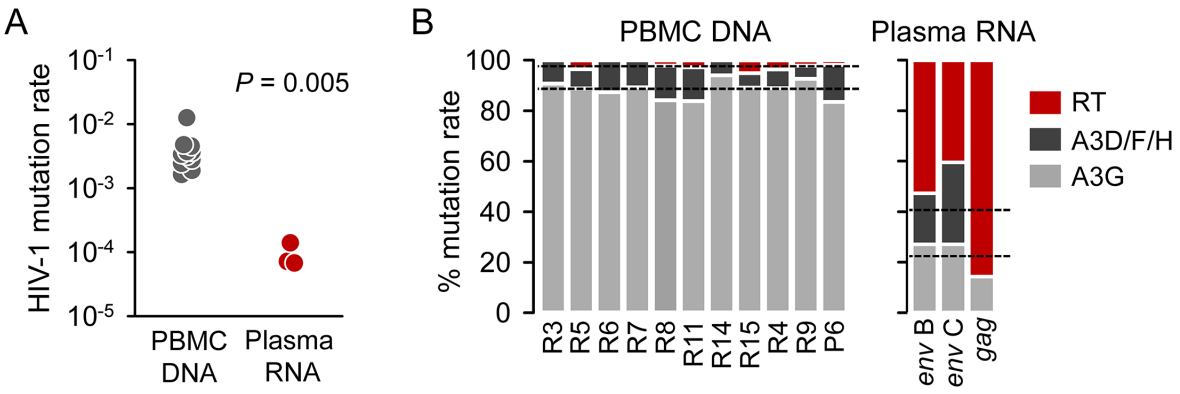


Figure 2. This shows the comparison of HIV-1 mutation rates of RT and A3D/F/G/H in the peripheral blood mononuclear cell (PBMC) DNA and plasma RNA 11 HIV-1 infected patients (Cuevas et. al., 2015).

3.4.2.2 APOBEC Mutation Signatures

A3D/F/G/H targets all Tryptophan amino acids (TGG codons) encoded in the cDNA strand. The sequence-context specificity is provided by the nucleotide that is encoded right after a TGG. The result of cytidine deamination on TGG codons is the base mutation from guanosine (G) to adenosine (A) on the complementary positive-sense gDNA strand. These G to A (GA) base mutations are detrimental for the infecting HIV virions because they impose PSCs throughout the viral DNA reverse transcribed in the host cell. APOBEC can induce PSCs (TAG, TGA or TAA) on the gDNA strand as a result of a GA mutation at the second, third or both the second and third bases, respectively (refer to Table 1). Since they frequently result in PSCs, A3D/F/G/H mutations undergo a purifying selection. Table 1 shows the sequence-context specific mutation signatures for A3D/F/G/H, laying out the ways in which APOBEC induces mutations leading to PSCs into the HIV genome. A SBM is sufficient to generate a TAG or TGA; however, two DBMs are required to generate a TAA (refer to Table 1). TAG is generated by A3G when there is a TGG codon followed by any of the four nucleotides; A3G causes a GA base mutation of the second base in TGG (refer to Table 1). TGA is generated by A3D/F/H when there is a TGG codon followed specifically by an A; A3D/F/H causes a GA base mutation of the third base in TGG (refer to Table 1). TAA is generated by DBMs rather than SBMs (refer to Table 1). For example, TAA can be generated from a TGG by a A3G-induced GA SBM of the second base and a A3D/F/H-induced GA SBM of the third base if the TGG codon is followed by an A.

3.4.2.3 APOBEC Contribution to HIV Mutagenesis

A recent study on the mutation rate of HIV-1 revealed that A3 contributed to 98% of the mutagenesis in the viral peripheral blood mononuclear cell (PBMC) DNA of their sample of 11 HIV infected patients (see Table 2) (Cuevas et al., 2015). Table 2 provides clinical data and a mutation rate summary for each patient. This study produced insightful information about the presence of A3D/F/G/H produced in HIV infected samples and demonstrated that A3 enzymes are a significant source of mutagenesis during HIV infections. It is important to note that RT could have also been responsible for causing the same GA base mutations as A3 enzymes are known to cause in the infected blood samples. However, given that the base mutations sequenced from the patients were found to be sequence-context dependent and RT is not known to have such a sequence-context motif, the vast majority of the GA base mutations found can be confidently attributed to A3 enzymes rather than RT (Cuevas et al., 2015).

3.4.2.4 APOBEC Mutation Rate in DNA vs RNA

The study discovered significantly different mutation rates between the sequenced DNA and RNA from their patient samples. Figure 2A shows the direct comparison of HIV-1 mutations rates between each patient's PBMC DNA and plasma RNA. The mutation rates found in the RNA samples were significantly (44 times) lower than that of the DNA samples (Cuevas et al., 2015). This difference of approximately two orders of magnitude indicates that a sizable proportion of viral DNA is lethally mutated to the point where they are unable to reach plasma due to the excessive amount of hypermutations (Cuevas et al., 2015). Figure 2B shows a clear

percentile comparison of the mutation rates determined from each patient's PBMC DNA and plasma RNA. The two graphs depict the clear difference in mutation rates traced back to RT, A3D/F/H, and A3G. For the PBMC DNA graph, it was determined that A3G and A3D/F/H contributed to 98% (88.4 ± 1.1% and 9.7 ± 1.1%, respectively) of the mutation rate while only 2.0 ± 0.54% was attributed to RT (Cuevas et al., 2015). For the plasma RNA graph, there is a noticeable disparity between the mutation rates caused by RT, A3D/F/H, and A3G (59.7 ± 13.5%, 17.6 ± 9.5%, and 22.8 ± 4.2%, respectively) (Cuevas et al., 2015). However, the RT-PCR step required to sequence the plasma RNA necessarily equates to low fidelity so the mutations rates for the viral RNA samples may be inaccurate to a certain degree (Cuevas et al., 2015).

3.4.3 Host Machinery

3.4.3.1 DNA Polymerase

DNA synthesis is well known to be highly accurate in most organisms, eukaryotic or prokaryotic, as it should be given its permanent effect on the genome of an organism. Minor genomic changes can have significant and sometimes even detrimental effects so it is extremely important that DNA polymerase has high replication fidelity. Although DNA polymerase makes approximately one error every 10^5 nucleotides incorporated during DNA synthesis, its inherent 3′→5′ exonucleolytic proofreading mechanisms and post-replication mismatch repair fix 99% of these errors (Albertson et al., 2006; Pray, 2008). Therefore, it is highly implausible that DNA polymerase poses a significant contribution to the overall mutagenesis of HIV.

3.4.3.2 RNA Polymerase

RNA polymerase mediates transcription but little is known about the inner biological mechanisms of the enzyme in regards to its fidelity and role in fixing transcription errors (Sydow & Cramer, 2009). The mutation rate for RNA polymerase has been approximated to be 10^{-3} - 10^{-5} per nucleotide (Figure 1, step 5) (James, Gamba, Rockwell & Zenkin, 2017; Wang, Opron, Burton, Cukier & Feig, 2015). Therefore RNA polymerase causes more mutations than DNA polymerase does. Given that transcription errors can be passed down to future generations of HIV in the form of mutated viral RNA copies or impaired expression of viral proteins, it is undeniable that RNA polymerase contributes to the genetic variation of the HIV genome.

4 Work Accomplished

4.1 Objective

My objective for this project is to identify a mutation signature for HIV-1 RT by removing the effects of selection and known non-RT sources of mutations. This will involve writing a programming script in Python that keeps track of nucleotide base mutations among the DNA sequences of infected HIV-1 patients. I will be removing the effects of selection from my mutation studies by only considering PSCMs rather than all base mutations. Furthermore, I will remove the mutations from a major source of mutations, APOBEC, to obtain a more neutral mutation spectrum of HIV-1 RT.

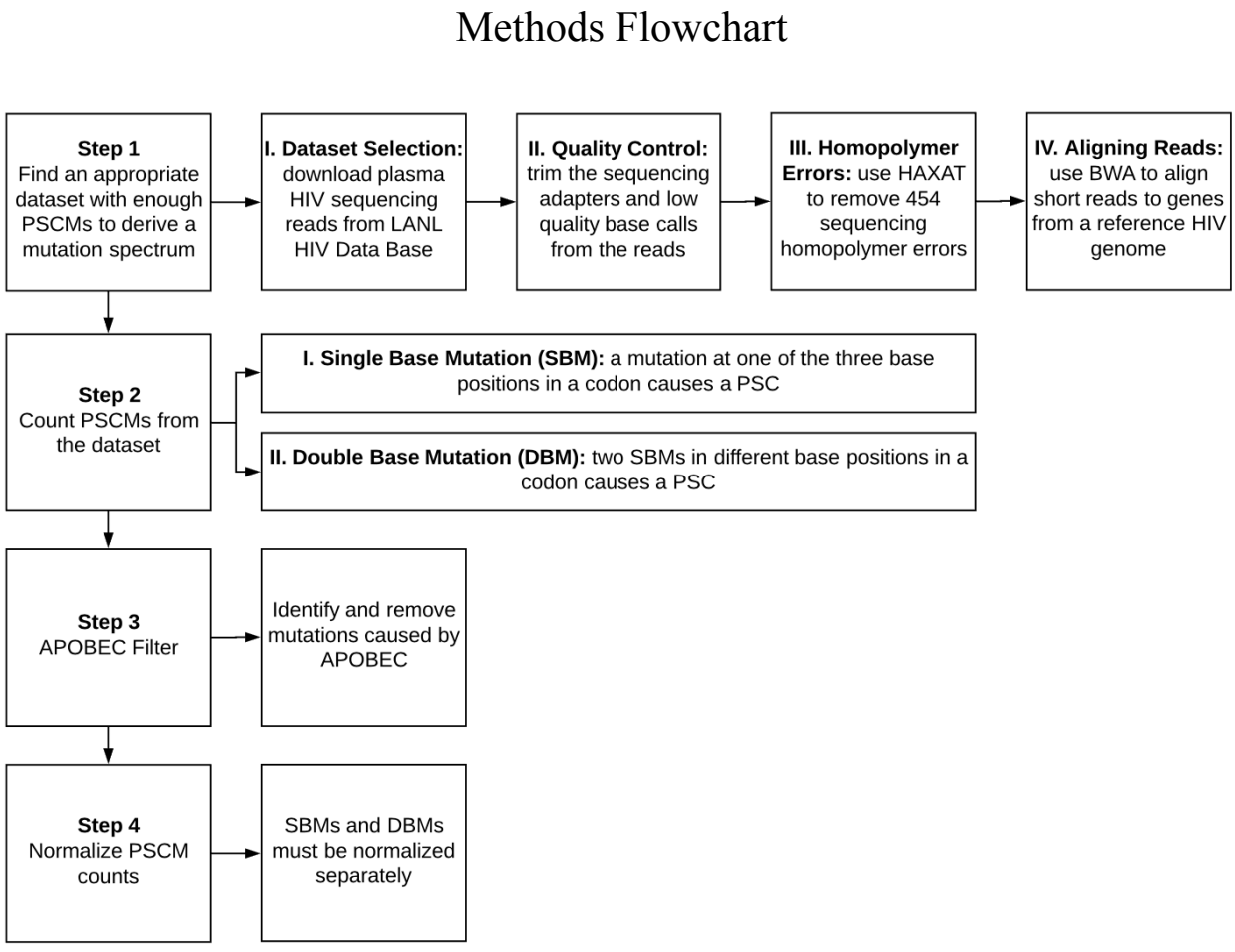


Figure 3. This flowchart shows the four major steps involved in my methods. Step 1 involves finding an appropriate dataset, step 2 involves counting and recording the PSCMs in the given dataset, step 3 involves filtering out APOBEC-induced GA base mutations, and step 4 involves normalizing the recorded PSCM counts. Refer to Figure 4 for the corresponding functions used to carry steps 2, 3, and 4 of my methods.

Functions Flowchart

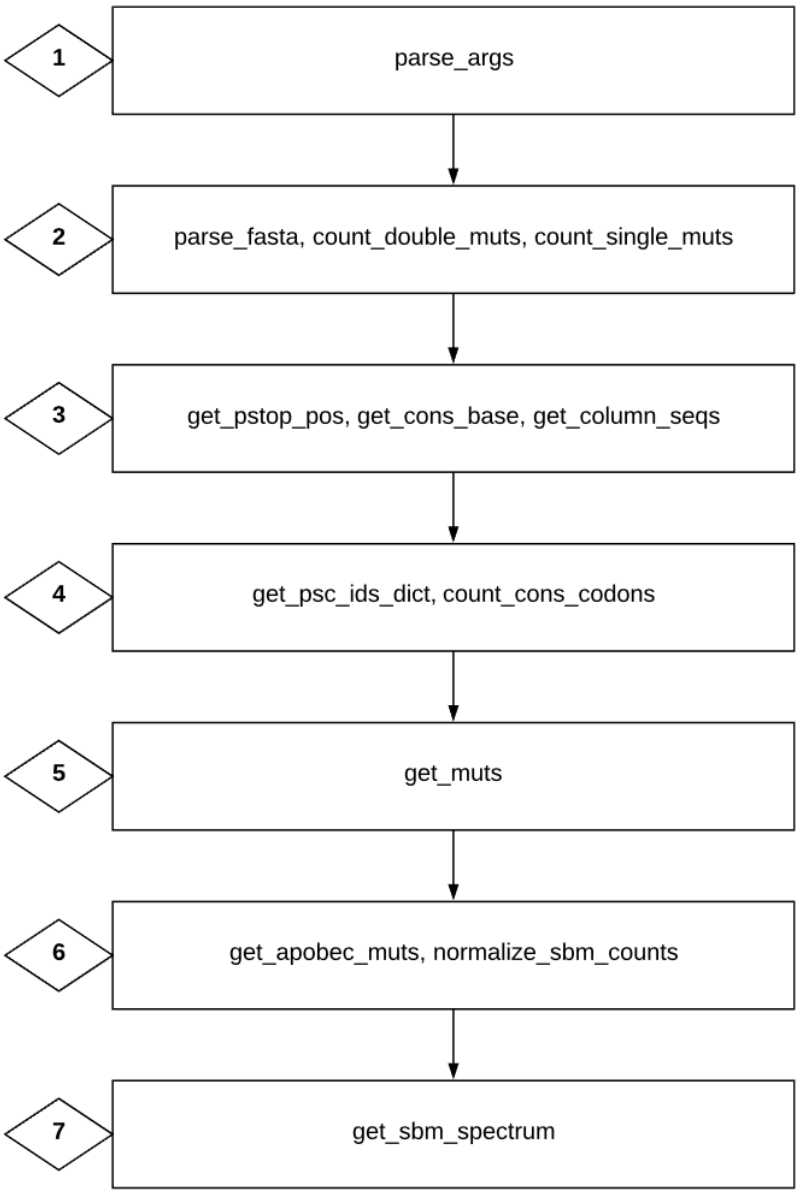


Figure 4. This flowchart shows all the functions I created to carry out my methods (refer to Figure 3). The functions in boxes 1, 2, and 3 as well as the get_psc_ids_dict() function in box 4 and the get_muts() function in box 5 were used to count PSCMs in a given HIV FASTA-format file. The get_apobec_muts() function in box 6 was used to filter out APOBEC-induced GA base mutations from the PSCM counts. The count_cons_codons() function in box 4 and the normalize_sbm_counts() function in box 6 were used to normalize the subsequent PSCM counts. Lastly, the get_sbm_spectrum() function in box 7 was used to map out the mutation spectrum of the normalized PSCM counts.

4.2 Methods

I performed large-scale computations involving genome sequence manipulation by running a custom Python script on three publicly available HIV genome studies (ERP001257, ERP001266, and SRP002483) from the LANL-HIV databases to obtain my neutral HIV-1 mutation spectrum. My project involved four major components: 1) accumulating the PSCMs from FASTA files, 2) filtering out PSCMs induced by APOBEC3, 3) normalizing the remaining PSCMs, and 4) generating a neutral HIV-RT mutation spectrum. Refer to Figure 3 for a flow-chart of the methods and Figure 4 for a flowchart of the functions I created to carry out my project.

4.2.1 Parsing HIV FASTA-Format Files

The first step in implementing my script for the project was to create a function that would parse input FASTA-format files containing HIV genome sequences. I wrote a function called parse_fasta() that recorded the IDs (unique patient identification numbers) and corresponding DNA sequences from a given FASTA-format file as the keys and values in a dictionary, respectively. A dictionary in Python is an array used to store associated data—namely unique keys corresponding to values. In order to find the PSCMs, I utilized four primary functions: 1) get_pstop_pos(), 2) get_psc_ids_dict(), 3) get_column_seqs() and 4) get_muts(). The get_pstop_-pos() function created a list containing unique amino acid (AA) positions where PSCs are located throughout a given FASTA-format file. This function works by iterating through each codon of every sequence in an input FASTA-format file, identifying PSCs, and saving the corresponding AA positions in a list called psc_pos.

This resulting list was used as a parameter (input variable) in the get_psc_ids_dict() function. The get_psc_ids_dict() function incorporated the psc_pos list elements as keys for a dictionary called psc_ids, where the corresponding value for each key was a list of the IDs that had sequences containing PSCs located at a the given key (distinct psc_pos list element). For each element in psc_pos, I went through each sequence in a given FASTA-format file and checked to see if that sequence had a PSC at that given position. If a sequence contained a PSC, then I would save that sequence's corresponding ID as an element of a distinct list that acts as the value corresponding to the given key (psc_pos element) for psc_ids. For example, for element 100 in psc_pos, the get_ids key would be 100 and the corresponding value would be a list of IDs of sequences contained a PSC at AA position 100 in a given FASTA-format input file. The psc_pos list and psc_ids dictionary were both used as parameters in the get_muts() function.

4.2.2 Determining PSCMs

The get_muts() function finds either single base mutations (SBMs) or double base mutations (DBMs) in any given sequence within a FASTA file. SBMs are mutations at a single base position of a codon that cause a PSC, whereas DBMs are two SBMs at different base positions of a codon that cause a PSC. For example, a TCA that is mutated to a TGA is caused by a CG SBM, and a TCC that is mutated to a TGA is caused by two DBMs, CG and CA. The get_muts() function worked by iterating through each element in the psc_pos list created by the get_pstop_pos() function. For each AA position, the function implemented a check to see which

FASTA IDs have sequences containing a PSC at that AA position. This check utilized the psc_ids dictionary created by the get_psc_ids_dict() function. For a given ID that had a PSC in its sequence, the SBM or DBMs that caused the PSC would be identified from the sequence. This was done with the help of a function called get_column_seqs(), which returns a string of bases all residing in the same base position across each sequence in a FASTA-format input file.

The get_muts() function used the consensus base at a given base position to identify what the mutating base was. For example, given a premature TAA stop codon at AA position 400, get_muts() would call get_column_seqs at each of the three base positions (1198 to 1200) corresponding to that AA position. From there, get_muts() would determine what the consensus base is at each of the three base positions by counting the frequency of each base and designating the consensus base as the one with a frequency of at least 90% throughout all the sequences at the given base position. I implemented this arbitrarily high threshold of 90% to ensure the validity of the consensus bases after getting advice from my principal investigator, Doctor Manel Camps and doctoral candidate Jay W. Kim. The get_muts() function would then iterate through each of the three consensus bases and compare it to the base corresponding to the same base position of the designated PSC. A PSCM is determined when a consensus base does not match the corresponding base of the PSC. For example, if the consensus base at base position 1198 was a C, then get_muts() would detect this difference and record a CT PSCM. This would be considered a SBM if that was the only difference in bases among the three base positions, 1198, 1199, and 1200. However, if there was another discrepancy between a consensus base and a PSC base, then that discrepancy would be considered an additional base mutation, meaning the PSC was caused by two DBMs rather than a SBM. I did not implement a check for triple base mutations because the likelihood of a triple base mutation is extremely low. After consulting the possibility of TBMs with Doctor Manel Camps and doctoral candidate Jay W. Kim, they also agreed that it would not be worth including as a PSCM check in the get_muts() function given its extremely low probability of occurring. Therefore I only included checks for SBMs and DBMs in the get_muts() function.

4.2.2.1 Debugging get_muts() Function

I noticed that after running the script on several FASTA files, get_muts() was not outputting many mutations. This seemed strange to me so I decided to perform some unit tests on the get_muts() function to identify whether this was due to a bug in the get_muts() function or one of the parameter functions of the get_muts() function. I created a simple test FASTA-format file for my Python script to run on. The file contained a few PSCMs at known AA positions. I included print statements throughout my get_muts() function to figure out where exactly the function was led astray. After running the script on my test file several times and analyzing the print statements, I figured out that the problem in the code was a simple calculation error. I was off by a single base position when referring to potential the PSC AA positions of the psc_pos list. This caused get_muts() to inaccurately look for PSCs while there was a frame shift of a single base. After fixing this error and running my script on my test file, the get_muts() function was able to work properly and detect all the PSCs present in my sample FASTA-format file.

4.2.3 Filtering Out A3 Mutations

After gathering all the PSCMs from a given FASTA file via the get_muts() function, my next objective was to filter out the GA base mutations caused by APOBEC3. I created a function called get_apobec_muts() to identify and remove A3-induced GA base mutations from the collection of PSCMs created from get_muts(). This function utilized the sequence-context specific mutation signatures for A3G and A3D/F/H in order to identify the A3-induced mutations (see Table 2). After finding all of the A3-induced mutations across all sequences in a given FASTA file, I stored the A3 mutation counts (separated by A3 enzyme) in a dictionary to keep track of how many GA mutations were being induced by APOBEC3.

4.2.3.1 Debugging get_cons_base() Function

After running my script on a relatively large dataset, I was not getting any mutations caused by A3. This seemed highly irregular to not get a single A3 mutation so I went through the code and backtracked from the get_apobec_muts() function. To unit test get_apobec_muts(), I created a simple FASTA-format test file that included A3-induced sequence-context specific GA mutations. By incorporating print statements throughout the get_apobec_muts() function and running the script on the test file, I was able to deduce that my get_cons_base() function used in get_apobec_muts() contained a problem.

The get_cons_base() function returns the consensus base for a specific base position. When testing the function with added print statements, I noticed less consensus bases outputted than there should have been because the function was including dashes (non-nucleotide characters) in the sequences when calculating the consensus base. Incorporating dashes into the consensus base calculation was the wrong to do because I realized that in many cases, the consensus base for a given base position was a dash, not a base character. Thus, consensus bases were not being recorded for many base positions. This meant that the get_apobec_muts() function was bypassing many base positions when considering them for SBMs. Removing the dashes as part of the consensus base finder allowed for a considerable increase in consensus base calls and subsequently more PSCMs recorded for the mutation spectrum. I decided it was necessary to implement the same 90% threshold requirement for calling the consensus base as we have done for all other base mutation calls in other functions. This way, we can be more confident in the consensus base outputted by get_cons_base(), even with the considerably smaller sample size with the majority of characters aligned per base position being dashes.

4.2.4 Normalization

4.2.4.1 First Normalization Approach Based on Nucleotide Frequency

Following the APOBEC filter, I went on to normalize the remaining selection-free base mutations. I initially normalized the base mutation counts by dividing the counts by the number of times the mutated base (base responsible for inducing a PSC) is seen across the three possible PSCs. Across the three types of stop codons, A appears four times, T appears three times and G appears two times. Therefore xA mutation counts were divided by 4, xT mutation counts by 3, and xG mutation counts by 2, where x is the base mutating to A, T or G, respectively, to cause a PSC. After consulting this normalization method with Doctor Manel Camps and doctoral

candidate Jay W. Kim, I found out that this is not the most accurate way to normalize the mutation counts. This normalization method was an oversimplification of the likelihoods of each type of base mutation occurring based on the frequency of nucleotides in each PSC.

Base Mutation	TAA	TAA-NF	TAG	TAG-NF	TGA	TGA-NF
AC	0	N/A	0	N/A	0	N/A
AG	0	N/A	0	N/A	0	N/A
AT	1	2	1	1	1	1
CA	2	1	1	1	1	1
CG	0	N/A	1	1	1	1
CT	1	2	1	1	1	1
GA	0	N/A	1	1	1	1
GC	0	N/A	0	N/A	0	N/A
GT	1	2	1	1	1	1
TA	2	1	1	1	1	1
TC	0	N/A	0	N/A	0	N/A
TG	0	N/A	1	1	1	1
ND	2	N/A	1	N/A	1	N/A

Table 3. This table shows the corresponding ND values for each PSC induced by SBMs and the PSC-specific NF values for each type of base mutation that has a chance of inducing that specific PSC.

Base Mutation	TAA	TAA-NF	TAG	TAG-NF	TGA	TGA-NF
AC	0	N/A	0	N/A	0	N/A
AG	0	N/A	5	6	5	6
AT	6	2	6	5	6	5
CA	12	1	6	5	6	5
CG	0	N/A	6	5	6	5
CT	6	2	6	5	6	5
GA	12	1	5	6	5	6
GC	0	N/A	0	N/A	0	N/A
GT	6	2	6	5	6	5
TA	12	1	6	5	6	5
TC	0	N/A	0	N/A	0	N/A
TG	0	N/A	6	5	6	5
ND	12	N/A	30	N/A	30	N/A

Table 4. This table shows the corresponding ND values for each PSC induced by DBMs and the PSC-specific NF values for each type of base mutation that has a chance of inducing that specific PSC.

4.2.4.2 Second Normalization Approach Based on PSCM Ratios

My second approach involved normalizing the ratios between them based upon how likely each specific type of SBM or DBM occurs. The new normalization method involved three steps: 1) calculate the "normalization dividend" (ND) of a base mutation, 2) calculate the "normalization factor" (NF) of a base mutation and 3) normalize the base mutation counts using the NF. The ND is the least common dividend that applies to the number of chances each type of base mutation has of occurring to induce a certain type of PSC. Tables 3 and 4 show the corresponding ND values for each PSC induced by SBMs or DBMs, respectively, as well as the PSC-specific NF values for each type of base mutation. For example, for SBMs that induce TAA PSCs, the TAA-specific ND is 2 because the AT, CT and GT base mutations have one chance of occurring while the CA and TA base mutations have two chances of occurring to induce a TAA stop codon. Thus, the least common dividend among these base mutations is 2 because 2 is the smallest dividend that applies for all the base mutations. After calculating the PSC-specific ND for each PSC for SBMs and DBMs, the NF can be calculated by dividing each PSC-specific ND by the number of chances a base mutation has to occur. Referring back to the earlier example, the NF for AT, CT and GT base mutations would be 2 and the NF for CA and TA base mutations would be 1. I normalized the SBM and DBM counts by dividing the recorded (pre-normalized) base mutation counts by their corresponding NF values. For example, I would normalize AT PSCMs by dividing the AT mutation count by 2.

Base Mutation	SBM-TAA	DBM-TAA	SBM-TAG	DBM-TAG	SBM-TGA	DBM-TGA
AC	X	X	X	X	X	X
AG	X	X	X	✔	X	✔
AT	✔	✔	✔	✔	✔	✔
CA	✔	✔	✔	✔	✔	✔
CG	X	X	✔	✔	✔	✔
CT	✔	✔	✔	✔	✔	✔
GA	X	✔	✔	✔	✔	✔
GC	X	X	X	X	X	X
GT	✔	✔	✔	✔	✔	✔
TA	✔	✔	✔	✔	✔	✔
TC	X	X	X	X	X	X
TG	X	X	✔	✔	✔	✔
Total	5	6	8	8	8	8

Table 5. This table compares the base mutations that induce each of the three types of (premature) stop codons—TAA, TAG, and TGA—based upon whether they are a SBM or DBM. The columns are paired off by stop codon type, with the left column of the pair being SBMs and the right column of the pair being DBMs. The discrepancies between SBM- and DBM-induced stop codons are 1) AG mutations that cause the TAG and TGA stop codons and 2) GA mutations that cause the TAA stop codon.

4.2.4.3 Normalization of SBMs with DBMs is Not Possible

After developing this new normalization method, I decided to set aside the DBMs for now because I realized DBMs cannot be normalized with the SBMs given that there are discrepancies in the types of base mutations that are possible among the two categories of base mutations. Specifically, SBMs do not allow for AG base mutations that cause TAG and TGA PSCs or GA mutations that cause TAA PSCs but DBMs do allow for such PSCMs (see Table 5). Table 5 compares the base mutations that induce each of the three types of (premature) stop codons—TAA, TAG, and TGA—based upon whether they are a SBM or DBM to show the discrepancies between SBMs and DBMs. The table columns are paired off by stop codon type, with the left column of the pair being SBMs and the right column of the pair being DBMs. The discrepancies between SBM- and DBM-induced stop codons are 1) AG mutations that cause the TAG and TGA stop codons and 2) GA mutations that cause the TAA stop codon. I concluded that I could not simply aggregate the corresponding base mutation counts between the SBM and DBM categories because of these unavoidable differences. I therefore decided to set aside the DBMs for now and focus on SBMs since SBM counts are more abundant than DBM counts so they contribute more base mutations to the mutation spectrum.

Base Mutation	TAA	TAG	TGA
AC	X	X	X
AG	X	X	X
AT	✔	✔	✔
CA	✔	✔	✔
CG	X	✔	✔
CT	✔	✔	✔
GA	X	✔	✔
GC	X	X	X
GT	✔	✔	✔
TA	✔	✔	✔
TC	X	X	X
TG	X	✔	✔
Total	5	8	8

Table 6. This table shows the non 1:1:1 ratio between the CG, GA, and TG base mutations in regards to what PSC they can induce.

4.2.4.4 Normalization of Only TAG and TGA PSCMs

I also decided to only consider TAG- and TGA-induced PSCs when calculating the neutral mutation spectrum. The reason is because TAG and TGA provide a broader mutation spectrum, given that they involve eight types of PSCMs—AT, CA, CG, CT, GA, GT, TA, TG—, whereas TAA only involves five types of PSCMs—AT, CA, CT, GT, TA—(see Table 6). Table 6 shows the non-1:1:1 ratio between the CG, GA, and TG base mutations in regards to what PSC

they can induce. The TAA stop codon therefore limits the scope of the mutation spectrum so it was better to leave it out and only use TAG and TGA stop codons. Also, the TAG and TGA have normalized ratios meaning they have a 1:1:1 ratio among its three incorporated bases. However, TAA has a 1:2 ratio regarding its T:A ratio, meaning it does not have equal representation. This makes it impossible to add the SBM counts from all three stop codon groups without introducing bias towards base mutations to A and against base mutations to G. Therefore, only using TAA and TGA stop codons allows for the uniform aggregation of base mutation counts.

SBM	Mutating Codons (MCs)	TAA-MCs	TAG-MCs	TGA -MCs
CA	TAC, TCA, TCG, TGC	TAC, TCA	TCG	TGC
GA	TGG	——	TGG	TGG
TA	TAT, TGT, TTA, TTG	TAT, TTA	TTG	TGT
CG	TAC, TCA	——	TAC	TCA
TG	TAT, TTA	——	TAT	TTA
AT	AAA, AAG, AGA	AAA	AAG	AGA
CT	CAA, CAG, CGA	CAA	CAG	CGA
GT	GAA, GAG, GGA	GAA	GAG	GGA

Table 7. This table shows the list of corresponding consensus mutating codons (MCs) for each type of SBM that can induce a PSC. The MCs are also separated by the premature stop codon that each type of SBM causes. Note that not all corresponding mutating codons were used in normalizing the SBM counts given that only TAG and TGA PSCMs were considered and normalized for the output mutation spectrum in the results section.

$$Normalization(SBM\ type\ XY) = \frac{XY\ Count}{frequency(mutating\ codon/s)}$$

Formula 1. This is a formula for normalizing SBM counts for each type of SBM, where 1) Normalization(SBM type XY) is the normalization of a certain type of SBM (denote this arbitrary SBM type as XY, where base X mutates to base Y), 2) XY count is the total number of XY mutations found in a given FASTA-format input file, and 3) frequency(mutating codon/s) is the frequency of the codons that may mutate to a premature stop codon due to a XY base mutation.

4.2.4.5 Third Normalization Approach Based on Codon Frequency

The count_cons_codons() function keeps track of the frequency of every consensus codon at every AA position of a FASTA file in a dictionary. The dictionary key is the consensus codon and the value is the frequency of that codon at any AA position throughout the given FASTA file. This resulting dictionary is used to normalize the SBM spectrum in the normalize_sbm_counts() function. My current normalization method involves dividing each type of SBM count by the probability of that PSCM occurring (see Table 7 & Formula 1). Formula 1 shows the general

normalization formula I implemented when normalizing the single base PSCMs. The denominator is the frequency of corresponding consensus mutating codons that a certain type of SBM count could mutate into a PSC. Table 7 shows the list of corresponding consensus mutating codons for each type of SBM that can cause a PSC. For example, to normalize the CG SBM count, the denominator would be the frequency of consensus TAC and TCA codons because TAC can mutate to TAG and TCA to TGA. Thus, each type of SBM has a unique normalization divisor.

4.2.5 Estimated Neutral Mutation Spectrum

Once the normalization step was completed, I was able to apply the get_muts(), get_apobec_muts() and normalize_sbm_counts() functions to create a neutral RT mutation spectrum. I created a function called get_sbmORdbm_spectrum() that uses the PSCMs found via get_muts() for the foundation mutation spectrum. I then removed the source of A3 mutations by using the get_apobec_muts() function to identify A3 caused mutations and removing them from the spectrum. From there, I normalized the neutral spectrum using the normalize_sbm_counts() function. I then ended up with an estimate neutral RT mutation spectrum. The get_sbm_-spectrum() function therefore utilizes all the main functions pertaining to the first three steps of my project—1) get_muts() for accumulating the PSCMs from FASTA files, 2) get_apobec_-muts() for filter out base mutations induced by APOBEC3, and 3) normalize_sbm_counts() for normalizing the PSCMs.

4.3 Results

A - Figure 5	Study	AT	CA	CG	CT	GA	GT	TA	TG	Total
	ERP001266	121	8	26	213	1259	304	77	194	2202
	ERP001257	0	0	0	1	8	1	0	0	10
	SRP002483	1	0	0	8	36	0	0	1	46
B - Figure 6	Study	AT	CA	CG	CT	GA	GT	TA	TG	Total
	ERP001266	121	8	26	213	1107	304	77	194	2050
	ERP001257	0	0	0	1	3	1	0	0	5
	SRP002483	1	0	0	8	15	0	0	1	25
C - Figure 7	Gene	AT	CA	CG	CT	GA	GT	TA	TG	Total
	pol	121	8	26	213	1259	304	77	194	2202
	gag	124	18	25	228	804	321	82	132	1734
	env	60	14	20	73	525	74	71	199	1036
D - Figure 8	Gene	AT	CA	CG	CT	GA	GT	TA	TG	Total
	pol	121	8	26	213	1107	304	77	194	2050
	gag	124	18	25	228	709	321	82	132	1639
	env	60	14	20	73	437	74	71	199	948
E - Figure 9	Study	AT	CA	CG	CT	GA	GT	TA	TG	Total
	In-vitro	3	38	1	142	259	4	52	14	513
	ERP001266	305	40	71	514	2588	699	230	525	4972

Table 8. Base mutation counts for Figures 5, 6, 7, 8, and 9 correspond to A, B, C, D, and E respectively.

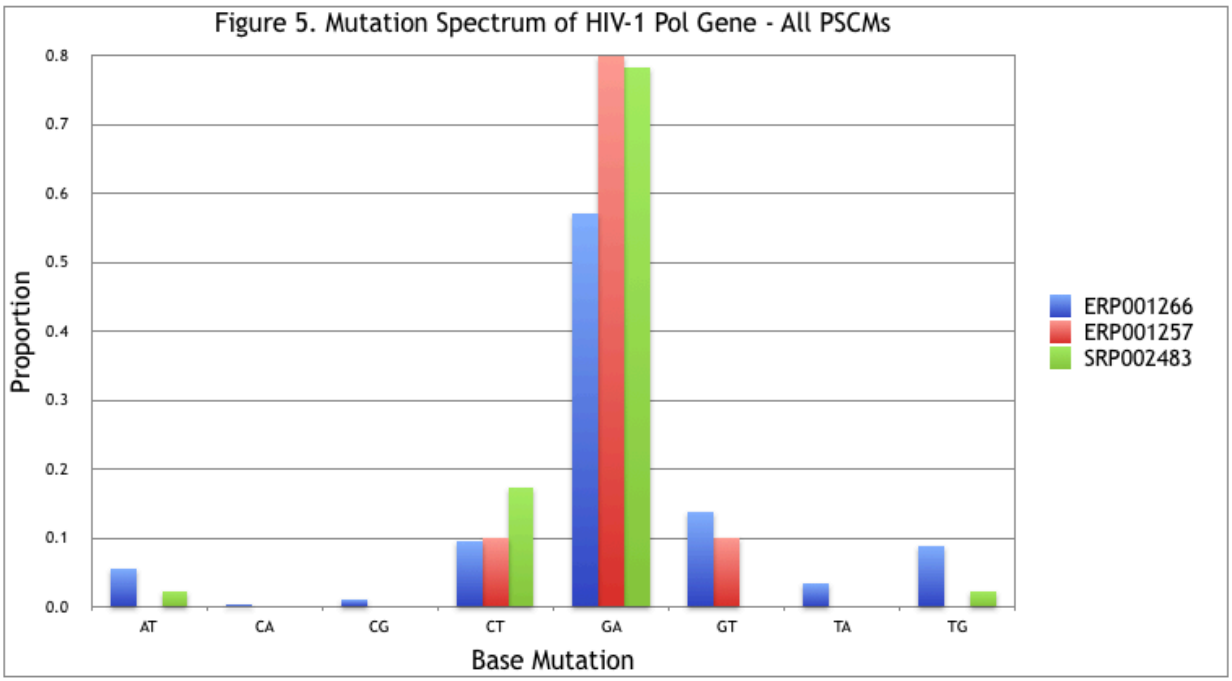


Figure 5. Mutation spectrum derived from only PSCMs found in the HIV-1 Pol gene. APOBEC mutations have not been filtered out. Refer to Table 8A for data.

Figure 6. Mutation spectrum derived from only PSCMs found in the HIV-1 Pol gene. APOBEC mutations have been filtered out. Refer to Table 8B for data.

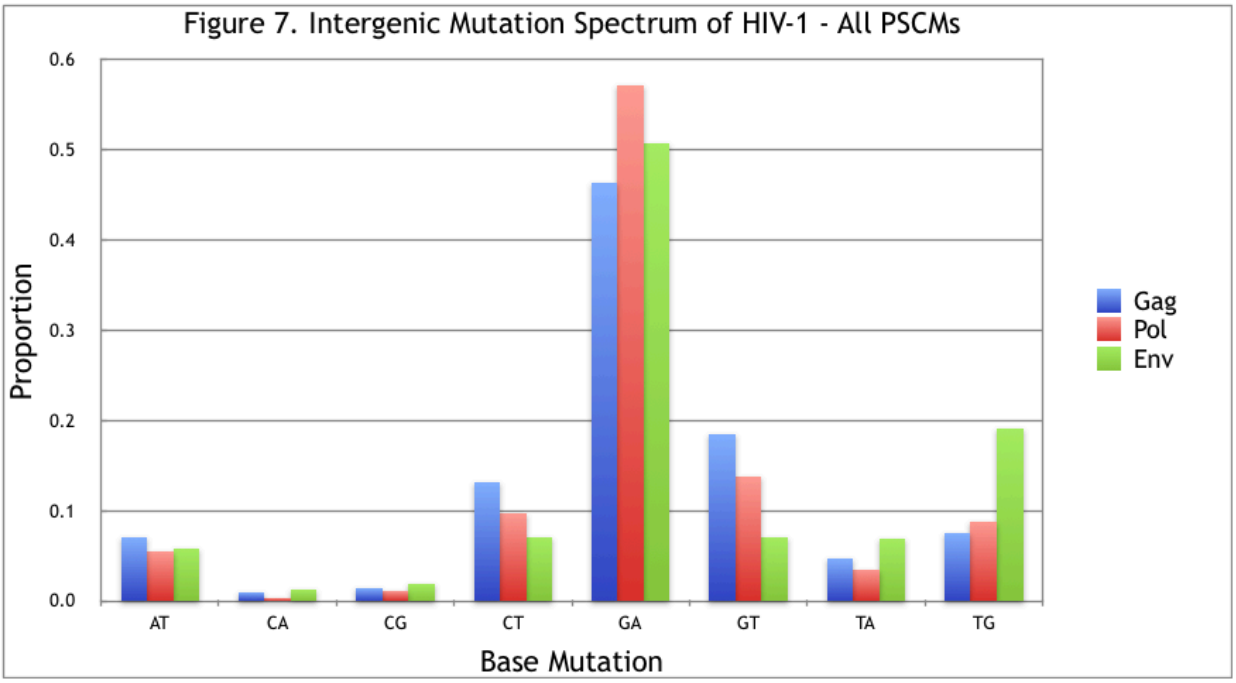


Figure 7. Intergenic mutation spectrum derived from only PSCMs found in the HIV-1 Gag, Pol, and Env genes from study ERP001266. APOBEC mutations have not been filtered out. Refer to Table 8C for data.

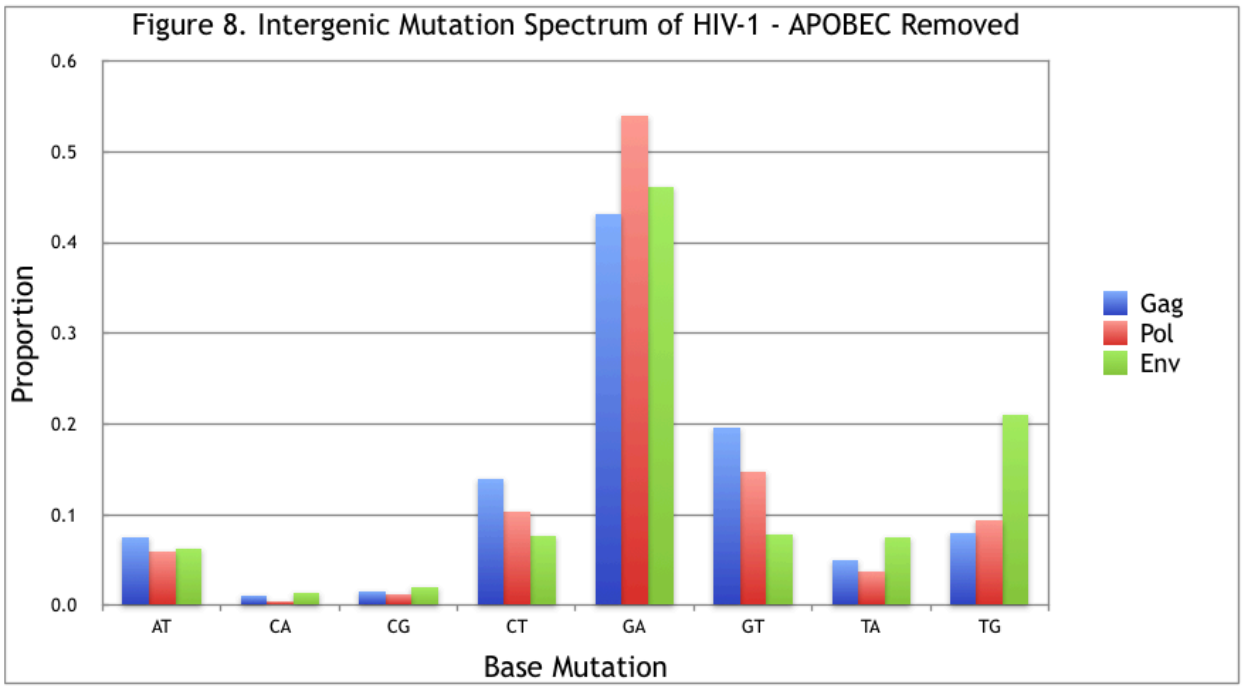


Figure 8. Intergenic mutation spectrum derived from only PSCMs found in the HIV-1 Gag, Pol, and Env genes from study ERP001266. APOBEC mutations have been filtered out. Refer to Table 8D for data.

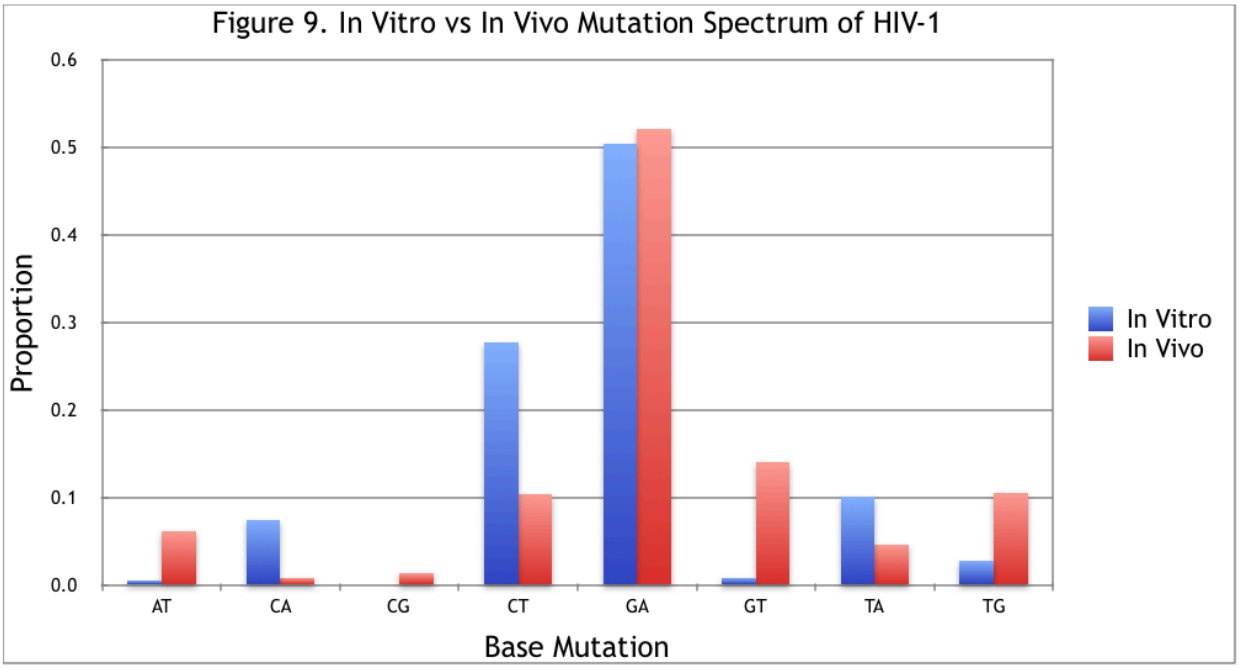


Figure 9. Mutation spectrum derived from only PSCMs found in the HIV-1 Gag, Pol, and Env genes from a publicly available in vitro HIV-1 study and study ERP001266 (in vivo). APOBEC mutations have not been filtered out. Refer to Table 8E for data.

4.4 Discussion

When looking just at the PSCM counts, it is clear that GA mutations are the most prevalent type of PSCM across the three studies (see Table 8). The neutral mutation spectra evens out across all studies when considering only PSCMs (see Figures 5, 6, 7, and 8). This is consistent with the idea that during infection, HIV sequence variation is shaped by strong selection pressure. Differences in the mutation spectra between the two ERP studies do not seem to be the result of sequencing bias, given that the post-selection mutations spectrums are fairly even for the Pol gene (see Figures 5 and 6). Instead, they may reflect differences in the activity of various other mutation sources between patient samples across studies. The selection-free intergenic mutation spectrum of study ERP001266 is fairly even across the Gag, Pol, Env genes (see Figures 7 and 8). Although study ERP001266 had by far the largest sample size among the three studies, the proportion of APOBEC mutations was the least and less than that of the previous study based on the provirus (Cuevas et al., 2015). This is consistent with the idea that most APOBEC mutations are deleterious. The mutation spectrum comparison between HIV-1 in vitro (publicly available) and in vivo (study ERP001266 chosen for its large sample size) show major discrepancies, specifically among the AT, CA, CT, and GT PSCMs, with TA and TG PSCMs to a lesser degree. Undoubtedly, there are some differences due to PSCMs resulting in TAA PSCs not being included in the in vivo mutation spectrum. However, as it stands, the in vitro and in vivo mutation spectra do not share any apparent trends with each other, which is expected.

5 Conclusion and Future Work

5.1 Conclusion

Having run my Python script on three publicly available HIV-1 genome studies, my resulting neutral mutation spectra are all fairly even across the three studies. This concurs with the idea that the HIV genome is strongly shaped by the effects of selection during infection. The relatively small proportion of APOBEC-induced GA PSCMs among study ERP001266 (study with the largest sample size of the three) is consistent with the idea that most APOBEC mutations are deleterious. The mutation spectra of HIV-1 in vitro and in vivo do not share any noticeable patterns with each other, as expected given the effects of selection and other mutation sources.

5.2 Future Work

To finish up my project, I will implement 1) my normalization code to normalize the mutation spectra, and 2) a non-negative matrix factorization-based blind source separation (BSS) algorithm on my neutral RT mutations to parse out individual mutation sources. It is unclear whether a selection-free mutation spectrum corresponds exclusively to an in vivo neutral RT spectrum because additional unknown sources of mutations is a possibility. The BSS algorithm will enable me to distinguish different mutation sources as well as hopefully discover a mutation signature for HIV-1 RT.

6 Authors

Jimmy Chan:

I wrote Python code for calling base mutations, identifying PSCMs, filtering out APOBEC mutations, and normalizing the mutation spectrum. These functions are listed in Figure 3, the functions flowchart.

Jay W. Kim:

Kim compiled the three HIV genome studies used in my project by 1) downloading the HIV-RNA genome sequences from the LANL-HIV databases through its Next-Gen Sequences Interface, 2) trimming the adapter sequences from the genome sequences and throwing out low-quality base calls, 3) aligning the short read sequences to the Gag, Pol and Env genes from a reference HIV genome (HIV-1 strain NL4-3) using the BWA package, and 4) removing the homopolymer errors from 454 sequencing reads based on optimal protein-nucleotide alignments using the HAXAT (Homopolymer Aware Cross Alignment Tool), an algorithm that strategically allows for frame shifts when aligning 454 reads. Kim also provided me with a skeleton Python script containing three functions (which I later modified): 1) capitalize_bases(), 2) parse_args(), and 3) print_spectrum().

7 References

Albertson, T.M. & Preston, B.D. (2006). DNA Replication Fidelity: Proofreading *in Trans. Current Biology,* 16(6), R209-R211. doi: 10.1016/j.cub.2006.02.031

Bakhanashvili, M., Novitsky, E., Levy, I. & Rahav, G. (2005). The fidelity of DNA synthesis by human immunodeficiency virus type 1 reverse transcriptase increases in the presence of polyamines. *Elsevier,* 579(6), 1435-1440. doi: 10.1016/j.febslet.2005.01.043

Cuevas, J.M., Geller, R., Garijo, R., López-Aldeguer, J., & Sanjuán, R. (2015). Extremely High Mutation Rate of HIV-1 In Vivo. *PLOS Biology,* 13(9), e1002251. doi:10.1371/journal.pbio.1002251

Gillick, K., et al. (2013). Suppression of HIV-1 Infection by APOBEC3 Proteins in Primary Human $CD4^+$ T Cells Is Associated with Inhibition of Processive Reverse Transcription as Well as Excessive Cytidine Deamination. *Journal of Virology,* 87(3), 1508-1517. doi: 10.1128/JVI.02587-12

Girerd-Genessay, I., et al. (2016). Higher HIV RNA Viral Load in Recent Patients with Symptomatic Acute HIV Infection in Lyon University Hospitals. *PLOS One*, 11(1), e0146978. doi: 10.1371/journal. pone.0146978

Goila-Gaur, R. & Strebel, K. (2008). HIV-1 Vif, APOBEC, and Intrinsic Immunity. *Retrovirology*, 5(51). doi: 10.1186/1742-4690-5-51

James, K., Gamba, P., Cockell, S.J. & Zenkin, N. (2017). Misincorporation by RNA polymerase is a major source of transcription pausing *in vivo. Nucleic Acids Research,* 45(3), 1105–1113. doi: 10.1093/nar/gkw969

Kunkel, T.A. (2004). DNA Replication Fidelity. *The Journal of Biological Chemistry,* 279, 16895-16898. doi: 10.1074/jbc.R400006200

Leal, E. S., Holmes, E. C., & Zanotto, P. M. (2004). Distinct patterns of natural selection in the reverse transcriptase gene of HIV-1 in the presence and absence of antiretroviral therapy. *Virology,* 325(2), 181-191.

Mak, J. & Kleiman, L. (1997). Primer tRNAs for Reverse Transcription. *Journal of Virology*. 71(11), 8087–8095.

Nelson, P.N., et al. (2003). Demystified . . . Human endogenous retroviruses. *Molecular Pathology*, 56(1), 11–18.

Pray, L. (2008). DNA Replication and Causes of Mutation. *Nature Education,* 1(1), 214.

Rebhandl, S., Huemer, M., Greil, R. & Geisberger, R. (2015). AID/APOBEC deaminases and cancer. *Oncoscience*, 2(5), 320-33. doi: 10.18632/oncoscience.155

Roberts, S.A., et al. (2013). An APOBEC Cytidine Deaminase Mutagenesis Pattern is Widespread in Human Cancers. *Nature Genetics*, 45(9), 970–976. doi: 10.1038/ng.2702

Sundquist, W. I. & Kräusslich, H. (2012). HIV-1 Assembly, Budding, and Maturation. *Cold Spring Harbor Perspectives in Medicine,* 2(7), a006924. doi: 10.1101/cshperspect.a006924

Sydow, J.F. & Cramer, P. (2009). RNA polymerase fidelity and transcriptional proofreading. *Current Opinion in Structural Biology*, 19(6), 732-739. doi: 10.1016/j.sbi.2009.10.009

Wang, B., Opron, K., Burton, Z.F., Cukier, R.I. & Feig, M. (2015). Five checkpoints maintaining the fidelity of transcription by RNA polymerases in structural and energetic details. *Nucleic Acids Research*, 43(2), 1133–1146. doi: 10.1093/nar/gku1370

Wilen, C.B., Tilton, J.C. & Doms, R.W. (2012). HIV: Cell Binding and Entry. *Cold Spring harbor Perspectives in Medicine*, 2(8), a006866. doi: 10.1101/cshperspect.a006866