Computational Analysis of Papionini Evolution Using Alu Insertions

Vallmer Edward Jordan II

Louisiana State University and Agricultural and Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_dissertations

Part of the Bioinformatics Commons, Computational Biology Commons, Evolution Commons, Genetics Commons, and the Genomics Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_dissertations/4646

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
COMPUTATIONAL ANALYSIS OF PAPIONINI EVOLUTION USING ALU INSERTIONS

A Dissertation

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College In partial fulfillment of the Requirements for the degree of Doctor of Philosophy

in

The Department of Biological Sciences

by

Vallmer Edward Jordan II
B.S. Morehouse College, 2013
August 2018
TABLE OF CONTENTS

ABSTRACT........................................................................................................................iii

CHAPTER ONE: BACKGROUND............................................................................................1

CHAPTER TWO: A COMPUTATIONAL RECONSTRUCTION OF PAPIO PHLOGENY USING ALU INSERTION POLYMORPHISMS.................................................12

CHAPTER THREE: COMPUTATIONAL ANALYSIS OF EVOLUTIONARY RELATIONSHIPS WITHIN PAPIONINA USING ALU POLYMORPHISMS..................43

CHAPTER FOUR: CONCLUSIONS......................................................................................51

APPENDIX A: SUPPLEMENTAL DATA..............................................................................53

APPENDIX B: LETTERS OF REQUEST AND PERMISSION............................................94

VITA.....................................................................................................................................97
ABSTRACT

Alu elements are primate specific retrotransposons that have remained active throughout the course of primate evolution. As a result of this sustained mobilization, Alu elements are present in greater copy number in primate genomes than any other transposable element. An average of over one million Alu elements has been identified in every sequenced haplorrhine genome to date. These characteristics qualify Alu elements as ideal characters for studying evolutionary relationship among primates.

The increasing availability of whole genome sequencing data presents novel challenges and opportunities for comparative genomic analyses. Genomic data is now publicly available for most primate species. Such an abundance of resources allows researchers to re-examine previously unresolved or unexplored evolutionary relationships applying a comprehensive whole genome approach. The implications of such research models for studying human biology and evolution. Historically, the Old World monkey primate models has been a popular choice for investigating the human condition. More specifically, no catarrhine taxon has been exploited more extensively than those belonging to the Papionini tribe.

This dissertation describes an innovation computational method suitable for examining complex phylogenetic relationships among primates. Furthermore, it utilizes a quickly expanding database of publicly available whole genome sequencing data to perform phylogenetic and population genetic analyses. Through an integrative approach, the reported algorithm can identify Alu insertions indicative of hybridization and admixture. In addition, this method can be used to construct fully resolved cladograms despite well-documented histories of admixture and hybridization.
CHAPTER ONE: BACKGROUND

Discovery of Mobile Elements

More than 150 years have elapsed since heritable genetic units were first described by Gregor Mendel (Smýkal et al., 2016). Much of the mystery shrouding early abstract concepts of inheritance and evolution has been dissipated by technological advancements allowing scientists to determine the precise molecular identity of DNA sequences (Sanger, Nicklen, & Coulson, 1977; Watson & Crick, 1953). Many principles and paradigms have been challenged and re-examined as novel discoveries and methodologies rapidly emerge. One particular concept that persisted for over a century was the notion that genes are fixed units (T. H. Morgan, 1922). Furthermore, this characteristic formed the foundation upon which many subsequent genetic principles were constructed (Carlson, 1966; Demerec, Kaufmann, Fano, Sutton, & Sansome, 1933; Wright, 1927). Thus, when this paradigm was challenged by findings reported in 1950 by Barbara McClintock, it was met with fierce resistance (Fedoroff, 2012; McClintock, 1950; Ravindran, 2012).

While studying chromosome breakage in Zea mays, Barbara McClintock identified genes capable of shifting their genomic position (Fedoroff, 2012; McClintock, 1950, 1965, 1987; Ravindran, 2012). These mutable loci are now known as transposable elements (TEs). In particular, McClintock discovered a transposable element system capable of modifying corn kernel color variation. The two TEs active within this system were the Activator element and the Dissociation element. McClintock found that the activity of the Dissociation element was dependent upon that of the Activator element. She observed that when the Dissociation element integrated into particular genic regions, the resulting mutation repressed pigment production. Alternatively, when the element was excised from this same region, pigment production
resumed. These findings were revolutionary because they demonstrated that both the position and structure of genetic material could be altered during development. At the time, however, it was widely accepted that DNA was stable and immutable (Carlson, 1966; McClintock, 1987; T. H. Morgan, 1922; Ravindran, 2012). Thus, it was impossible to neatly integrate McClintock’s findings into such a rigid pre-existing paradigm. As a result, it took decades for the implications of her work to be widely recognized and accepted by her contemporaries. Ultimately, TEs were identified ubiquitously in other organisms, starting with viruses and bacteria (Engels & Preston, 1981; McClintock, 1987; Shapiro, 1969; Taylor, 1963). In 1983, Barbara McClintock won a Nobel Prize in Physiology or Medicine for the discovery and characterization of TEs.

**Transposable Elements**

TEs can be defined as DNA sequences capable of changing their genomic position. Conservative estimates compute that up to 90% of the genomes in some plants are comprised of TEs, and more than 50% in most mammals (Chimpanzee Sequencing and Analysis Consortium, 2005; Feschotte & Mouches, 2000; Izsvák et al., 1999; Lander et al., 2001; Locke et al., 2011; McLain et al., 2012; Meyer et al., 2012; Meyers, Tingey, & Morgante, 2001; G. T. Morgan, 1995; Oosumi, Garlick, & Belknap, 1996; SanMiguel et al., 1996; A. Smit & Riggs, 1996; Tu, 1997, 2001; Ünsal & Morgan, 1995; Vicent et al., 1999). TEs can be divided into two subclasses: DNA transposons and retrotransposons. The main difference between the two classes is mode of mobilization. DNA transposons move via a “cut-and-paste” mechanism. DNA transposon sequences are excised from one genomic location and later integrated into a novel location (A. Smit & Riggs, 1996; A. F. Smit, 1996). On the other hand, retrotransposons mobilize via a copy-and-paste mechanism. First, an RNA copy of the retrotransposon sequence is transcribed; later this RNA copy is reverse-transcribed into DNA at a novel genomic position.
(A. F. Smit, Toth, Riggs, & Jurka, 1995). As a result of each successful retrotransposition event, a new copy of the element is introduced to the genome, with the sequence also preserved at its original location. As a result, retrotransposons can exponentially increase their copy number within a host genome (Cordaux & Batzer, 2009). DNA transposons, though active in some mammalian genomes, are thought to be currently inactive in primate genomes (Pace & Feschotte, 2007).

Retrotransposons can be further separated into two subclasses based on the presence of long terminal repeats (LTRs): LTR and non-LTR retrotransposons. In primates, LTR retrotransposons are known as endogenous retroviruses. Non-LTR retrotransposons can be classified as either autonomous or non-autonomous, based on whether or not their DNA sequence encodes the enzymatic machinery requisite for retrotransposition. The two retrotransposons having the greatest impact on primate genome structure and evolution are long interspersed element 1 (LINE1 or L1) and Alu (Batzer & Deininger, 2002; Cordaux & Batzer, 2009; Konkel, Walker, & Batzer, 2010; Locke et al., 2011; McLain et al., 2012; Meyer et al., 2012; Ray, Xing, Salem, & Batzer, 2006; A. Roy-Engel, A Batzer, & Deininger, 2008; A. M. Roy-Engel et al., 2002; A.-H. Salem et al., 2003; Salem, Kilroy, Watkins, Jorde, & Batzer, 2003).

**Figure 1.1.** Canonical structure of an L1 element. Target site duplications (TSD, red) flanking both ends of the elements. The 5' and 3' UTR are represented by blue boxes. The 5' UTR contains the RNA polymerase II promoter. Both open reading frames are positioned between the UTRs (ORF1, green; ORF2, brown). The L1 terminates with an A-rich tail (purple).

L1 is the only autonomous retrotransposon known to be currently active in primates (Lander et al., 2001). Roughly 6 kb, a full length L1 sequence is comprised of a 5'UTR
containing an RNA polymerase II promoter, two open reading frames, a 3’ UTR terminating in a polyadenylation signal, and an A-rich tail (Deininger & Batzer, 2002; Kazazian & Moran, 1998). An RNA-binding protein is encoded by ORF1, with ORF2 encoded a bi-functional protein with endonuclease and reverse transcriptase activities (Feng, Moran, Kazazian, & Boeke, 1996; Jurka, 1997; Mathias, Scott, Kazazian, Boeke, & Gabriel, 1991). L1 is the most successful TE in primate genomes (sequenced to date) in terms of mass, representing roughly 17% of the genome in humans (Lander et al., 2001). L1 is thought to have inserted over 150 million years ago, and thus has amplified extensively in most mammalian genomes.

**Figure 1.2.** Canonical structure of an *Alu* element. Target site duplications (TSD, red) flanking both ends of the *Alu* elements. The left (green) and right (brown) monomers are separated by an A-rich region (white). The blue portions located on the left monomer represent the A and B boxes containing promoter regions. The *Alu* terminates with an A-rich tail (purple).

*Alu* elements are a primate specific-short interspersed element (SINE) roughly 300 bases in length (full-length). *Alu* elements are non-autonomous retrotransposons, relying on the enzymatic machinery encoded by L1 (Schmid, 2003). A full-length *Alu* element is dimeric in structure, containing a left and right monomer separated by an A-rich linker region. The left monomer encodes an RNA polymerase III promotor, with the right monomer followed by an A-rich tail (Batz & Deininger, 2002; Batzer et al., 1990; A. Roy-Engel et al., 2008). *Alu* elements are thought to have evolved from a 7SL RNA roughly 65 million years ago during the genesis of primate evolution (Okada, 1991; Ullu & Tschudi, 1984). *Alu* elements are the most successful TE in primate genomes in terms of copy number, with over 1 million copies identified in all
sequenced haplorrhine genomes to date (Chimpanzee Sequencing and Analysis Consortium, 2005; Lander et al., 2001; J. Li et al., 2009; Locke et al., 2011).  

**Retrotransposons as Primate Phylogenetic Characters**

Retrotransposons are powerful characters for resolving complex phylogenetic relationships between primate taxa (Chimpanzee Sequencing and Analysis Consortium, 2005; J. Li et al., 2009; Locke et al., 2011; McLain et al., 2012; Meyer et al., 2012; Pace & Feschotte, 2007; Ray et al., 2005; A.-H. Salem et al., 2003; Stoneking et al., 1997; Walker et al., 2017; Xing et al., 2005; Xing et al., 2007). This utility can be attributed largely to the assumption that retrotransposons are identical-by-decent, meaning that shared insertions are inherited from a common ancestor, rather than the result of convergent evolution (Ray et al., 2006). Furthermore, it can be assumed that the ancestral state of each locus is the absence of the retrotransposon. Additionally, retrotransposon are useful characters because they can be have remained active throughout the course of primate evolution (Batzer & Deininger, 2002; Deininger & Batzer, 2002; A. Roy-Engel et al., 2008; A.-H. Salem et al., 2003).

Both *Alu* and L1 have been successfully utilized to resolve evolutionary relationships between primate species (Jordan et al., 2018; J. Li et al., 2009; McLain et al., 2012; Meyer et al., 2012; Ray et al., 2005; Xing et al., 2005). Both are considered nearly homoplasy-free, with most potential cases of homoplasy resolved easily using Sanger sequencing. However, *Alu* characters have been utilized more extensively due to their small compact size. In past studies, polytomies have been successfully resolved through the analysis of *Alu* insertions polymorphic between the taxa of interest. Such a panel of phylogenetically informative *Alu* characters has been previously ascertained from comparisons of high-quality draft reference genomes. Although the availability of such genomic data has increased exponentially following the successful completion of the
Human Genome Project, reference genomes have not been constructed for all extant species contained within most primate genera. In such cases, available reference genomes are evaluated using software that identifies retrotransposons and compares analogous sequences between species (McLain et al., 2012; Meyer et al., 2012; Ray et al., 2005; Xing et al., 2005). This initial analysis serves as a preliminary screening mechanism for identifying potentially informative phylogenetic characters. Next, a PCR analysis is conducted using a panel of individuals representing the taxa of interest. The resulting genotypes can be used to construct a phylogenetic cladogram.

Although this method has proven useful in the past, it is limited by the availability of high-quality draft reference genomes. Such sequences are usually generated through large-scale collaborative genome sequencing projects. Alternatively, publicly available whole genome sequence (WGS) data is generated at a much faster pace. Unlike the high-quality draft reference genomes, WGS data is currently available for several primate genera with multiple individuals representing all extant species. In response to this increase in data availability, we created a computational algorithm to identify phylogenetically informative Alu insertions using WGS data.

**Computational Phylogenetic Analysis**

Little is known about the utility of Alu elements for resolving phylogenetic relationship within taxa with an extensive history of admixture. In theory, such elements would introduce high levels of homoplasy. In chapter two and three of this dissertation, we explore the power of a novel computational method designed to target Alu elements polymorphic within a dataset. We used this method to examine evolutionary relationships between *Papio* baboons. Extant *Papio* baboons have several well-documented hybrid zones. In addition, hybridization has also been documented between *Papio* and *Theropithecus*. Such intra-genus hybridization might be
indicative of a complex history of hybridization persisting throughout the course of *Papionini* evolution. The later form of hybridization is evaluated in chapter 3.

**References**


Wright, S. (1927). The Effects in Combination of the Major Color-Factors of the Guinea Pig. *Genetics, 12*(6), 530-569.


CHAPTER TWO: A COMPUTATIONAL RECONSTRUCTION OF PAPIO PHYLOGENY USING ALU POLYMORPHISMS

Background

The burgeoning diversity and availability of whole genome sequencing (WGS) data offers intriguing possibilities for the field of comparative primate genomics. Currently, WGS data are publicly available for over 100 primate species (NCBI Resource Coordinators 2016). Traditionally, significant interest in the genetics of non-human primates stems from their sustained role as popular research models for studying human biology and evolution (Bontrop, 2001; Goodman & Check, 2002; Jolly, 2001; Patterson & Carrion, 2005; Phillips et al., 2014).

One such primate—well established as a model for human genetics and disease susceptibility—is the *Papio* baboon (Cox et al., 2013; Dormehl, Hugo, & Beverley, 1992; Jerome, Kimmel, McAlister, & Weaver, 1986; Kammerer, Cox, Mahaney, Rogers, & Shade, 2001; McGill, McMahan, Kruski, & Mott, 1981; VandeBerg, Williams-Blangero, & Tardif, 2009). In addition to close genetic relatedness, the temporal and ecological landscape of early *Papio* evolution bears striking resemblance to that of early hominins (Jablonski & Frost, 2010; Jolly, 2001; Rodseth et al., 1991; Strum & Mitchell, 1987). Both include ancient episodes of admixture, as well as migration out of Africa into the Arabian Peninsula during the Pleistocene (Groucutt et al., 2015; Jolly, 2001; Kopp et al., 2014; Racimo, Sankararaman, Nielsen, & Huerta-Sanchez, 2015; Sankararaman et al., 2014; Wildman et al., 2004; Winney et al., 2004). Appropriately, *Papio* baboons represent an intriguing model for human evolution.

*Papio* baboons occupy the largest geographical distribution of any non-human primate genus on the African continent (Barrett & Henzi, 2008; Caldecott, Miles, Programme, & Centre, 2005; Eeley & Foley, 1999). These ground dwelling Old World monkeys inhabit most of sub-Saharan Africa, to the exclusion of the tropical rainforests of West Africa and the Congo Basin,
and also extend into the south-western region of the Arabian Peninsula (Groves, 2001; Jolly, 1993). *Papio* systematics have been extensively studied over the past 60 years with much debate as to which forms warrant species status (Groves, 2001). The disagreement is in essence philosophical, centered on the question of what constitutes a species. However, recent studies employ a phylogenetic species concept (Jolly, Burrell, Phillips-Conroy, Bergey, & Rogers, 2011; Steely et al., 2017; Walker et al., 2017; Zinner, Wertheimer, Liedigk, Groeneveld, & Roos, 2013), positing that consistent differences in physical appearance, ecology and social behavior justify the recognition of six extant species: *P. anubis*, *P. hamadryas*, *P. papio*, *P. cynocephalus*, *P. ursinus* and *P. kindae*. In this study, we recognize all six as species.

Despite considerable interest in *Papio* systematics, a fully resolved consensus phylogeny remains undetermined (Wildman et al., 2004; Zinner, Groeneveld, Keller, & Roos, 2009; Zinner et al., 2013). Interfertility has been documented between all neighboring species, with persisting natural hybrid zones in several regions where distinct morphotypes (species) come into contact (Alberts & Altmann, 2001; Bergman, Phillips-Conroy, & Jolly, 2008; Jolly et al., 2011; Maples & McKern, 1967; Nagel, 1973; Phillips-Conroy, Jolly, & Brett, 1991; Szmulewicz et al., 1999). Thus, discordance between mitochondrial, morphological, and nuclear phylogenetic reconstructions could in part stem from a dense history of admixture and reticulation persisting throughout the course of *Papio* evolution. Mitochondrial based phylogenies support the divergence of *Papio* into northern and southern lineages (Zinner et al., 2009; Zinner et al., 2013). Individuals belonging to *P. anubis*, *P. papio* and *P. hamadryas* are consistently placed within the northern clade; with individuals belonging to *P. kindae* and *P. ursinus* comprising the southern clade. In these analyses, however, the placement of *P. cynocephalus* remains unclear with individuals found in both clades. In addition, such reconstructions have proven unsuccessful at
resolving phylogenetic relationships within each clade. Thus additional analyses employing novel methodologies could further serve to elucidate evolutionary relationship within *Papio*.

*Alu* elements are well-established DNA markers for the study of systematic and population genetic relationships (Batzer et al., 1996; J. Li et al., 2009; McLain et al., 2012; Meyer et al., 2012; A.-H. Salem et al., 2003; Schmitz, Ohme, & Zischler, 2001; Shedlock & Okada, 2000; Stoneking et al., 1997; Watkins et al., 2003; Xing et al., 2005; Xing et al., 2007). In part, they are effective evolutionary characters because of their high copy number in primate genomes and sustained mobilization throughout the course of primate evolution (~65 MY) (Batzer & Deininger, 2002; Deininger & Batzer, 2002; A. Roy-Engel et al., 2008). Over 1.2 million copies have been identified in the human genome (Lander et al., 2001), with similar numbers reported for all other haplorrhine genomes sequenced to date (Chimpanzee Sequencing and Analysis Consortium, 2005; Gibbs et al., 2007; Locke et al., 2011; Scally et al., 2012). *Alu* elements are discrete primate-specific DNA sequences (~300 bp) belonging to a class of non-LTR (long terminal repeat) retrotransposons termed short interspersed elements (SINEs). Following the transcription of a SINE, the mRNA sequence can be reverse transcribed into DNA, producing a new copy at a novel position in the host genome (Cost, Feng, Jacquier, & Boeke, 2002; Luan & Eickbush, 1995; Luan, Korman, Jakubczak, & Eickbush, 1993). Over time, this process known as target primed reverse transcription (TPRT) can exponentially increase the retrotransposon content of a host genome. *Alu* elements, as well as all other SINEs, lack the requisite enzymatic machinery for TPRT; thus they require proteins encoded by larger retrotransposons known as LINEs (long interspersed elements) (Batzer & Deininger, 2002; Comeaux, Roy-Engel, Hedges, & Deininger, 2009; Dewannieux, Esnault, & Heidmann, 2003).
SINEs are valuable evolutionary characters because they can be assumed to be identical by descent, meaning that insertions shared between individuals were inherited from a common ancestor, rather than acquired by independent events (Ray et al., 2006). Additionally, retrotransposons have known directionality (Batzer & Deininger, 1991; Konkel et al., 2010), with the ancestral state being the absence of the insertion. *Alu* elements are popular retrotransposon markers because their short length makes them particularly easy to assay using standard PCR. Considered nearly homoplasy-free (Batzer & Deininger, 2002; Deininger & Batzer, 2002), most potential sources of homoplasy involving *Alu* elements can be resolved through Sanger sequencing (J. Li et al., 2009; Ray et al., 2006; Xing et al., 2005). Recent studies demonstrate the utility of *Alu* elements for *Papio* species identification, as well as retrieving population structure within distinct *Papio* species (Steely et al., 2017; Walker et al., 2017). Furthermore, *Alu* elements have been successfully used to resolve controversial relationships between primates (J. Li et al., 2009; Meyer et al., 2012; Roos & Geissmann, 2001; A.-H. Salem et al., 2003). However, little is known about the efficacy of *Alu* elements to resolve phylogenetic relationships involving high levels of admixture.

Although a high-quality reference assembly currently exists for only one *Papio* species (*P. anubis*), WGS data have been generated for individuals representing all six *Papio* species through the Baboon Genome Consortium. Thus it is possible to conduct a comprehensive whole genome analysis of *Papio* phylogeny using *Alu* polymorphisms between species of the genus. For the present study, we created a computational pipeline to identify and characterize recently integrated *Alu* elements polymorphic within the genus *Papio*. These *Alu* insertion polymorphisms were used to reconstruct phylogenetic relationships within *Papio*. By utilizing *M. mulatta* as our reference, our approach placed equal evolutionary distance between each
*Papio* diversity sample and the reference assembly [Mmul8.0.1]. The computational analyses performed in this study generated a well-supported phylogeny of *Papio* baboons and represents the most comprehensive *Alu*-based phylogenetic analysis reported to date. In addition, we report a novel approach to admixture and reticulation analysis using *Alu* insertions.

**Methods**

**Samples**

Whole-genome sequencing was performed by the Baylor College of Medicine Human Genome Sequencing Center on a panel of fifteen *Papio* baboons: four *P. anubis*, two *P. papio*, two *P. hamadryas*, three *P. kindae*, two *P. cynocephalus*, and two *P. ursinus*. In order to sample an equal number of individuals from each species, we used two individuals from each of the six extant *Papio* species (we randomly selected two individuals from *P. anubis* and *P. kindae*) to conduct our computational analysis. Lastly, our panel included WGS data from the macaque sample used to build the latest *M. mulatta* assembly [Mmul8.0.1] (Table A.1).

WGS data were accessed from the NCBI-SRA database (Coordinators, 2016). The SRA-toolkit (fastq-dump utility) (Leinonen, Sugawara, Shumway, & on behalf of the International Nucleotide Sequence Database, 2011) was used to download paired-end next generation sequencing reads and convert them from .sra files to interleaved fastq files. We then used nesoni ([https://github.com/Victorian-Bioinformatics-Consortium/nesoni; last accessed March 2018](https://github.com/Victorian-Bioinformatics-Consortium/nesoni)) to prune all known adapters, cleave bases with a phred quality score of 10 or lower, and exclude reads shorter than 24 base pairs in length. Two output fastq files were produced: one containing clean paired-end reads (both reads passed the nesoni filter), and a second containing unpaired orphan reads (one of the paired-end reads was excised).
**Polymorphic Alu Insertion Detection**

We developed a computational pipeline to identify and characterize recently integrated *Alu* elements in paired-end next-generation sequencing (NGS) data. Our approach targeted young *Alu* insertions still polymorphic within the panel of individuals listed in the previous section. The approximate chromosomal position of each candidate insertion was estimated using a split-read method (Figure 2.1). The resulting genotypes, generated for all individuals in our panel, revealed markers that provided phylogenetic signal.

The *Alu*Y subfamily has been identified as youngest and most active *Alu* subfamily in Simiiformes (Batzer & Deininger, 2002; Batzer et al., 1990; Bennett et al., 2008; Carroll et al., 2001). Thus, in the alignment phase, we used BWA mem (H. Li, 2013) to map paired-end NGS reads to a consensus *Alu*Y sequence obtained from Repbase (Jurka, 1998). Individual reads were required to map to either the head (5’) or tail (3’) of the *Alu*Y consensus sequence. In addition, reads mapping to the head of an *Alu* insertion were required to contain at least 15 bp of unmapped/non-*Alu* sequence directly upstream of the (5’) start of the *Alu* sequence. Likewise, reads mapping to the tail of the consensus *Alu* sequence were required to contain no less than 15 bases of unmapped sequence directly flanking the (3’) end of the sequence. Reads were mapped to the *Alu*Y consensus twice: once using the standard BWA mem parameters, and a second time using more liberal parameters (described in Table A.2). Split-reads identified using standard parameters were later used to predict the location of an *Alu* integration site, while those identified during the liberal run were used simply to provide additional support for the insertion event. The *Alu* portion of each candidate split-read was then cleaved and remaining sequence aligned to Mmul8.0.1 using bowtie2 (Langmead & Salzberg, 2012). Split-reads were categorized as sequences that mapped uniquely to the *Alu*Y consensus and the Mmul8.0.1 assembly.
The approximate genomic position of each candidate insertion was calculated directly from the mapping positions of split-reads to Mmul8.0.1 and the AluY consensus. Alu insertion orientation was inferred from the alignment orientation of the supporting reads when mapped to the AluY consensus and Mmul8.0.1 assembly. During this phase the integration orientation of each candidate insertion was predicted in the forward orientation if positioned 5’ to 3’ on the sense strand, and the reverse orientation if positioned 5’ to 3’ on the anti-sense strand. If a split-read mapped in the same orientation to the consensus AluY and the Mmul8.0.1 assembly, it was predicted in the forward orientation. If the alignment orientations were discordant, the insertion was predicted in the reverse orientation.

**Figure 2.1.** Computational detection of Alu insertion polymorphisms using split-reads. Alu insertions were identified using sequencing reads spanning the Alu integration locus whether these split-reads spanned the 5’ (A and C) or the 3’ (B and D) end of the insertion. The four split-reads represented in this figure are labeled A, B, C, and D. Green boxes represent Alu sequence; gray boxes denote flanking sequence. If the split-read is paired and its read-pair mapped to the flanking sequence (B and C), these mapping coordinates were used to provide additional support for the location predicted by the split-read. If the split-read’s read-pair mapped to the Alu (A and D), this was used to provide additional support for the presence of the predicted Alu insertion.
Approximate genomic positions for non-reference (absent in Mmul8.0.1) Alu insertions, predicted in any of the twelve Papio individuals, were concatenated into a comprehensive list with the goal of identifying phylogenetically informative markers. All of these insertions were predicted from split-reads obtained during the standard Alu alignment run. In principle, phylogenetically informative Alu elements would have integrated into the Papio lineage following its divergence from Macaca. Thus, insertions shared between Papio and the Macaca mulatta sample were excluded. Likewise Alu elements identified in only one Papio sample were phylogenetically-uninformative, and thus were also excluded from this portion of the study. The remaining loci were genotyped in every individual on the panel. The three possible genotypes – homozygous present, homozygous absent, and heterozygous – were determined by analyzing sequences spanning the insertion locus. It was initially assumed that an individual was homozygous present for every insertion predicted in that sample. Likewise, it was initially assumed that an individual was homozygous absent for every locus not predicted in that individual. Insertions initially determined to be homozygous present were then re-evaluated to determine if they were in fact heterozygous present. Heterozygosity was determined by evaluating reads that mapped uniquely to the Mmul8.0.1 assembly. An insertion was reclassified as heterozygous if we identified reads in that individual that mapped continuously (without interruption) through the homologous empty site in the Mmul8.0.1 assembly. This empty site was defined as a sequence containing at least 15bp of flanking both upstream and downstream from the predicted insertion locus. Additionally, if a homozygous absent genotype was predicted in a region with a local read-depth less than two standard deviations from the global mean, the genotype was instead considered unknown.
PCR Validation

The performance of the algorithm used in this study was assessed by comparing PCR validations performed for 494 loci in a panel of six Papio baboons: one from each extant Papio species (Steely et al., 2017). From this dataset, our algorithm correctly predicted 98% of the PCR-validated events for presence/absence. In addition, the correct genotype (homozygous present, homozygous absent, or heterozygous) was computationally predicted for 93% of all events.

Basal Divergence Analysis

Previous phylogenetic analyses support the ancestral divergence of Papio into two clades: northern and southern lineages (Zinner et al., 2009; Zinner et al., 2013). To evaluate this hypothesis we created a computational method to identify the basal divergence model best supported by our Papio dataset. A genus comprised of six species with three different possible phylogenetic topologies generates 31 different unique models for estimating the basal divergence (Figure A.1). For each model we determined the total number of insertions that supported and conflicted with each basal divergence. We calculated the standard deviation and z-score for each model. The model with the highest z-score represents the basal divergence model best supported by the dataset.

Phylogenetic Analysis

We used the model representing the basal divergence with the highest z-score (described in the previous section) as a pre-condition for our phylogenetic analysis. A comprehensive list of Alu insertions supporting this model (consistent with the north-south split hypothesis) were used to further resolve phylogenetic relationships within Papio. A heuristic search was performed using PAUP* 4.0b10 (Swofford, 2011). Since it is assumed that the absence of an Alu insertion
is the ancestral state of each locus, Dollo’s law of irreversibility (Durrant, 1954) was used in the analysis. Thirteen individuals were evaluated in this analysis: twelve *Papio* baboons, two representing each of the six extant *Papio* species, along with the *M. mulatta* sample used to build the Mmul8.0.1 assembly. Each individual received a score for each locus based on its computationally derived genotype. The presence of an insertion was scored as “1” for a filled site and “0” for an empty site; unknown genotypes were scored as “?”.

Using PAUP we conducted a heuristic search using genotype data from *Alu* polymorphisms concordant with the north-south split with *M. mulatta* set as the outgroup. All loci were classified as individual insertions and set to Dollo.up for parsimony analysis as described previously (Xing et al., 2005). 10,000 bootstrap replicates were performed with the maximum tree space set to all possible trees.

We wrote a series of Python scripts to sort *Alu* insertions into clusters based on which baboons shared the insertion. This allowed us to determine the total number of *Alu* insertions shared between different sets/combinations of baboons. Each cluster contained *Alu* insertions shared among a distinct combination of baboons, yet absent from all other samples. For example, one cluster contained all *Alu* insertions shared between the *P. cynocephalus* samples and the *P. kindae* samples, yet absent from all remaining samples. Another cluster was comprised of *Alu* insertions shared between all six northern baboons, yet absent from all six southern baboons. Each cluster represents the total number of insertions shared uniquely between a particular "combination/set" of baboons. The resulting clusters were then analyzed to identify patterns of shared *Alu* polymorphisms. Using this script we quantified the total number of *Papio* indicative *Alu*-insertions, markers present in all six extant *Papio* species, yet absent from the *M. mulatta* sample. Clade indicative *Alu* polymorphisms were defined as insertions present in every species belonging to one clade, yet absent from all individuals in the other clade. In addition, we
evaluated patterns of shared Alu polymorphism exhibited within each clade. In this analysis, we identified Alu polymorphisms exclusive to either the northern or southern clade, yet not present in all species within that clade. Lastly, we quantified the total number of species indicative Alu elements, defined as Alu polymorphisms present in both individuals belonging to a species, yet absent from all other Papio individuals in our panel.

Results

Polymorphic Alu Identification

WGS data for multiple Papio baboons were generated through the Baboon Genome Analysis Consortium and made available on NCBI. From this dataset we selected a diversity panel consisting of twelve Papio baboons: two from each of the six extant species. We then used our computational pipeline to process these WGS samples, targeting Alu insertions present in multiple diversity samples, yet absent from the latest M. mulatta reference assembly [Mmul8.0.1]. In total, we identified 187,379 Alu insertions fitting this criterion.

Basal Divergence Modeling

We evaluated 31 distinct basal divergence models (see Methods), to determine the one best supported by our computational genotype data (Figure A.1). The model with the highest z-score divided the Papio genus into two lineages: a northern clade containing P. papio, P. anubis, and P. hamadryas; and a southern clade consisting of P. cynocephalus, P. ursinus, and P. kindae (Table A.3). Of the 187,379 non-reference insertions (not present in Mmul8.0.1) reported in the previous section, 123,120 were concordant with this north-south basal divergence model (~66%) and 64,259 (~34%) were discordant.
**Papio Phylogeny**

Using the data obtained from the panel of twelve *Papio* individuals, we constructed an *Alu*-based phylogeny of *Papio* baboons. For this analysis we used genotype data for 123,120 *Alu* insertions concordant with the north-south split hypothesis. The resulting cladogram resolved relationships within *Papio* with 100% bootstrap support at each node (C.I. = 0.703, H.I. = 0.297) (Figure 2.2). Bootstrap values along with the total number of insertions supporting each node are included in Figure 2.2.

![Phylogenetic tree of Papio baboons](image)

**Figure 2.2.** *Alu*-based phylogeny of extant *Papio* baboon species. Phylogenetic relationships of *Papio* baboons constructed using 123,120 *Alu* insertion polymorphisms. Genotypes computationally determined in 12 *Papio* baboons were used to construct a Dollo parsimony tree using *M. mulatta* as an outgroup. The percentage of bootstrap replicates (out of 10,000 iterations) is listed below each branch; the number of *Alu* insertions supporting each node is listed above each branch. Homoplasy index (H.I.) and consistency index (C.I.) are included below the cladogram.

- H.I. = 0.297
- C.I. = 0.703
To further examine evolutionary relationships within *Papio*, *Alu* insertions shared among multiple samples were clustered according to the patterns of shared *Alu* insertion polymorphisms determined for our *Papio* samples. This analysis was conducted multiple times, using various combinations of individuals from each species. Regardless of the representative individual selected for each species, the rank and size of each cluster, remained consistent. However, because we were particularly interested in observing clusters formed between individuals belonging to different species, we used one representative sample from each species. In each species, we selected the individual with sequencing coverage closest to the average coverage determined across all samples (Table A.1). The resulting clusters are displayed in Figure 2.3. Of the 187,379 *Alu* insertions identified in all twelve samples, we retained only those shared among multiple individuals from our panel of six *Papio* individuals. In total, we identified 106,204 such elements grouped into 57 unique clusters [Table A.4]. Figure 2.3 displays the fifteen largest clusters, representing a total of 76,264 *Alu* insertions (~72% of the dataset). The largest cluster contained 32,156 markers present in all six *Papio* species (Figure 2.3). Seven of the eight next largest clusters were shared exclusively between baboons belonging to the same clade (north/south). In total, these seven clusters contained 27,314 *Alu* insertions (~26% of the dataset). Of the remaining clusters, four consisted of markers shared between five of the six *Papio* species (10,568 *Alu* insertions, ~10% of the dataset), and three clusters consisted of insertions shared between *P. kindae*, and at least one of the northern baboons (6,226 *Alu* insertions, ~6% of the dataset).

Northern and southern clade phylogenies were then re-evaluated using all twelve *Papio* baboons: two from each of the six extant *Papio* species, with all 187,379 *Alu* insertions. *Alu* insertions shared exclusively between multiple individuals belonging to the same clade were
classified as clade-specific markers. A total of 95,703 such markers were identified: 39,795 in the northern clade and 55,908 in the southern clade. These markers were clustered based on precise presence/absence genotypes determined for all twelve Papio baboons. Species indicative markers were defined as Alu insertions present in both individuals representing the same species, yet absent from all other members on the panel. In total we identified 48,808 species indicative markers: 23,578 markers were identified in the northern clade, with 25,230 identified in the southern clade.

**Figure 2.3.** Common patterns of shared Alu insertion polymorphisms. (A) The number of Alu insertions shared exclusively between the species highlighted in each row. Markers were clustered based on precise presence/absence genotype data determined for six Papio baboons: one representing each Papio species. This figure displays the fifteen largest clusters identified in this analysis. The colors correspond to the (B) Geographical distributions of the six Papio species. Map extrapolated from (Zinner et al., 2009). White/empty boxes indicate an empty site in that species.
Figure 2.4. Analysis of phylogenetically informative Alu insertions. (A) Species indicative Alu insertion polymorphisms. For each species, the total number of Alu insertion polymorphisms shared exclusively between individuals belonging to that species. All species indicative markers were identified in multiple representative individuals. Also displayed is the number of Alu insertion polymorphisms supporting alternative northern (B) and southern (C) clade phylogenies. These markers were shared between multiple individuals belonging to each of the sister taxa displayed, yet absent from the third divergent species. Each phylogeny corresponds to the data point above it.

The total number of species indicative markers determined for each Papio species is displayed in Figure 2.4A. Among northern baboon species, the highest number of species indicative Alu polymorphisms was determined for P. papio (10,873), followed by P. hamadryas (8,060) and P. anubis (4,645). In the southern clade, P. kindae reported the highest number of species indicative markers (12,891), followed by P. ursinus (9,545), and P. cynocephalus (2,794). Furthermore we evaluated inter-species relationships by targeting clade-specific markers shared between all individuals belonging to two species within a clade, yet absent from both individuals from the remaining species. Within both clades, three unique clusters were formed from these data, each supporting a different clade phylogeny (Figure 2.4 B and C). A total of 7,436 such elements was determined: 4,220 in the northern clade and 3,216 in the southern clade.
Of the markers identified in the northern clade, 52% were shared exclusively between *P. anubis* and *P. papio* (1613 loci), 34% were shared between *P. anubis* and *P. hamadryas* (1153 loci), and the remaining 14% were shared between *P. papio* and *P. hamadryas* (450 loci). In the southern clade analysis, 43% of the Alu insertions were shared between *P. ursinus* and *P. kindae* (1766 loci), 36% were shared between *P. cynocephalus* and *P. ursinus* (1483 loci), and 28% were shared between *P. kindae* and *P. cynocephalus* (971 loci).

**Figure 2.5.** Low allele frequency Alu insertions polymorphic among *Papio* species. A diversity panel of twelve *Papio* baboons was used in this analysis: two representing each extant species. The only elements used in this analysis were those shared uniquely between two species. Each pie chart represents the average values determined from the two individuals representing that *Papio* species (the species name is listed above each pie chart). The size of every pie chart slice represents the number of Alu insertions shared between the species listed above that particular chart and the species represented by the color of the slice (indicated by the legend on the right). The numbers outside each pie chart correspond to the total number of Alu insertions represented by each slice.

In addition, we evaluated low-allele frequency Alu polymorphisms using data obtained from our complete panel of twelve individuals: two representing each *Papio* species. Alu insertions used in this analysis were those shared uniquely between only two species, and absent from the other four. Thus the overall number of these insertions among *Papio* was relatively low.
We clustered these elements based on their precise presence/absence genotypes. Clusters identified for each species are displayed in Figure 2.5.

The numbers of insertions listed correspond to the average of the two individuals from each species. With the exception of *P. papio* and *P. hamadryas*, the largest clusters identified in *Papio* species contained *Alu* insertions shared between individuals belonging to the same clade (north/south). Although the single largest cluster identified in both *P. papio* and *P. hamadryas* consisted of elements shared with *P. anubis*, the second largest cluster was shared with *P. kindae*. All of the northern baboons shared more insertions with *P. kindae* than with the other two southern baboon species combined (*P. cynocephalus* and *P. ursinus*).

**Discussion**

With the increasing availability of WGS data, admixture remains a fundamental challenge for evolutionary biologists. Nevertheless, the abundance of genomic data provides scientists the opportunity to use novel methodologies to re-examine complex evolutionary relationships. Well-documented extant hybrid zones coupled with a dense history of reticulation complicate the task of neatly organizing *Papio* baboons into a phylogenetic tree. Baboons are popular well-established research models for studying human disease and evolution, and therefore understanding the pattern of genetic variation within and between baboon species is important. As a result, an accurate and detailed understanding of *Papio* genomic evolution is quite valuable.

Despite the increasing availability of WGS data, high quality assemblies are not commonly constructed for multiple species belonging to the same genus. Instead, one individual is often used to build an assembly representative of an entire genus. However, often times WGS data are generated from individuals belong to different species within that genus. For *Papio*
baboons, a high quality (chromosome-level resolution) reference assembly exists only for *Papio anubis*, yet WGS data have been generated for multiple individuals from each extant *Papio* species. A traditional method used to identify *Alu* elements polymorphic within a genus involved identifying markers present in an assembly of interest, yet absent from the closest primate relative with a draft assembly. For *Papio* baboons, a lineage-specific *Alu* polymorphism would be defined as an element present in *P. anubis*, yet absent in rhesus macaques [as represented by the assembly Mmul8.0.1]. Since all of the subsequent markers would be identified in a *P. anubis* individual, this would introduce sampling bias towards markers present in *P. anubis*. However, our computational approach allowed us to align all of our representative *Papio* samples against the outgroup rhesus macaque [Mmul8.0.1], placing equal evolutionary distance between each *Papio* individual and the reference assembly. As a result, we were able to identify polymorphic *Alu* elements with minimal directional bias.

Analyses conducted using mitochondrial DNA support the most basal divergence of *Papio* into northern and southern clades. However, these analyses were unable to produce a phylogeny that fully resolved evolutionary relationships between *Papio* species. Our findings provide support for this basal north-south split hypothesis. Furthermore, this study produces the first whole genome computational analysis of *Alu* polymorphisms within *Papio*. By designing a computational method to detect and characterize *Alu* polymorphisms from multiple *Papio* individuals representing all known extant species and evaluating various basal divergence models, we were able to produce a fully resolved phylogeny of *Papio* baboons with 100% bootstrap support at each node.

In addition, our analysis of elements discordant with this phylogenetic model may offer insights into a complex history of admixture and reticulation within the *Papio* lineage. In the
southern lineage, *P. kindae* shows the highest incidence of *Alu* insertions shared with the northern clade, yet absent from the other southern clade samples (11,286 elements). In total, we identified 64,259 elements discordant with topology of the phylogenetic tree (Figure 2.2) that could be due to incomplete lineage sorting (ILS) or hybridization/admixture. Continued analyses involving a greater number of individuals would be necessary to accurately explain the taxonomic distribution of these insertions. Such analyses could potentially elucidate insertions indicative of speciation, the north-south split, hybridization, and many other evolutionary events. Thus, the data presented in this paper may be utilized to further evaluate *Papio* evolution. Such studies are likely necessary given the rich diversity that exists within the genus *Papio*. Furthermore, this approach has outstanding potential to inform analyses of other primate genera with complex evolutionary histories (e.g. *Cercopithecus, Macaca, Chlorocebus, Aotus, Microcebus, Saimiri* and others).

Contemporary arguments in favor of applying a phylogenetic species concept to the *Papio* genus rely heavily on the rich species diversity exhibited between morphotypes. Our findings provide support for the genetic diversity that exists within the genus *Papio*. In each extant species, we found an average of over 8,000 elements shared exclusively between members belonging to that species. Despite previous debate as to whether *P. kindae* warrants species level classification, the largest number of species-specific elements characterized in this study were identified in *P. kindae* (12,891).

One limitation of this study is that it is based on only twelve *Papio* individuals: two representing each species. It is very likely that the genetic diversity observed in each individual does not comprehensively represent diversity existing with the species as a whole. Each wild *Papio* species occupies a large range across the African continent; thus proximity to hybrid zones
may contribute to interspecies diversity that is not captured in this analysis. Several species occupy ranges that contact other *Papio* species (Figure 2.3B). Little is known about within species diversity. Only through further large-scale sampling and analyses can this be evaluated.

**Conclusions**

In conclusion, this study exhibits the utility and efficacy of a whole genome analysis of *Alu* polymorphisms for resolving controversial phylogenetic relationships. In addition, it demonstrates the importance of employing diverse methodologies. Knowledge of the initial divergence of *Papio* into northern and southern clades, produced by previous studies and supported in this study, was instrumental in our analysis of *Papio* evolution. Despite high incidence of hybridization and sustained hybrid zones, we were able to produce a highly supported cladogram, resolving relationships within both the northern and southern clades. These data represent the most comprehensive *Alu*-based phylogenetic reconstruction reported to date. In addition, this study also produces the first fully resolved *Alu*-based phylogeny of *Papio* baboons. Our approach may offer useful applications for investigating other unresolved branches of the primate evolutionary tree.

**References**


doi:10.1016/j.cub.2008.02.074


CHAPTER THREE: COMPUTATIONAL ANALYSIS OF EVOLUTIONARY RELATIONSHIPS WITHIN PAPIONINA USING ALU POLYMORPHISMS

Background

The phylogenetic position of the gelada baboon has been debated since it was first identified in the 1800’s (J. Cronin & Meikle, 1979). Originally named *Macacus gelada* in 1835, it was later placed in a genus of its own (J. Cronin & Meikle, 1979; J. E. Cronin & Meikle, 1982) were it remains today as the only extant species belonging to the genus *Theropithecus*. Furthermore, *Theropithecus* belongs to a larger *Papionini* subtribe known as *Papionina*. This subtribe contains *Papio* baboons, *Theropithecus gelada*, as well the tree-dwelling arboreal mangabeys belonging to the genus *Lophocebus*. Phylogenetic relationships between the three *Papionina* genera have not been resolved (Guevara & Steiper, 2014; Liedigk, Roos, Brameier, & Zinner, 2014). Some studies have placed Theropithecus basal to a clade consisting of *Papio* and *Lophocebus* (Guevara & Steiper, 2014; Harris, 2000), while other analyses have placed *Theropithecus* and *Papio* as sister taxa, with *Lophocebus* diverging first (Liedigk et al., 2014).

The fact that extensive molecular evidence has yet to resolve this trichotomy implicates a complex history of admixture and reticulation with possible inter-genera hybridization among the lineages. Observations of hybridization in the wild between *Theropithecus gelada* males and *Papio anubis* females have been reported (Dunbar & Dunbar, 1974; Jolly, Woolley-Barker, Shimelis, Disotell, & Phillips-Conroy, 1997). A similar complex history of evolution has been reported among species within the genus *Papio* (Jolly, 2001; Jordan et al., 2018; Newman, Jolly, & Rogers, 2004; Steely et al., 2017; Szmulewicz et al., 1999; Walker et al., 2017; Zinner et al., 2009; Zinner et al., 2013) in which mitochondrial and morphological based phylogenies could not fully resolved the placement of the all six currently recognized extant species.
We recently reported a computational reconstruction of *Papio* phylogeny using *Alu* insertion polymorphisms which employed a large scale WGS analysis of over 187,000 *Alu* events to produce a fully resolved phylogeny (Jordan et al. 2018). This method not only determined the most likely branching order within *Papio* with 100% bootstrap support, but also identified the number of *Alu* insertion events supporting alternative topologies, demonstrating the efficacy of whole genome analysis of *Alu* polymorphisms for resolving controversial phylogenetic relationships among primates. It is likely that an extensive analysis of this type is required to possibly help resolve the phylogeny within *Papionina*. Although whole genome sequence data is not currently available for individuals of the genus *Lophocebus*, it is available for one Theropithecus gelada. Therefore, as proof of concept, we compared *Alu* insertion polymorphism between *Theropithecus gelada* and the *Papionini* panel utilized the Jordan et al. (2018) study. *Alu* insertion events recent enough to remain polymorphic among *Papio* species would be expected to have integrated after the split from *Theropithecus* and therefore be absent from *Theropithecus*. Conversely, *Theropithecus* would be expected to have lineage specific insertions. Alu insertions present in both genera would be expected to be fixed present in all species. Observations contrary to these hypotheses would implicate incomplete lineage sorting, hybridization or both between the genera.

**Methods**

**Samples**

Whole-genome sequencing was performed by the Baylor College of Medicine Human Genome Sequencing Center on a panel of *Papionini* individuals. The same dataset described in the methods section of chapter two was used in this analysis (Jordan et al., 2018): two individuals from each of the six extant *Papio* species (we randomly selected two individuals
from *P. anubis* and *P. kindae*) to conduct our computational analysis; along with WGS data from the macaque sample used to build the latest *M. mulatta* assembly [Mmul8.0.1]. Unlike the experiment described in chapter two, we included WGS data determined for one *Theropithecus gelada*.

WGS data were accessed from the NCBI-SRA database (Coordinators, 2016). The SRA-toolkit (fastq-dump utility) (Leinonen et al., 2011) was used to download paired-end next generation sequencing reads and convert them from .sra files to interleaved fastq files. Nesoni was then utilized to [https://github.com/Victorian-Bioinformatics-Consortium/nesoni; last accessed March 2018](https://github.com/Victorian-Bioinformatics-Consortium/nesoni; last accessed March 2018) to prune all known adapters, cleave bases with a phred quality score of 10 or lower, and exclude reads shorter than 24 base pairs in length. Two output fastq files were produced: one containing paired-end reads where both reads passed the nesoni filter, and a second containing unpaired orphan reads where nesoni excised one of the paired-end reads.

**Polymorphic Alu Insertion Detection**

We used the computational pipeline reported in Jordan et al. (2018) to perform our analysis. Our approach targeted young *Alu* insertions present in *T. gelada* yet polymorphic within *Papio*. The approximate chromosomal position of each candidate insertion was estimated using a split-read method (Figure 2.1). The resulting genotypes, generated for all individuals in our panel, isolated phylogenetically informative markers.

In the alignment phase, we used BWA mem (H. Li, 2013) to map paired-end NGS reads to a consensus *Alu*Y sequence obtained from Repbase (Jurka, 1998). Reads mapping to the head of an *Alu* insertion were required to contain at least 15 bp of unmapped/non-*Alu* sequence directly upstream of the (5’) start of the *Alu* sequence. Likewise, reads mapping to the tail of the consensus *Alu* sequence were required to contain no less than 15 bases of unmapped sequence.
directly flanking the (3’) end of the sequence. We performed this mapping step twice: once using the standard BWA mem parameters, and a second time using more liberal parameters (described in Table A.2). Prediction made using the standard parameters were ultimately used to predict the location of an Alu integration site, with those identified during the liberal run providing additional support for an insertion event. The Alu portion of each candidate split-read was then cleaved and remaining sequence aligned to Mmul8.0.1 using bowtie2 (Langmead & Salzberg, 2012). Split-reads were categorized as sequences that mapped uniquely to the AluY consensus and the Mmul8.0.1 assembly. We identified 1,863 candidate loci meeting this criteria.

**Oligonucleotide Primer Design and Alu Selection**

From the 1,863 candidate loci, we randomly selected a total of 155 genomic loci. Oligonucleotide primers for PCR were also designed using the rhesus macaque genome [rheMac8]. These loci were then tested using the In-Silico PCR function of BLAT (Kent, 2002) through the University of California Santa Cruz (USCS) Genome Browser. We used Insilco-PCR (Kent, 2002) tell vet our primer pairs. Using this software we obtained estimates for our expected PCR product sizes in rheMac8 and papAnu2.

**DNA Panel**

The remaining loci were evaluated on a DNA panel including three *P. anubis*, two *P. ursinus*, two *P. papio*, two *P. kindae*, two *P. cynocephalus*, one *P. hamadryas*, and two *T. gelada*. A human (*HeLa*) sample was used as a positive control and TLE was used as a negative control. Information about the samples is provided in Table A.6 including their common name, origin, and ID.
Genotype Analysis and Allele Frequency

PCR was used to genotype each candidate locus in our DNA panel. This data is displayed in S.6 spreadsheet was used to analyze the PCR genotypes at each candidate Alu locus. The genotypes were assigned the following in the spreadsheet: “0:0” for homozygous absent (no Alu insertion), “1:1” for homozygous present (Alu insertion of both copies), “1:0” for heterozygous present (Alu insertion of one copy), and “-9:-9” for no band detected (no amplification). These genotypes were entered for every locus evaluated then used to calculate the allele frequency in each population.

Results

Allele Frequency and Alu Subfamily Analysis

A total of 105 loci were tested by PCR for an Alu insertion in Papio and T. gelada. Out of the 105 loci tested for insertions, 102 loci generated interpretable results. The allele frequency results determined that 93% of the amplified loci in Papio had an allele frequency less than one for the Alu insertion, meaning it had not reached fixation in the Papio population. Meanwhile, 52% of the amplified loci in gelada had an allele frequency greater than zero for the Alu insertion. These results are generally consistent with and support the initial computational method for identifying polymorphic Alu insertions.

The data were evaluated to isolate Alu insertions polymorphic in both genera, revealing 28 such loci. A list of these 28 loci along with their PCR primers, genomic coordinates is shown in Table A.6. Of these 28, only 4 were present in the papAnu2 reference genome from which the complete Alu sequence was obtained. After evaluating these Alu sequences using RepeatMasker, we determined that the insertions belonged to old AluY subfamilies present in M. mulatta, meaning that these subfamilies were active prior to the divergence of Papionina from Macaca.
One interesting finding was that the elements were shared randomly throughout the *Papio* species and *T. gelada*. There was no indication of the *Alu* marker being shared specifically between one *Papio* species and gelada, which would suggest an ancestral hybrid zone unequally shared *T. gelada* and all six *Papio* species.

**Discussion**

In this analysis, we computationally identified 1,863 *Alu* insertions polymorphic between *Theropithecus* and *Papio*. Our findings suggest that it is likely that a sufficient number of polymorphic *Alu* insertions exist within *Papionina* to resolve its phylogeny. When WGS data is generated for *Lophocebus*, the computational method described in Jordan et al. 2018 can be utilized to resolve the *Papionina* polytomy. Likewise, if additional *T. gelada* samples are sequenced along with multiple *Lophocebus* individuals, we would also be able to estimate genus-specific rates of *Alu* mobilization. Lastly, we would be able to evaluate patterns indicative of incomplete lineage sorting and hybridization using the basal divergence model. We then could isolate elements conflicting with the most well-supported tree topology to uncover patterns of inter-genera admixture. Ultimately, the resulting data could greatly augment pre-existing body of research previously generated on *Papionina* evolution.

**References**


CHAPTER FOUR: CONCLUSIONS

Incomplete lineage sorting and admixture present fundamental challenges for phylogenetic analyses. Although one of the functional goals of taxonomy is to enable meaningful communication and collaboration between different researchers, the ultimate objective should be to recreate and represent systems as they exist in nature. With regards to phylogeny, the goal should be simply to “resolve” problems created by admixture, but to reveal its evolutionary footprint. For many organisms, including humans, admixture and hybridization have played an important role in genome evolution. To adequately gauge the impact, however, requires an integrative approach.

The computational method utilized in this paper is unique in that it is the first phylogenetic tool purposed to identify retrotransposons indicative of admixture. By constructing a dataset comprised of Alu insertions polymorphic within your data, it uses these markers to evaluate all possible basal divergence topographies. The model best supported by the data is then used identify all of the polymorphic insertions exhibiting computational derived genotypes that conflict with the basal divergence model. The resulting data can be used to identify patterns of cross-clade shared Alu polymorphisms prevalent in excess of what could reasonably be attributed to incomplete lineage sorting.

Future directions include adding a feature that facilitates de-novo reconstruction of individual Alu sequences obtained from WGS data. Successful sequence reconstruction would allow for sub-family characterization. In theory, this information could provide insights into the temporal landscape of different retrotranspositional events. However, one major obstacle to the efficacy of such a tool is the quality and quantity of next-generation sequencing reads spanning the entire length of the insertion. Additionally, it could be useful to see this method extended to L1s, SVAs, ERVs, and other TEs active during recent periods of genome evolution. As the
diversity and quality of sequenced genomes continues to increase, so does the need for comprehensive integrative tools and methodologies.
APPENDIX A: SUPPLEMENTAL DATA

Table A.1. Sequencing information for the thirteen WGS samples used in this study. Individuals listed in bold indicate the panel of *Papio* samples used to conduct the clustering analysis in which one representative sample was used for each species.

<table>
<thead>
<tr>
<th>WGS samples</th>
<th>Species</th>
<th>Isolate</th>
<th>Source</th>
<th>Sex</th>
<th>Platform</th>
<th>Runs</th>
<th>Bytes</th>
<th>Bases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Macaca mulatta</em></td>
<td>17573</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>22</td>
<td>56.2</td>
<td>102.3</td>
</tr>
<tr>
<td></td>
<td><em>Papio anubis</em></td>
<td>LIV5</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>6</td>
<td>52.6</td>
<td>92.89</td>
</tr>
<tr>
<td></td>
<td><em>Papio anubis</em></td>
<td>L142</td>
<td>Blood</td>
<td>Male</td>
<td>Illumina</td>
<td>6</td>
<td>42</td>
<td>77.35</td>
</tr>
<tr>
<td></td>
<td><em>Papio cynocephalus</em></td>
<td>16066</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>7</td>
<td>56.3</td>
<td>92.69</td>
</tr>
<tr>
<td></td>
<td><em>Papio cynocephalus</em></td>
<td>16098</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>6</td>
<td>41.7</td>
<td>76.68</td>
</tr>
<tr>
<td></td>
<td><em>Papio hamadryas</em></td>
<td>97074</td>
<td>Blood</td>
<td>Not determined</td>
<td>Illumina</td>
<td>5</td>
<td>48.3</td>
<td>80.3</td>
</tr>
<tr>
<td></td>
<td><em>Papio hamadryas</em></td>
<td>97124</td>
<td>Blood</td>
<td>Not determined</td>
<td>Illumina</td>
<td>4</td>
<td>49.3</td>
<td>77.16</td>
</tr>
<tr>
<td></td>
<td><em>Papio kindae</em></td>
<td>34449</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>5</td>
<td>84.5</td>
<td>131.2</td>
</tr>
<tr>
<td></td>
<td><em>Papio kindae</em></td>
<td>34474</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>4</td>
<td>76.9</td>
<td>117.5</td>
</tr>
<tr>
<td></td>
<td><em>Papio papio</em></td>
<td>28547</td>
<td>Blood</td>
<td>Not determined</td>
<td>Illumina</td>
<td>5</td>
<td>61.5</td>
<td>100.4</td>
</tr>
<tr>
<td></td>
<td><em>Papio papio</em></td>
<td>30388</td>
<td>Blood</td>
<td>Male</td>
<td>Illumina</td>
<td>5</td>
<td>60.7</td>
<td>98.85</td>
</tr>
<tr>
<td></td>
<td><em>Papio ursinus</em></td>
<td>28697</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>5</td>
<td>44.7</td>
<td>79.01</td>
</tr>
<tr>
<td></td>
<td><em>Papio ursinus</em></td>
<td>28755</td>
<td>Blood</td>
<td>Female</td>
<td>Illumina</td>
<td>8</td>
<td>48.6</td>
<td>83.38</td>
</tr>
</tbody>
</table>
Table A.2. Computational pipeline command line arguments. An outline detailing the programs utilized in the computational pipeline. Command line arguments used in each run are provided.

<table>
<thead>
<tr>
<th>Command Line Arguments</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I. Fastq-dump</strong></td>
<td>i. fastq-dump --split-spot -Z [accession number] &gt; your.fastq</td>
</tr>
<tr>
<td><strong>II. Nesonl</strong></td>
<td>i. nesonl clip: clipped --gzip no interleaved: your.fastq</td>
</tr>
</tbody>
</table>
| **III. BWA**          | i. Standard run:  
|                        |   i. bwa mem -t 8 -C path2ref path2fq > path2out.bam  
|                        |   ii. Liberal run:  
|                        |     i. bwa mem -t 8 -p -C -k 10 -T 15 path2ref path2fq > path2out.bam |
| **IV. bowtie2**       | i. bowtie2 [options] -p 8 -very-sensitive -x <path2ref_index> -U path2fq -S outputpath |

**BWA OPTIONS [default]:**
- t INT  Number of threads [1]
- k INT  Minimum seed length. Matches shorter than INT will be missed. The alignment speed is usually insensitive to this value unless it significantly deviates 20. [19]
- p     Assume the first input query file is interleaved paired-end FASTA/Q. See the command description for details.
- T INT  Don’t output alignment with score lower than INT. This option only affects output. [30]
- C     Append FASTA/Q comment to SAM output. This option can be used to transfer read meta information (e.g. barcode) to the SAM output. Note that the FASTA/Q comment (the string after a space in the header line) must conform to the SAM spec (e.g. BC:Z:CGTAC). Malformed comments lead to incorrect SAM output.

**Bowtie2 OPTIONS:**
Main arguments
- x <bt2-idx> The basename of reference genome index. We used bowtie2 to index the reference genome.
- U <> Path to input fastq.
- S <sam> Path to output SAM file
- D <int> Max <int> consecutive seed extension attempts
- R <int> Max <int> attempt to “re-seed” reads with repetitive seeds
- N <int> Max <int> allowed mismatches
- L <int> sets the length of the seed substrings
- i <int> the interval between seed substring

---very-sensitive option is the same as running with options: -D 20 -R 3 -N 0 -L 20 -i S,1,0.50.
Table A.3. Statistical analysis of *Papio* basal divergence models. The values listed correspond to the 31 possible phylogenetic models displayed in Figure A.1. For each model, the number of concordant insertions are provided in the third column; the number of discordant insertions can be found in the fourth column. The z-score determined for the number of discordant insertions is listed in the last column. The lowest z-score (indicating smallest proportion of discordant elements in group) is shown in bold font and indicates scenario III-A to be the most likely basal divergence model.

<table>
<thead>
<tr>
<th></th>
<th>Concordant</th>
<th>Discordant</th>
<th>Z-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>A 56424</td>
<td>8523</td>
<td>0.292688</td>
</tr>
<tr>
<td>I</td>
<td>B 57072</td>
<td>7875</td>
<td>-0.40374</td>
</tr>
<tr>
<td>I</td>
<td>C 58112</td>
<td>6835</td>
<td>-1.52148</td>
</tr>
<tr>
<td>I</td>
<td>D 56821</td>
<td>8126</td>
<td>-0.13398</td>
</tr>
<tr>
<td>I</td>
<td>E 56484</td>
<td>8463</td>
<td>0.228203</td>
</tr>
<tr>
<td>I</td>
<td>F 55265</td>
<td>9682</td>
<td>1.538313</td>
</tr>
<tr>
<td>II</td>
<td>A 55537</td>
<td>9410</td>
<td>-1.30308</td>
</tr>
<tr>
<td>II</td>
<td>B 55004</td>
<td>9943</td>
<td>-1.03091</td>
</tr>
<tr>
<td>II</td>
<td>C 56546</td>
<td>8401</td>
<td>-1.81831</td>
</tr>
<tr>
<td>II</td>
<td>D 51607</td>
<td>13340</td>
<td>0.703724</td>
</tr>
<tr>
<td>II</td>
<td>E 51278</td>
<td>13669</td>
<td>0.871723</td>
</tr>
<tr>
<td>II</td>
<td>F 54501</td>
<td>10446</td>
<td>-0.77406</td>
</tr>
<tr>
<td>II</td>
<td>G 52092</td>
<td>12855</td>
<td>0.456065</td>
</tr>
<tr>
<td>II</td>
<td>H 51674</td>
<td>13273</td>
<td>0.669511</td>
</tr>
<tr>
<td>II</td>
<td>I 50500</td>
<td>14447</td>
<td>1.268998</td>
</tr>
<tr>
<td>II</td>
<td>J 50820</td>
<td>14127</td>
<td>1.105595</td>
</tr>
<tr>
<td>II</td>
<td>K 52467</td>
<td>12480</td>
<td>0.264576</td>
</tr>
<tr>
<td>II</td>
<td>L 55389</td>
<td>9558</td>
<td>-1.2275</td>
</tr>
<tr>
<td>II</td>
<td>M 53807</td>
<td>11140</td>
<td>-0.41968</td>
</tr>
<tr>
<td>II</td>
<td>N 52160</td>
<td>12787</td>
<td>0.421342</td>
</tr>
<tr>
<td>II</td>
<td>O 51395</td>
<td>13552</td>
<td>0.811979</td>
</tr>
<tr>
<td>III</td>
<td>A 58017</td>
<td>6930</td>
<td><strong>-2.72828</strong></td>
</tr>
<tr>
<td>III</td>
<td>B 51544</td>
<td>13403</td>
<td>0.184783</td>
</tr>
<tr>
<td>III</td>
<td>C 50859</td>
<td>14088</td>
<td>0.493056</td>
</tr>
<tr>
<td>III</td>
<td>D 51660</td>
<td>13287</td>
<td>0.13258</td>
</tr>
<tr>
<td>III</td>
<td>E 52162</td>
<td>12785</td>
<td>-0.09334</td>
</tr>
<tr>
<td>III</td>
<td>F 51388</td>
<td>13559</td>
<td>0.254988</td>
</tr>
<tr>
<td>III</td>
<td>G 52105</td>
<td>12842</td>
<td>-0.06768</td>
</tr>
<tr>
<td>III</td>
<td>H 50922</td>
<td>14025</td>
<td>0.464704</td>
</tr>
<tr>
<td>III</td>
<td>I 50116</td>
<td>14831</td>
<td>0.82743</td>
</tr>
<tr>
<td>III</td>
<td>J 50773</td>
<td>14174</td>
<td>0.531759</td>
</tr>
</tbody>
</table>
**Table A.4.** The number of Alu insertions shared exclusively between the species highlighted in each row. Markers were clustered based on precise presence/absence genotype data determined for six *Papio* baboons: one representing each *Papio* species. This figure displays the fifteen largest clusters identified in this analysis. The colors correspond to the geographical distributions of the six *Papio* species displayed in Figure 2.3B. White/empty boxes indicate an empty site in that species.

<table>
<thead>
<tr>
<th>Complete Cluster List</th>
<th>P. <em>papio</em></th>
<th>P. <em>anubis</em></th>
<th>P. <em>hamadryas</em></th>
<th>P. <em>cynocephalus</em></th>
<th>P. <em>kindae</em></th>
<th>P. <em>ursinus</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Markers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32156</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4509</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4338</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4326</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4238</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3705</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3670</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3392</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2806</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2609</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2301</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2269</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2042</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1988</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1915</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1915</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1728</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1338</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1307</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1197</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1157</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1132</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>959</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>925</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>842</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>830</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>774</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>754</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>747</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>746</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>731</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>715</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>703</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>671</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A.5. Python script used to run the computational whole genome analysis. Using this script we identified the polymorphic Alu insertions use in the analyses described in chapter two and three.

```python
# -*- coding: utf-8 -*-

Spyder Editor
This is a temporary script file.

import sys
import re
from sam import *
from essentials import *
from meiDetect import *
from sra import *
import linecache
# import numpy as np
import os
from subprocess import Popen, PIPE
from timeit import default_timer as timer
from polyAnna import *

path2aluConsensus, path2referenceGenome, path2_PE_fq, path2_SE_fq, path2outdir, orgID, path2polyA = sys.argv[1], sys.argv[2], sys.argv[3], sys.argv[4], sys.argv[5], sys.argv[6], sys.argv[7]
```
def editFQ_title(inpath,outpath):
    with open(inpath,'r') as infastq:
        with open(outpath,'w') as outfastq:
            for line in infastq:
                if line[:4]=='@SRR':
                    outfastq.write('@'+str(zeroThehero(12,(line.split()[0].split('.')[1])))+'	'+line.split()[1]+'
')
                else:
                    outfastq.write(line)

def bwaMeM(path2ref,path2fsq,P_S,S_L,outpath):
    """The Purpose of This is to Execute Bwa Mem""
    with open(outpath,'w') as out:
        if S_L=='S': #This means that we are using the BWA standard parameters
            if P_S=='P':
                subprocess.call(['bwa', 'mem', '-t', '8', '-p', '-C', path2ref, path2fsq],stdout=out)
            elif P_S=='S':
                subprocess.call(['bwa', 'mem', '-t', '8', '-C', path2ref, path2fsq],stdout=out)
        elif S_L=='L': #This means that we are using the BWA more liberal parameters
            if P_S=='P':
                subprocess.call(['bwa', 'mem', '-t', '8', '-p','-C','-k','10','-T','15', path2ref, path2fsq],stdout=out)
            elif P_S=='S':
subprocess.call(['bwa', 'mem', '-t', '8', '-C', '-k', '10', '-T', '15', path2ref, path2fsq], stdout=out)

def bowtie2(path2ref,path2fsq,P_S,outpath):
    """The Purpose of This is to Execute bowtie2""
    with open(outpath,'w') as out:
        if P_S=='S': #This means that we are using the BWA standard parameters
            subprocess.call(['bowtie2', '-p', '8', '--very-sensitive', '-x', path2ref, '-U', path2fsq, '-S', outpath], stdout=out)
        elif P_S=='P': #This means that we are using the BWA more liberal parameters
            subprocess.call(['bowtie2', '-p', '8', '--very-sensitive', '-x', path2ref, '--interleaved', path2fsq, '-S', outpath], stdout=out)

def mappOnly(in_sam,out_sam):
    with open(in_sam,'r') as inSam:
        with open(out_sam,'w') as outSam:
            for line in inSam:
                if line[0]!='@':
                    if line.split()[2]!=='*':
                        outSam.write(line)

def aluClipped(samline):
    s,tooShort=sam(samline),False
    if (s.cigar!='*') and (int(s.pos)<=270):
        cigar_span,cigar_char=splitCigar(s.cigar)
        position='M'
        if (len(cigar_span)>1) and ('H' not in s.cigar):
mFirst, mLast = cigar_char.index('M'), find_last(cigar_char, 'M')
i, m=mFirst, 0

while i<=mLast:
    m+=int(cigar_span[i])
i+=1

if int(s.pos)<80:
    m+=(int(s.pos)-1)
    position='H'
elif (int(s.pos)+m)>=267:
    if (int(s.pos)+m)>282:
        position='P'
    else:
        position='T'
else:
    m=len(s.seq)
    position='M'
if mFirst==0:
    if (len(s.seq)-m)>=20:
        seq, qual = s.seq[m:], s.qual[m:]
    else:
        tooShort=True
elif mLast==(len(cigar_char)-1):
    if (len(s.seq)-m)>=20:
        seq, qual = s.seq[:len(s.seq)-m], s.qual[:len(s.seq)-m]
    else:
        tooShort=True
else:
    begin, end, b, e = 0, 0, 0, (mLast+1)
while b<mFirst:
    begin+=int(cigar_span[b])
    b+=1
while e<len(cigar_char):
    end+=int(cigar_span[e])
    e+=1
if (len(s.seq)-m)>20:
    if b>e:
        if b>=20:
            seq,qual=s.seq[:b],s.qual[:b]
        else:
            tooShort=True
    elif e>b:
        if e>=20:
            seq,qual=s.seq[:e],s.qual[:e]
        else:
            tooShort=True
    else:
        tooShort=True
else:
    tooShort=True
elif (len(cigar_span)==1):
    mFirst,mLast=cigar_char.index('M'),find_last(cigar_char, 'M')
i,m=mFirst,0
while i<=mLast:
    m+=int(cigar_span[i])
i+=1
if int(s.pos) <= 80:
    m += (int(s.pos) - 1)
    position = 'H'
elif (int(s.pos) + m) >= 267:
    if (int(s.pos) + m) > 282:
        position = 'P'
    else:
        position = 'T'
else:
    m = len(s.seq)
    position = 'M'
    tooShort = True
else:
    tooShort = True
if tooShort == True:
    return ('', '', 'fully_clipped', position)
else:
    if s.orientation == '0':
        return (seq, qual, 'partially_clipped', position)
    elif s.orientation == '1':
        qual = qual[::-1]
        return (reverseComplement(seq), qual, 'partially_clipped', position)
    else:
        if s.cigar != '*':
            if s.orientation == '0':
                return (s.seq, s.qual, 'unclipped', 'N')
            elif s.orientation == '1':
                qual = s.qual[::-1]
```python
return(reverseComplement(s.seq),qual,'unclipped','N')
else:
    return(s.seq,s.qual,'unclipped','N')

def clipOutAluPFU(cigar,pos,seq,qual):
    if cigar=='*':
        return(seq,qual,'unclipped','N')
    elif (int(pos)>270):
        return(seq,qual,'unclipped','N')
    else:
        cigar_span_list,cigar_character_list = splitCigar(cigar)
        mcount,n=0,0
        while n<len(cigar_character_list):
            if cigar_character_list[n]=='M':
                mcount+=int(cigar_span_list[n])
                n+=1
            if int(pos)<=50:
                position='H'
            elif (int(pos)+mcount)>=250:
                if (int(pos)+mcount)>=282:
                    position='P'
                else:
                    position='T'
            else:
                position='M'
        if 'S' not in cigar:
            return('','',fully_clipped',position)
        else:
```
cigar_span_list, cigar_character_list = splitCigar(cigar)

frontEnd, backEnd, f, b, part = 0, 0, -1, False

while cigar_character_list[f] != 'M':
    frontEnd += int(cigar_span_list[f])
    f += 1

while cigar_character_list[b] != 'M':
    backEnd += int(cigar_span_list[b])
    b -= 1

if ((frontEnd - int(pos)) >= 15) and (backEnd == 0):
    part = True
    return (seq[:frontEnd], qual[:frontEnd], 'partially_clipped', position)

elif (backEnd >= 15) and ((int(pos) + (len(seq)) - (backEnd + frontEnd)) >= 267) and (int(pos) <= 270):
    part = True
    return (seq[backEnd:], qual[backEnd:], 'partially_clipped', position)

if part == False:
    return ('', '', 'fully_clipped', position)

---

def processSamA(path2Psam, path2Ssam, reference):
    with open(reference + '.DS.info', 'w') as DS_info:
        with open(reference + '.SP.info', 'w') as SP_info:
            with open(reference + '.DS.fq', 'w') as fastqDS:
                with open(reference + '.SP.fq', 'w') as fastqSP:
                    with open(path2Psam, 'r') as samfileP:
                        with open(path2Ssam, 'r') as samfileS:
                            pe, ds, sp = 1, 1, 1
                            temp, n = [], 0
for line in samfileP:
    if line[0] != '@':
        if n != 0:
            if line.split()[0] != n:
                if len(temp) == 2:
                    for t in temp:
                        if 'OP:i:1' in t:
                            fwd_seq, fwd_qual, fwd_tag, positionF = clipOutAluPFU(t.split()[5], t.split()[3], t.split()[9], t.split()[10])
                            forward_read = t
                            elif 'OP:i:2' in t:
                                rvs_seq, rvs_qual, rvs_tag, positionR = clipOutAluPFU(t.split()[5], t.split()[3], t.split()[9], t.split()[10])
                                reverse_read = t
                                posPlus = positionF + positionR
                                if (posPlus != 'NN'):
                                    if (fwd_tag != 'fully_clipped') and (rvs_tag != 'fully_clipped') and (posPlus != 'NN'):
                                        nothin = 'nothin'
                                        elif (fwd_tag == 'fully_clipped'):
                                            if rvs_tag == 'partially_clipped':
                                                rSam = sam(reverse_read)

fastqSP.write('@' + zeroThehero(12, sp) + '
+rvs_seq+''\n''+rvs_qual+''\n''+rvs_tag+''\n''')

SP_info.write(zeroThehero(12, sp) + ''t''+rSam.qname+''t''+rSam.pos+''t''+rSam.cigar+''t''+rSam.seq+''t''+positionR+''t''+rSam.orientation+''\n'')

sp += 1
elif rvs_tag=='unclipped':
    fSam=sam(forward_read)

    fastqDS.write('@'+zeroThehero(12,ds)+'

        rvs_seq

+n+rvs_qual+n+


DS_info.write(zeroThehero(12,ds)+'	'+fSam.qname+'	'+fSam.pos+'	'+fSam.cigar+'	'+fSam.seq+'	'+'positionF'+'t'+fSam.orientation+'
')

ds+=1

    if rvs_tag=='fully_clipped':

        if fwd_tag=='partially_clipped':

            fSam=sam(forward_read)

            fastqSP.write('@'+zeroThehero(12,sp)+'

        SP_info.write(zeroThehero(12,sp)+'	'+fSam.qname+'	'+fSam.pos+'	'+fSam.cigar+'	'+fSam.seq+'	'+'positionR'+'t'+fSam.orientation+'
')

        sp+=1

            elif fwd_tag=='unclipped':

                rSam=sam(reverse_read)

                fastqDS.write('@'+zeroThehero(12,ds)+'

        DS_info.write(zeroThehero(12,ds)+'	'+rSam.qname+'	'+rSam.pos+'	'+rSam.cigar+'	'+rSam.seq+'	'+'positionR'+'t'+rSam.orientation+'
')

        ds+=1

        temp=[line.strip('
')]

        n=line.split()[0]

        else:

            temp.append(line.strip('
'))

            n=line.split()[0]
else:

    temp.append(line.strip('\n'))

n=line.split()[0]

for sline in samfileS:

    if sline[0]!('@'):

        fwd_seq,fwd_qual,fwd_tag,positionF=clipOutAluPFU(sline.split()[5],sline.split()[3],sline.split()[9],sline.split()[10])

        if fwd_tag=='partially_clipped':

            fSam=sam(sline)

            fastqSP.write('\n'+zeroThehero(12,sp)+'\n+fwd_seq+\n+fwd_tag+\n+fwd_qual+\n')

            SP_info.write(zeroThehero(12,sp)+'t'+fSam.qname+'t'+fSam.pos+'t'+fSam.cigar+'t'+fSam.seq+'t'+positionF+'t'+fSam.orientation+'\n')

            sp++1

def processSam1(path2Psam,path2Ssam,reference):

    '''The Purpose is to: 1.) Condense Each Line by Removing the SEQ and QUAL 2.) Separate the Forward and Reverse 3.) Parse out only the mapped reads'''

    with open(reference+'.PE.fq','w') as fastqPE:

        with open(reference+'.PE.info','w') as PE_info:

            with open(path2Psam,'r') as samfileP:

                with open(path2Ssam,'r') as samfileS:

                    with open(reference+'_fwd.dup.sam','w') as fwd_dup:

                        with open(reference+'_rvs.dup.sam','w') as rvs_dup:
tempF, tempR, n, mapped, f, r, [ ], [ ], 1, False, θ, θ

p, d, s, p = 1, 1, 1

for line in samfileP:
    if line[0] != '@':
        s = sam(line)
        if int(s.qname) == n:
            if 'OP:i:1' in line:
                tempF.append([line])
                f += 1
                if (s.cigar != '*') and (int(s.pos) <= 270):
                    mapped = True
            elif 'OP:i:2' in line:
                tempR.append([line])
                r += 1
                if (s.cigar != '*') and (int(s.pos) <= 270):
                    mapped = True
            else:
                if mapped:
                    if (f == 1) and (r == 1):
                        fSam, rSam = sam(tempF[0][0]), sam(tempR[0][0])
                        # forward.write(tempF[0][0])
                        # reverse.write(tempR[0][0])

                        fwd_seq, fwd_qual, fwd_tag, positionF = aluClipped(tempF[0][0])

                        rvs_seq, rvs_qual, rvs_tag, positionR = aluClipped(tempR[0][0])

                        posPlus = positionF + positionR
if (fwd_tag!='fully_clipped') and (rvs_tag!='fully_clipped') and (posPlus!='NN'):

fastqPE.write('@'+zeroThehero(12,pe)+'
'+fwd_seq+'
'+fwd_qual+'
'+zeroThehero(12,pe)+'
'+rvs_seq+'
'+rvs_qual+'
')

PE_info.write(zeroThehero(12,pe)+'
'+tfSam.qname+'
'+tfSam.pos+'
'+tfSam.cigar+'
'+tfSam.seq+'
'+positionF+'
'+tfSam.orientation+'
'+zeroThehero(12,pe)+'
'+rfSam.qname+'
'+rfSam.pos+'
'+rfSam.cigar+'
'+rfSam.seq+'
'+positionR+'
')

pe+=1

n,tempF,tempR,mapped,f,r=int(s.qname),[],[],False,0,0

if 'OP:i:1' in line:
    tempF.append([line])
    f+=1
    if s.cigar!='*':
        mapped=True

eelif 'OP:i:2' in line:
    tempR.append([line])
    r+=1
    if s.cigar!='*':
        mapped=True

def matchedReads(info1,info2,read):
    r,i1,i2,match1,match2=sam(read),info(info1),info(info2),False,False
    if (r.seq in i1.seq) or (r.seq in reverseComplement(i1.seq)):
        match1=True
        final=info1
    if (r.seq in i2.seq) or (r.seq in reverseComplement(i2.seq)):
        match2=True
final=info2

if (match1==True) and (match2==True):
    double=True
else:
    double=False
return(final,double)

def mapSpan(cigar):
    span,char=splitCigar(cigar)
    i,mapp=char.index('M'),0
    while i<len(span):
        mapp+=int(span[i])
        i+=1
    return(str(mapp))

def predictLocus(read,in_fo,read_type):
    r,i=sam(read),info(in_fo)
    dist_Alu,dist_ref,aluMap=distanceM(i.cigar),distanceM(r.cigar),0
    alu_span,alu_char=splitCigar(i.cigar)
    alu_index=alu_char.index('M')
    ref_span,ref_char=splitCigar(r.cigar)
    refMap=0
    for char in range(len(ref_char)):
        if ref_char[char]=='M':
            refMap+=int(ref_span[char])
while alu_index<len(alu_char):
    aluMap+=int(alu_span[alu_index])
    alu_index+=1
if read_type!='D':
    if r.orientation==i.orientation:
        orientation='0'
        if int(i.pos)<4:
            start=str(int(r.pos) - int(dist_ref) + int(dist_Alu) - int(i.pos))
return(r.rname+':'+start+'\'+i.pos+'\'+i.flag+'\'+read_type+'\'+i.newName+':'+i.oL
Dname+'\'+i.seq+'\'+orientation)
    elif ((int(i.pos)+aluMap)>267) and (alu_char[-1]!='M'):
        start=r.pos
return(r.rname+':'+start+'\'+i.pos+'\'+i.flag+'\'+read_type+'\'+i.newName+':'+i.oL
Dname+'\'+i.seq+'\'+orientation)
else:
    return('Null')
else:
    if int(i.pos)<4:
        start = str(r.pos)
        orientation='1'
return(r.rname+':'+start+'\'+i.pos+'\'+i.flag+'\'+read_type+'\'+i.newName+':'+i.oL
Dname+'\'+i.seq+'\'+orientation)
    elif ((int(i.pos)+aluMap)>267) and (alu_char[-1]!='M'):
        start=str(int(r.pos)+refMap)
        orientation='1'
return(r.rname+'\':\t'+i.pos+'\t'+i.flag+'\t'+read_type+'\t'+i.newName+'\':\t'+i.oldName+'\t'+i.seq+'\t\'+i.orientation)

else:
    return('Null')
else:
    if r.orientation=='0':
        start=str(int(r.pos)+aluMap)
    if i.orientation=='0':
        orientation='1'
    elif i.orientation=='1':
        orientation='0'
    elif r.orientation=='1':
        start=r.pos
    if i.orientation=='0':
        orientation='0'
    elif i.orientation=='1':
        orientation='1'

return(r.rname+'\':\t'+i.pos+'\t'+i.flag+'\t'+read_type+'\t'+i.newName+'\':\t'+i.oldName+'\t'+i.seq+'\t\'+i.orientation)

def pairedEndPredict(read1,read2,info1,info2):
    r1,r2,i1,i2=sam(read1),sam(read2),info(info1),info(info2)
    predictions=[]
    info_r1,double1=matchedReads(info1,info2,read1)
    info_r2,double2=matchedReads(info1,info2,read2)
    inf1,inf2=info(info_r1),info(info_r2)
    if (int(r1.mapq)>>1) and (int(r2.mapq)>>1):
if ('M' in inf1.cigar):
    if double1==False:
        predictions.append(predictLocus(read1,info_r1,'P'))
if ('M' in inf2.cigar):
    if double2==False:
        predictions.append(predictLocus(read2,info_r2,'P'))
else:
    if (int(r1.mapq)>=1):
        if ('M' in inf1.cigar):
            if double1==False:
                predictions.append(predictLocus(read1,info_r1,'S'))
        if ('M' in inf2.cigar):
            if double2==False:
                predictions.append(predictLocus(read1,info_r2,'D'))
    elif (int(r2.mapq)>=1):
        if ('M' in inf2.cigar):
            if double2==False:
                predictions.append(predictLocus(read2,info_r2,'S'))
        if ('M' in inf1.cigar):
            if double1==False:
                predictions.append(predictLocus(read2,info_r1,'D'))
    if predictions!=[]:
        return(predictions)
else:
    return(['Null'])

def predictInsertionOG(head_tag,outfile):
with open(head_tag+'.PE.sam','r') as pairedEnd:
    with open(head_tag+'.SP.sam','r') as splitRead:
        with open(head_tag+'.DS.sam','r') as discordant:
            with open(outfile,'w') as final:

            temp,finalList,ds_info,sp_info,pe_info=[],[],lister(head_tag+'.DS.info'),lister(head_tag+'.SP.info'),lister(head_tag+'.PE.info')

            for line in pairedEnd:
                if line[0]!='$@

                    if temp==[]:
                        temp.append(line)
                    else:
                        r1,present=sam(temp[0]),sam(line)
                        if r1.qname==present.qname:
                            temp.append(line)
                        elif len(temp)==2:
                            r2=sam(temp[1])

                predictions=pairedEndPredict(temp[0],temp[1],pe_info[(int(r2.qname)*2)-2],pe_info[(int(r2.qname)*2)-1])

                for p in predictions:
                    if p!='$Null$'
                        finalList.append(p)
                        temp=[line]
                    else:
                        temp=[line]

                for line in splitRead:
                    if line[0]!='$@

                        if line.split()[2]!='$\star$'
sprd=sam(line)

if int(sprd.mapq) >= 1:

prediction=predictLocus(line,sp_info[(int(sprd.qname)-1)],'S')

if prediction!='Null':
    finalist.append(prediction)

for line in discordant:
    if line[0]!='@':
        if line.split()[2]!='*':
            dscr=sam(line)
        if int(sprd.mapq) >= 1:

            prediction=predictLocus(line,ds_info[(int(dscr.qname)-1)],'D')

            if prediction!='Null':
                finalist.append(prediction)

            for f in finalist:
                try:
                    final.write(f+'\n')
                except TypeError:
                    print(f,file=sys.stderr)

def predictInsertion(head,tag,outfile):

    with open(head+'.'+tag+'.sam','r') as infile:

        with open(outfile,'a') as final:

            temp,finalist,info=[],[],lister(head+'.'+tag+'.info')

            if tag=='PE':

                for line in infile:

                    if line[0]!='@':
if temp==[]:
    temp.append(line)
else:
    r1,present=sam(temp[0]),sam(line)
    if r1.qname==present.qname:
        temp.append(line)
    elif len(temp)==2:
        r2=sam(temp[1])
predictions=pairedEndPredict(temp[0],temp[1],info[(int(r2.qname)*2)-2],info[(int(r2.qname)*2)-1])
    for p in predictions:
        if p!='Null':
            final.write(p+'\n')
    temp=[line]
else:
    temp=[line]
elif tag=='SP':
    for line in infile:
        if line[0]=='@':
            if line.split()[2]=='*':
                sprd=sam(line)
                if int(sprd.mapq)>>1:
                    prediction=predictLocus(line,info[(int(sprd.qname)-1)],'S')
                    if prediction!='Null':
                        final.write(prediction+'\n')
elif tag=='DS':
    for line in infile:
if line[0]!='@':
    if line.split()[2]!='*':
        dscr=sam(line)
        if int(dscr.mapq)>>1:
            prediction=predictLocus(line,info[int(dscr.qname)-1])['D']
            if prediction!='Null':
                final.write(prediction+'
')

def removeDupsSAM(path2sam,outpath):
    with open(path2sam[-3]+'rpt','w') as sam:
        with open(path2sam[-3]+'uniq','w') as uniq:
            subprocess.call(['uniq', '-w', '12', '-D', path2sam],stdout=sam)
            subprocess.call(['uniq', '-w', '12', '-u', path2sam],stdout=uniq)
    previous='nothing'
    with open(path2sam[-3]+'rpt','r') as enput:
        with open(path2sam[-3]+'norpt','w') as output:
            for line in enput:
                if line[:12]!=previous:
                    output.write(line)
                    previous=line[:12]
    with open(path2sam[-3]+'cat','w') as cat:
        with open(outpath,'w') as out:
subprocess.call(['cat', path2sam[:3]+'uniq', path2sam[:3]+'norpt'], stdout=cat)
subprocess.call(['sort','-nk1',path2sam[:3]+'cat'], stdout=out)
subprocess.call(['rm -f '+path2sam[:3]+'rpt', shell=True])
subprocess.call(['rm -f '+path2sam[:3]+'norpt', shell=True])
subprocess.call(['rm -f '+path2sam[:3]+'cat', shell=True])
subprocess.call(['rm -f '+path2sam[:3]+'uniq', shell=True])

def aluLine_process(line,s_or_l):
    aluSplit=line.split()
    tag,cigar=aluSplit[1],aluSplit[5]
    orient=orientConverter(tag)
    aluPos,mapping_score=aluSplit[3],aluSplit[4]
    tags=aluSplit[-1].split('_')
    for t in tags:
        if t[4:]=='MD:Z':
            tagMD=t
        if orient=='0':
            readPos=distanceM(cigar)
            cigar_span_list,cigar_character_list=splitCigar(cigar)
        elif orient=='1':
            cigar_span_list,cigar_character_list=splitCigar(cigar)
            readPos=distanceM(flipCigar(cigar_span_list,cigar_character_list))
    cigar_span_list,cigar_character_list=splitCigar(flipCigar(cigar_span_list,cigar_character_list))
    last_M_index,first_M_index=''.join(cigar_character_list).rfind('M'),cigar_character_list.index('M')
if last_M_index==first_M_index:

indexRange_MpBs=(str(readPos)+':'+str(int(readPos)+int(cigar_span_list[first_M_index])))

else:
    total,i=0,first_M_index
    while i<=last_M_index:
        total+=int(cigar_span_list[i])
        i+=1
    indexRange_MpBs=str(readPos)+':'+str(int(readPos)+total)

return(str(aluPos)+'_'+str(int(readPos)+1)+'_s_or_l_orient'+mapping_score+'_'+
cigar+'_'+indexRange_MpBs+'_'+tagMD)

def clippers(temp,seq):
    seq=seq.strip('
')
    temp_split=temp[0].split()[1].split('DEW')
    id_N,coZ=temp[0].split()[0],temp_split[0]
    if temp_split[2]=='N/A':
        info_split=temp_split[2].split('_')
    else:
        info_split=temp_split[1].split('_')

alu_pos,read_pos,orient,index_range,cigar=int(info_split[0]),
int(info_split[1]),info_split[3],info_split[6].split(':'),info_split[5]
mappedBases=int(index_range[1])-int(index_range[0])
tagMD=info_split[-1]
tag='N/A'

if alu_pos<=70:
    if orient=='0':
        if (read_pos-alu_pos)>=30:
            seq=seq[:((read_pos-alu_pos)+1)]
            tag='Head'
    else:
        seq=seq[:30]
else:
    seq='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN'
    elif orient=='1':
        if mappedBases<=70:
            seq=seq[mappedBases-1:]
            tag='Head'
        else:
            seq=seq[:30]
elif alu_pos>=180:
    if orient=='0':
        if int(index_range[1])<=70:
            seq=seq[int(index_range[1]):]
            tag='Tail0'
        if (alu_pos+mappedBases)>=285:
            tag='Tail1'
    else:
        seq='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN'
    elif orient=='1':
if int(index_range[0])>=30:
    seq=seq[:int(index_range[0])]
    tag='Tail0'

if (alu_pos+mappedBases)>=285:
    tag='Tail1'
else:

    seq='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
        NNNNNNNNNNNNNNNNNN'
    else:

    seq='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
        NNNNNNNNNNNNNNNNNN'

additionalInfo='\tCO:Z:aluMD=\'+tagMD+'\tCO:Z:orientation=\'+orient+'\tCO:Z:aluHT=\'+tag+'\tCO:Z:cigar=\'+cigar

return(id_N\\t'coZ'||\tCO:Z:aluPos=\'+str(alu_pos)+'\tCO:Z:readPos=\'+str(read_pos)+additionalInfo+'\n'+seq+'\n')

def reformatFQ_wAluData(path2fq,path2aluS,fwd_or_rv):
    with open(path2fq,'r') as alu_fq:
        alu_S=lister(path2aluS)

        with open('/home/vallmer/papio_pipe/30388/aligned/clippedAlu.'+fwd_or_rv+'*.fq','w') as outFastQ:
            temp,clipped,seq,qual=[],False,False,False

            for line in alu_fq:
                if (line[0]=='@') and (len(line.split()[0])==13):
                    readN=int(line.split()[0][1:])-1

                    if alu_S[readN].split()[5] != '*':
aluS_tag = aluLine_process(alu_S[readN], 'S')
clipped = True
else:
    aluS_tag = 'N/A'
if alu_L[readN].split()[5] != '*':
    aluL_tag = aluLine_process(alu_L[readN], 'L')
    clipped = True
else:
    aluL_tag = 'N/A'
if clipped == True:
    temp.append(line.split()[0] + '\t' + line.split()[1] + 'DEW' + aluS_tag + 'DEW' + aluL_tag)
    seq, clipped = True, False
else:
    outFastQ.write(line)
elif (seq == False) and (qual == False):
    outFastQ.write(line)
elif seq == True:
    clipString = clippers(temp, line)
    seqLen = len(clipString.split('\n')[-1])
    outFastQ.write(clippers(temp, line))
    seq, temp, = False, []
    qual = 'Plus'
elif qual == 'Plus':
    qual = True
    outFastQ.write(line)
elif qual == True:
    outFastQ.write(line[:seqLen] + '\n')
def toSplit_or_not2split(path2fwd, path2rvs):
    clipped_fwd, masked_fwd, unaltered_fwd = set(), set(), set()
    clipped_rvs, masked_rvs, unaltered_rvs = set(), set(), set()
    with open(path2fwd, 'r') as fqF:
        for line in fqF:
            if 'CO:Z:aluHT=' in line:
                if 'N/A' not in line:
                    clipped_fwd.add(int(line.split()[0][1:]))
                else:
                    masked_fwd.add(int(line.split()[0][1:]))
            elif 'CO:Z:F' in line:
                unaltered_fwd.add(int(line.split()[0][1:])),
    with open(path2rvs, 'r') as fqR:
        for line in fqR:
            if 'CO:Z:aluHT=' in line:
                if 'N/A' not in line:
                    clipped_rvs.add(int(line.split()[0][1:]))
                else:
                    masked_rvs.add(int(line.split()[0][1:]))
            elif 'CO:Z:R' in line:
                unaltered_rvs.add(int(line.split()[0][1:])),
    final = set()
    for readNum in clipped_fwd:
        if (readNum in unaltered_rvs):
            final.add(readNum)
    for readNum in clipped_rvs:
if (readNum in unaltered_fwd):
    final.add(readNum)
return(final)

def parseSplitFQ(path2fq,rnSet,path2outFQ):
    head,seq,plus=False,False,False
    with open(path2fq,'r') as inFq:
        with open(path2outFQ,'w') as outFq:
            for line in inFq:
                if head==True:
                    outFq.write(line)
                if seq==True:
                    if plus==True:
                        head,seq,plus=False,False,False
                    else:
                        plus=True
                else:
                    seq=True
    elif 'CO:Z:' in line:
        if int(line.split()[0][1:]) in rnSet:
            head=True
        outFq.write(line)

def sortPredict(path2predict,outfile):
    with open(outfile,'w') as out:
        sorting_piles,chrom_list=[],[]
        for i in range(1,21):
sorting_piles.append([])
chrom_list.append(str(i))

for p in lister(path2predict):
candidate = predict(p)
if candidate.chromosome in chrom_list:
    cols3plus = p.split()[1:]
    sorting_piles[int(candidate.chromosome) - 1].append(candidate.chromosome + '	' + candidate.locus + '	' + '	'.join(cols3plus))

for s in sorting_piles:
s_sort = (sortMe(s, 2))
for item in s_sort:
    out.write(item + '
')

def cleanup(label, orgID):
    if label != 'A':
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.DS.fq'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.DS.info'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.DS.sam'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.PE.sam'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.PE.fq'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.PE.info'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.SP.sam'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.SP.fq'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '.SP.info'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '_fwd.dup.sam'], shell=True)
        subprocess.call(['rm -f ' + orgID + '+label+' + label + '_rvs.dup.sam'], shell=True)
        subprocess.call(['rm -f ' + orgID + 'alu.P.' + label + '.mapped.sam'], shell=True)
subprocess.call(['rm -f ' + orgID + 'alu.P.' + label + '.sam'], shell=True)
subprocess.call(['rm -f ' + orgID + 'alu.S.' + label + '.mapped.sam'], shell=True)
subprocess.call(['rm -f ' + orgID + 'alu.S.' + label + '.sam'], shell=True)
subprocess.call(['rm -f ' + orgID + 'alu.S.' + label + '.predict'], shell=True)
else:
    subprocess.call(['rm -f ' + orgID + 'alu.P.polyA.sam'], shell=True)
    subprocess.call(['rm -f ' + orgID + 'alu.S.polyA.sam'], shell=True)
    subprocess.call(['rm -f ' + orgID + 'alu.S.polyA.fq'], shell=True)
    subprocess.call(['rm -f ' + orgID + 'alu.S.polyA.info'], shell=True)
    subprocess.call(['rm -f ' + orgID + 'alu.S.polyA.sam'], shell=True)

editFQ_title(path2_PE_fq, path2outdir + orgID + '.PE.fq')
editFQ_title(path2_SE_fq, path2outdir + orgID + '.SE.fq')
subprocess.call(['rm -f clipped_paired.fq'], shell=True)
subprocess.call(['rm -f clipped_single.fq'], shell=True)

bwaMeM(path2aluConsensus, path2outdir + orgID + '.PE.fq', 'P', 'S', path2outdir + orgID + 'alu.P.S.sam')
bwaMeM(path2aluConsensus, path2outdir + orgID + '.SE.fq', 'S', 'S', path2outdir + orgID + 'alu.S.S.sam')
mappOnly(path2outdir + orgID + 'alu.P.S.sam', path2outdir + orgID + 'alu.P.S.mapped.sam')
mappOnly(path2outdir + orgID + 'alu. S.S.sam', path2outdir + orgID + 'alu.S.S.mapped.sam')
processSam1((path2outdir + orgID + 'alu.P.S.sam'), (path2outdir + orgID + 'alu.S.S.sam'), (orgID + '.S'))
processSamA((path2outdir + orgID + 'alu.P.S.mapped.sam'), (path2outdir + orgID + 'alu.S.S.mapped sam'), (orgID + '.S'))
bowtie2(path2referenceGenome, (orgID + '.S.PE.fq'), 'P', (orgID + '.S.PE.sam'))
bowtie2(path2referenceGenome,(orgID+'.S.DS.fq'),'S',(orgID+'.S.DS.sam'))
bowtie2(path2referenceGenome,(orgID+'.S.SP.fq'),'S',(orgID+'.S.SP.sam'))
predictInsertionOG(orgID+'.S',orgID+'.S.predict')
sortPredict(orgID+'.S.predict',orgID+'.S.predict.sort')
cleanup('S',orgID)

bwaMeM(path2aluConsensus,path2outdir+orgID+'.PE.fq','P','L',path2outdir+orgID+'.alu.P.L.sam')
bwaMeM(path2aluConsensus,path2outdir+orgID+'.SE.fq','S','L',path2outdir+orgID+'.alu.S.L.sam')
mappOnly(path2outdir+orgID+'.alu.P.L.sam',path2outdir+orgID+'.alu.P.L.mapped.sam')
mappOnly(path2outdir+orgID+'.alu.S.L.sam',path2outdir+orgID+'.alu.S.L.mapped.sam')
processSam1((path2outdir+orgID+'.alu.P.L.sam'),(path2outdir+orgID+'.alu.S.L.sam'),(orgID+'.L'))
processSamA((path2outdir+orgID+'.alu.P.L.mapped.sam'),(path2outdir+orgID+'.alu.S.L.mapped.sam'),(orgID+'.L'))
bowtie2(path2referenceGenome,(orgID+'.L.PE.fq'),'P','L',(orgID+'.L.PE.sam'))
bowtie2(path2referenceGenome,(orgID+'.L.DS.fq'),'S','L',(orgID+'.L.DS.sam'))
bowtie2(path2referenceGenome,(orgID+'.L.SP.fq'),'S',(orgID+'.L.SP.sam'))
predictInsertionOG(orgID+'.L',orgID+'.L.predict')
sortPredict(orgID+'.L.predict',orgID+'.L.predict.sort')
cleanup('L',orgID)

bwaMeM(path2polyA,path2outdir+orgID+'.PE.fq','P','L',(orgID+'.P.polyA.sam'))
bwaMeM(path2polyA,path2outdir+orgID+'.SE.fq','S','L',(orgID+'.S.polyA.sam'))
processSam_polyA((orgID+'.P.polyA.sam'),(orgID+'.S.polyA.sam'),(orgID+'.polyA.fq'),(orgID+'.polyA.info'))
bowtie2(path2referenceGenome,(orgID+'.polyA.fq'),'S',(orgID+'.polyA.sam'))

predictPolyA((orgID+'.polyA.sam'),(orgID+'.polyA.predict'),(orgID+'.polyA.info'))

cleanup('A',orgID)
Table A.6. PCR genotype data for the 28 loci that were polymorphic in *Papio* and *Theropithecus*

<table>
<thead>
<tr>
<th>Species</th>
<th>Common name</th>
<th>Origin</th>
<th>Sample ID</th>
<th>17</th>
<th>17</th>
<th>21</th>
<th>21</th>
<th>25</th>
<th>25</th>
<th>23</th>
<th>23</th>
<th>24</th>
<th>24</th>
<th>32</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Homunculus</em></td>
<td></td>
<td></td>
<td>Hel.a CCL-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Reference Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>27861</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>T.142</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28697</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>28557</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>30388</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio kimbang</em></td>
<td>Kinds Baboon</td>
<td>Jeff Rodgers</td>
<td>34474</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><em>Papio kimbang</em></td>
<td>Kinds Baboon</td>
<td>Jeff Rodgers</td>
<td>34472</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16066</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16098</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio hamadryas</em></td>
<td>Ethiopian wild hamadryas</td>
<td>Jeff Rodgers</td>
<td>97124</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Theropithecus gelada</em></td>
<td>Gelada Baboon</td>
<td>T. Dicottel</td>
<td>TD071167</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Theropithecus gelada</em></td>
<td>Gelada Baboon</td>
<td>SDFZ</td>
<td>KB10538</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Species</th>
<th>Common name</th>
<th>Origin</th>
<th>Sample ID</th>
<th>34</th>
<th>34</th>
<th>35</th>
<th>35</th>
<th>35</th>
<th>35</th>
<th>36</th>
<th>36</th>
<th>41</th>
<th>41</th>
<th>44</th>
<th>44</th>
<th>44</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Homunculus</em></td>
<td></td>
<td></td>
<td>Hel.a CCL-2</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Reference Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>27861</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>T.142</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28697</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>28557</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>30388</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><em>Papio kimbang</em></td>
<td>Kinds Baboon</td>
<td>Jeff Rodgers</td>
<td>34474</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio kimbang</em></td>
<td>Kinds Baboon</td>
<td>Jeff Rodgers</td>
<td>34472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16066</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16098</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><em>Papio hamadryas</em></td>
<td>Ethiopian wild hamadryas</td>
<td>Jeff Rodgers</td>
<td>97124</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Theropithecus gelada</em></td>
<td>Gelada Baboon</td>
<td>T. Dicottel</td>
<td>TD071167</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Theropithecus gelada</em></td>
<td>Gelada Baboon</td>
<td>SDFZ</td>
<td>KB10538</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Species</td>
<td>Common name</td>
<td>Origin</td>
<td>Sample ID</td>
<td>48</td>
<td>48</td>
<td>57</td>
<td>57</td>
<td>58</td>
<td>58</td>
<td>64</td>
<td>64</td>
<td>66</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>-----------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Homo sapiens</em></td>
<td>Human</td>
<td>ATCC</td>
<td>HeLa CCL-2</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Reference Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>27861</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>L142</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>L354</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28697</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28755</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>28547</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>30338</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Kinda Baboon</td>
<td>Jeff Rodgers</td>
<td>34474</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Kinda Baboon</td>
<td>Jeff Rodgers</td>
<td>34472</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16046</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16098</td>
<td>1</td>
<td>1</td>
<td>-9</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Papio hamadryas</em></td>
<td>Ethiopian-wild hamadryas</td>
<td>Jeff Rodgers</td>
<td>97124</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theropithecus gelada</td>
<td>Gelada Baboon</td>
<td>T. Dissetti</td>
<td>TD871187</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theropithecus gelada</td>
<td>Gelada Baboon</td>
<td>SDPZ</td>
<td>KB10538</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Species</th>
<th>Common name</th>
<th>Origin</th>
<th>Sample ID</th>
<th>75</th>
<th>73</th>
<th>72</th>
<th>72</th>
<th>84</th>
<th>84</th>
<th>85</th>
<th>85</th>
<th>86</th>
<th>86</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Homo sapiens</em></td>
<td>Human</td>
<td>ATCC</td>
<td>HeLa CCL-2</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Reference Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>27861</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>L142</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>L115</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28697</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28755</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>28547</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>30338</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Kinda Baboon</td>
<td>Jeff Rodgers</td>
<td>34474</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Kinda Baboon</td>
<td>Jeff Rodgers</td>
<td>34472</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16046</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16098</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio hamadryas</em></td>
<td>Ethiopian-wild hamadryas</td>
<td>Jeff Rodgers</td>
<td>97124</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Theropithecus gelada</td>
<td>Gelada Baboon</td>
<td>T. Dissetti</td>
<td>TD871187</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Theropithecus gelada</td>
<td>Gelada Baboon</td>
<td>SDPZ</td>
<td>KB10538</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Species</td>
<td>Common name</td>
<td>Origin</td>
<td>Sample ID</td>
<td>87</td>
<td>87</td>
<td>90</td>
<td>90</td>
<td>97</td>
<td>97</td>
<td>98</td>
<td>98</td>
<td>108</td>
<td>108</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td><em>Homo sapiens</em></td>
<td>Human</td>
<td>ATCC</td>
<td>HeLa CCL-2</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Reference Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>27861</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>L142</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio anubis</em></td>
<td>Olive Baboon</td>
<td>Jeff Rodgers</td>
<td>LIV5</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28697</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><em>Papio ursinus</em></td>
<td>Chacma baboon</td>
<td>Jeff Rodgers</td>
<td>28755</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>28547</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio papio</em></td>
<td>Guinea baboon</td>
<td>Jeff Rodgers</td>
<td>30388</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio kirkii</em></td>
<td>Kinds Baboon</td>
<td>Jeff Rodgers</td>
<td>34474</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio kirkii</em></td>
<td>Kinds Baboon</td>
<td>Jeff Rodgers</td>
<td>34472</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16066</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>Papio cynocephalus</em></td>
<td>Yellow Baboon</td>
<td>Jeff Rodgers</td>
<td>16098</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td><em>Papio hamadryas</em></td>
<td>Ethiopian-wld hamadryas</td>
<td>Jeff Rodgers</td>
<td>97124</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Theropithecus gelada</td>
<td>Gelada Baboon</td>
<td>T. Disotell</td>
<td>TED167</td>
<td>0</td>
<td>0</td>
<td>-9</td>
<td>-9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Theropithecus gelada</td>
<td>Gelada Baboon</td>
<td>S0F2</td>
<td>KB10538</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A.6 cont.
Figure A.1. *Papio* basal divergence models. Comprehensive representation of possible basal divergence model reconstructions generated using all six extant *Papio* species. A maximum of 31 rooted monophyletic models can be generated from such a genus comprised of six species. These models can be further organized into three distinct groups based on the number of species contained in the subsequent clades. Group I depicts the six different scenarios when one of the six species diverges prior to the other five. Group I-A) illustrates *P. kindae* diverging first, followed by B) *P. ursinus* first, then C-F) *P. cynocephalus, P. papio, P. hamadryas,* and *P. anubis* diverging first, respectively. Group II depicts the 15 different models when two of the six species diverge prior to the other four. All possible combinations of this scenario are illustrated in Group II A-O. Group III depicts the ten different models generated from a basal divergence that forms two clades each comprised of three species. All ten combinations are listed in Group III A-J.
Figure A.1. *Papio* basal divergence models cont.
APPENDIX B: LETTERS OF REQUEST AND PERMISSION

from: Vallmer Jordan <jordanvallmer@gmail.com>
to: info@biomedcentral.com
date: Thu, Apr 5, 2018 at 10:56 AM
subject: Request for Written Permission to Publish Mobile DNA Article in Dissertation

Dear Mobile DNA Editorial Staff,

I am contacting you to request written permission to publish my first author paper, "A computational reconstruction of Papio phylogeny using Alu insertion polymorphisms" in my Ph.D. dissertation (https://doi.org/10.1186/s13100-018-0118-3).

Best,

Vallmer Edward Jordan II
Ph.D. Candidate
Bridges to the Doctorate Fellow
Batzer Lab (Laboratory of Comparative Genomics)
Department of Biological Sciences
A653 Life Sciences Building
Louisiana State University
Baton Rouge, LA, 70803 USA

from: "Neil Castil" <neil.castil@springernature.com> <neil.castil@springernature.com> via rbmffo6vu6ky8858.0elaaj3klvrw6e96x.q2gxauw.2-cuteam.eu9.bnc.salesforce.com
to: "jordanvallmer@gmail.com" <jordanvallmer@gmail.com>
date: Thu, Apr 5, 2018 at 11:29 AM
subject: 00845313 RE: Request for Written Permission to Publish Mobile DNA Article in...
Dear Vallmer,

Thank you for contacting Springer Nature.

The open access articles published in BioMed Central’s journals are made available under the Creative Commons Attribution (CC-BY) license, which means they are accessible online without any restrictions and can be re-used in any way, subject only to proper attribution (which, in an academic context, usually means citation).

The re-use rights enshrined in our license agreement (http://www.biomedcentral.com/about/policies/license-agreement) include the right for anyone to produce printed copies themselves, without formal permission or payment of permission fees. As a courtesy, however, anyone wishing to reproduce large quantities of an open access article (250+) should inform the copyright holder and we suggest a contribution in support of open access publication (see suggested contributions at http://www.biomedcentral.com/about/policies/reprints-and-permissions/suggested-contributions).

Please note that the following journals have published a small number of articles that, while freely accessible, are not open access as outlined above: Alzheimer’s Research & Therapy, Arthritis Research & Therapy, Breast Cancer Research, Critical Care, Genome Biology, Genome Medicine, Stem Cell Research & Therapy. You will be able to find details about these articles at http://www.biomedcentral.com/about/policies/reprints-and-permissions.

If you have any questions, please do not hesitate to contact me.

With kind regards,

---

Neil Castil
Global Open Research Support Executive
Global Open Research Support

Springer Nature
T +44 (0)203 192 2009
www.springernature.com

---

Springer Nature is a leading research, educational and professional publisher, providing quality content to our communities through a range of innovative platforms, products and services.
Every day, around the globe, our imprints, books, journals and resources reach millions of people – helping researchers, students, teachers & professionals to discover, learn and achieve.

---

In the US: Springer Customer Service Center LLC, 233 Spring Street, New York, NY 10013
Registered Address: 2711 Centerville Road Wilmington, DE 19808 USA
State of Incorporation: Delaware, Reg. No. 4538065
Rest of World: Springer Customer Service Center GmbH, Tiergartenstraße 15 – 17, 69121 Heidelberg
Registered Office: Heidelberg | Amtsgericht Mannheim, HRB 336546
Managing Directors: Martin Mos, Dr. Ulrich Vest
Vita

Vallmer Edward Jordan, II is the son of Vallmer Wayman Jordan and Vanessa Hinton Jordan. He was born in Chicago, IL in 1990. Vallmer graduated from Morehouse College with a BA in Psychology in 2013. In the summer of 2013, Vallmer worked as an undergraduate research associate in the laboratory of Dr. Mark Batzer. At the end of the summer, Vallmer began his doctoral studies in the Department of Biological Sciences at Louisiana State University in Baton Rouge, Louisiana. He has remained under the tutelage of Dr. Mark Batzer throughout his entire matriculation at Louisiana State University. Vallmer will graduate with the degree of Doctor of Philosophy in August, 2018.