2023

Population Differences in Human Mandibular Growth

Madison Hubbart
University of Central Florida

POPULATION DIFFERENCES IN HUMAN MANDIBULAR GROWTH

By

MADISON HUBBART
B.S. Florida State University, 2021

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Arts
in the Department of Anthropology
in the College of Sciences
at the University of Central Florida
Orlando, Florida

Spring Term
2023

ABSTRACT

Mandibles are one of the most common bones encountered in the human archaeological record. Variation in mandibular morphology is often associated with differences in subsistence strategy as masticatory stresses influence bone growth and development. Bone growth is stimulated by bone modeling, the process by which formation and resorption occur through the uncoupled activities of osteoblasts and osteoclasts, respectively. There is a limited understanding of bone modeling patterns in humans due to a lack of quantitative data and small sample sizes. The aim of this research was to address the question: is there a shared bone modeling pattern in the mandible of *Homo sapiens*? To address this question, this research analyzed bone modeling patterns during ontogeny in a sample of 48 mandibles from three geographically diverse populations with differing subsistence strategies: Western Europe (France and Germany), Greenland (Inuit), and South Africa (Khoe khoe and San). The sample was divided into four age categories. Epoxy replicas of the bone were analyzed under a digital microscope, and bone resorption was identified and quantified to create digital bone modeling maps. This study found subtle population differences throughout ontogeny, with bone modeling patterns that diverge during adulthood, possibly related to subsistence strategy. This study contributes to research on bone modeling patterns in the craniofacial system of *H. sapiens*, expanding on our understanding of bone growth dynamics and morphological adaptations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION & PURPOSE

Many of the key features that define us as *Homo sapiens* and differentiate us from our fossil ancestors are found on the face (Stringer, 2016). Biological anthropologists often rely on facial morphology to assess human ancestry and estimate biological sex and ontogenetic age. Variability in facial morphology is likely the result of many concurrent factors that influence bone development, including genetic, diet, environmental pressures (i.e. climate), and cultural practices in past and present day human populations (e.g. Molnar 2008; Hubbe et al., 2009; Evteev et al., 2017; Wroe et al., 2018; Stansfield et al., 2021; Butaric et al., 2022), and the mandible, being the last bone in the face to finish growing, it is especially susceptible to such factors (Kurihara et al. 1980, Polanski, 2011). As adult morphology is the final product of development, identifying population-level differences in ontogeny are important for interpreting and understanding the factors contributing to adult variation. However, the developmental processes underlying facial morphology are not fully understood. By investigating patterns of development in present day humans, we can disentangle factors attributed to facial variability and help clarify which features are developmentally homologous, as well as ancestral (or derived), when interpreting the human fossil record. Furthermore, understanding the ontogenetic process is essential for sexing, aging, and assessing ancestry from subadult skeletal remains, which is often challenging (Viðarsdóttir, O'Higgins, & Stringer, 2002; Wilson & Humphrey, 2017).

One approach to understanding bone development is histology. Histology is the study of tissue at the microscopic level (White et al. 2012). Bone histology commonly involves transverse cross sections, which is damaging to the bone, and for that reason, rarely used in fossil and archaeological studies. Alternatively, bone microstructure can be studied using surface histology

(Bromage, 1984; Bromage 1985). Developed from the common, destructive methods of histology, surface histology analyzes the external, or periosteal, surface of bone using microscopy without damaging the bone tissue. Both within and outside biological anthropology, surface histology is scarcely used in favor of more invasive procedures, such as the slicing of bone.

Surface histology has been used to identify patterns associated with bone modeling (Table 1). Bone modeling is the dynamic process by which bone is formed (DiGangi & Moore, 2013); this process removes or builds bone to ensure that bone structure is maintained and functional. Bone formation through osteoblastic activity and bone resorption by osteoclast activity occurs asynchronously and at different rates (DiGangi & Moore, 2013). Patterns of location in bone formation and resorption can aid researchers in analyzing growth directionality and provide direct evidence of the development of morphological features that are often used for ancestry, sex, and age estimation. Additionally, studies have also found bone modeling patterns to be species-specific (Enlow, 1962; Bromage, 1989; Martinez-Maza et al., 2006). However, many surface histological studies thus far lack the quantitative data and sample sizes necessary to make robust claims about bone modeling patterns for both populations and species levels (Lacruz et al., 2015). Consequently, our understanding of the variability in bone modeling patterns in past and present-day populations is limited.

A series of studies by Schuh et al. (2019, 2020, 2021), addressed methodological shortcomings through the use of advanced three-dimensional imaging techniques to develop a novel approach for quantifying and visualizing bone modeling patterns on the maxilla, a bone that contains important anatomical information for the reconstruction of phylogenetic relationships in hominins. They merged geometric morphometric methods with surface

microstructure analyses to establish a comparative frame of bone growth dynamics at the micro- and macroscopic levels. Applying these methods to large sample sizes, the results of these studies suggest maxillary growth in humans is highly constrained from early stages in ontogeny, and that morphological changes are driven by changes in osteoblastic and osteoclastic rates of expression, rather than differences in bone modeling patterns (i.e., changes in location of formation and resorption). This study is a follow-up to Schuh and co-authors, but applied to the mandible, using the same populations and methodology. By studying bone modeling patterns in the mandible, in addition to integrating larger sample sizes and quantitative data, we can better understand the pressures affecting the ontogenetic trajectory, growth, and development of the mandible as it relates to the craniofacial system as well as potentially aiding juvenile age, sex, and ancestry estimation (Viðarsdóttir, O'Higgins, & Stringer, 2002).

Purpose

This research answers the question: is there a shared bone modeling pattern in the mandible of *H. sapiens*? To answer this question, this research evaluates the ontogenetic bone modeling patterns during ontogeny in the modern human mandible from three geographically diverse and morphologically distinct populations: Western European, Greenland Inuit, and South African. These populations are known for their differences in facial morphology (Freidline et al., 2015). The Greenlandic Inuit are characterized by their large faces with slight prognathism (Freidline et al., 2015; Hylander, 1977), while the Western Europeans and South Africans differ from one another in subtle aspects of mandibular ramal shape and overall facial size and shape (Cakirer et al., 2001; Dennison & Healey, 2007; Moisik & Dediu, 2017; von Cramon-Taubadel, 2011).

Following Schuh's et al.'s (2020) study on the maxilla using the same sample and methodology, it was expected that discrepancies in location and expression of bone modeling patterns may appear between these populations in the areas of greatest shape differences. This research identifies and compares bone modeling patterns of the mandible. More specifically, this research evaluates whether there are differences in bone modeling patterns 1) throughout ontogeny, 2) between populations, and 3) between biological sexes. Potential divergence in bone modeling patterns can be indicative of morphological shape differences observed between the three geographic groups. Furthermore, if the mandible displays different bone modeling patterns across populations, it becomes necessary to understand why these patterns may differ. Thus, an understanding of weaning, diet, and subsistence practices was reviewed to see if variation can be seen within groups as well as between groups. As such, groups may have additional dental pressures due to cultural behaviors, such as the use of the teeth as tools. As a result of these behaviors, there may also be differences in osteoblastic and osteoclastic activity due to functional adaptation, which may indicate differences between males and females for a variety of gendered community roles.

Another objective of the research is to develop an error test. The development of an error test is significant to this research as the interpretation of bone modeling patterns, or presence or absence of bone resorption and formation, is subjective to the observer. An interobserver and intraobserver error test was conducted throughout the research process to identify potential discrepancies in the identification of bone resorption and formation. Of the studies that have used surface histological methods, this will be the first project to use error tests.

CHAPTER 2 BACKGROUND

Bone Biology and Terminology

Bone's primary function is to provide structure and support, while also operating as the primary metabolic reserve of calcium in the human body (Hadjidakis & Androulakis, 2006). Bone is composed of cells, vessels, and hydroxyapatite, as well as extracellular matrix and collagen. There are two main types of bone: trabecular and cortical. Cortical bone or compact bone is dense, highly calcified, and is resistant to bending or twisting. Conversely, trabecular bone is more elastic and less dense, healing quickly with a high turnover rate and larger metabolic function. A higher turnover rate, in osteology, refers to the rate to which bone is resorbed and formed. Bone formation, or deposition, is characterized by the appearance of collagen fibers bundles while bone resorption is observed through the identification of Howship's lacunae. Bone formation and resorption have two phases: active and resting (Martinez-Maza et al. 2013). In the active phase, collagen fiber bundles delineate and retain a preferred orientation while Howship's lacunae are of varying size and shape, defined by more precise boundaries. In the resting phase, osteoblasts are flattened, and collagen fiber bundles are masked, leaving behind a smooth surface and mineralized bone. Howship's lacunae, in the resting phase, are shallower with poorer defined concavities.

To understand bone formation and resorption, it is imperative to acknowledge the cells that carry out such processes. Osteoblasts, involved in bone formation, have the responsibility of bone matrix component production. Osteoblasts line the matrix and coming from mesenchymal stem cells, get caught in the new bone matrix to become an osteocyte or become flat lining cells along the bone surface. Their functional activity depends on the age of the cells. Osteoclasts are the bone lining cells that are responsible for bone resorption. Unlike osteoblasts, osteoclasts are

multinucleated and come from hematopoietic cells (DiGangi & Moore, 2013). Osteoclasts resorb bone by acidification and the breakdown of proteins into amino acids. From here, hydroxyapatite crystals, followed by collagen fibers, are digested (Hadjidakis and Androulakis, 2006). Bone remodeling is the result of both osteoblastic and osteoclastic activity as the process serves to repair microfractures. This is performed by the successive process of resorption, reversal, and formation (Hadjidakis &Androulakis, 2006).

The terms bone modeling and bone remodeling are often used interchangeably, alluding to difficulties in literature that must be clarified. Martinez-Maza et al. (2006), has suggested that, though there are differences in terminology, our understanding of modeling and remodeling according to Frost is synonymous to growth remodeling and haversian remodeling, respectively, per Enlow. In order to interpret the results of this project, patterns in the mandible are the result of bone modeling, rather than bone remodeling. According to Barak (2020), bone remodeling is a successive process where bone resorption is always followed by bone formation. Osteoclasts resorb bone and leave an eroded surface behind. Formation begins on this eroded surface, directly replacing old with new bone. In these situations, bone is deposited by osteoblasts in layers of lamellae. In bone remodeling, deposition can take place on bone surfaces or be seen in haversian systems as bone tissue creates cutting cones filled by concentric lamellae. Bone modeling has contrasting conditions. Osteoclast and osteoblast activity is uncoupled and acts primarily independently (Frost, 1990; Martinez-Maza et al, 2006). However, it is possible for resorption and formation to occur simultaneously but always at separate locations on the bone. Bone modeling is also always on the surface of bone, consisting of lamellar, woven, or parallel fibers of bone. The amount of bone being resorbed or deposited is more than in remodeling. It is important to note that functional adaptation is due to modeling and not remodeling, as cellular

processes are occurring independently (Barak, 2020; DiGangi & Moore, 2013). Studies have shown that bone turnover through bone modeling begins during the first trimester of gestation, when the skeleton begins to develop (Allen & Burr, 2019; Radlanski & Klarkowski, 2001).

Mandibular Anatomy

The masticatory system is comprised of the maxilla, temporal, and mandible (White et al., 2012). The integration and development of the maxilla and temporal bones allow for the final location of the mandible and associated musculature. Because the mandible is the last part of the face to finish growing, it is more susceptible to external factors such as diet and mastication (Stansfield, Evteev, & O'Higgins, 2018). There are four main muscles used in mastication that may affect the rugosity of muscle attachment sites on the mandible: temporalis, medial pterygoid, lateral pterygoid, and masseter (Figure 1). The temporalis muscle originates at the frontal, parietal, and temporal bones, inserting at the coronoid process and anterior border of the mandibular ramus. The masseter muscle begins at the zygomatic arch and inserts at the mandibular ramus and angle, as well as the lateral surface of the coronoid process. The origin of the pterygoid medialis muscle is at the lateral pterygoid plate, palatine, and maxilla, while the insertion of this muscle is at the medial surface of the mandibular surface of the ramus and gonial angle. Finally, the pterygoid lateralis muscles begins at the lateral pterygoid plate and the lateral aspect of the greater wing of the sphenoid. This muscle inserts at the mandibular condyle and the temporomandibular joint (White et al., 2012). The integration of the muscles in the masticatory system amplifies the importance of the impact of the growth and development of the craniofacial system as muscles are attached to multiple bones in the skull. The pattern of integration in the craniofacial system, however, is dynamic, constantly growing and displacing to accommodate dentition and muscle attachment sites.

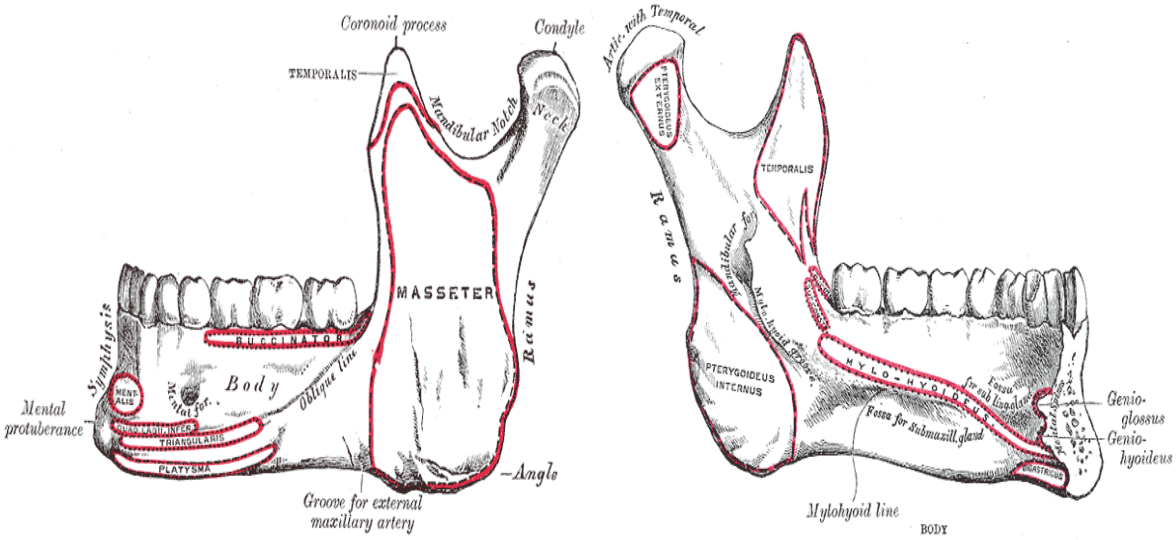


Figure 1 Mandible with marked muscle attachments. Left: lateral aspect of the mandible showing attachment site of the masseter. Right: medial aspect of the mandible showing attachment sites of the temporalis, pterygoid medialis, and pterygoid lateralis (Gray, 1918).

Mandibular Growth

Craniofacial growth occurs due to two types of movements during ontogeny: cortical drift and bone displacement (Enlow & Moyers, 1971). Cortical drift is the result of bone resorption and formation on bone surfaces, or bone modeling. As bone modeling progresses, bone begins to enlarge and increase in its anterior-posterior dimensions. The second movement involves the displacement of bone as growth moves bones away from each other, repositioning them. As growth occurs in a posterior direction, it also displaces anterior to accommodate proportional size and shape differences. The craniofacial system is simultaneously growing and displacing in order to comfortably fit bones together. This concept became known as the "Principle of the V" (Enlow & Bang, 1965). This principle can be seen in the maxillary arch and mandible where growth is occurring at the wide end of the "V" shaped bone, and the growth and location of the mandible is largely the result of the movement and growth of the maxilla. The

maxilla has been shown to operate on the basis of Wolff's law which claims that bone will adapt to pressure (Frost, 1990).

Enlow's (1966) classic work on the face used histology to analyze differences in bone modeling patterns between 11 rhesus monkeys. For both *H. sapiens* and the monkeys, a general pattern of facial growth could be seen. However, differences appeared in bone modeling patterns. The maxillary arch in the rhesus monkey is depository, growing in a forward and downward direction. The maxillary arch in the modern human showed resorption, with a downward growth movement related to orthognathism.

Recently, Schuh and co-authors' (2021) compared a large sample of chimpanzees and modern humans to examine bone modeling variation in the maxilla. This work corresponded with Enlow's (1966) and other findings (Bromage, 1989; McCollum, 2008; Martinez-Maza et al., 2010, Martinez-Maza et al., 2015) where orthognathism for the modern humans was found to be related to high amounts of bone resorption in the maxilla during development (Schuh et al., 2021). Chimpanzees, on the other hand, characterized by maxillary prognathism, expressed lower amounts of bone resorption the periosteal surface of the maxilla throughout ontogeny. By studying patterns of bone modeling variation in the mandible, we can make comparisons to other primates and the implications for the orthognathic human face.

Macroscopic Studies on Mandibular Growth

Mandibular growth during infancy was studied by Liu et al. (2010). This work was completed through the use of cephalograms to describe the growth, maturation and remodeling of the mandible for both infancy and early childhood. Remodeling, in this study, is not defined, with much of the focus of the study being on mandibular morphological changes. According to their work, mandibular growth was seen to be most rapid at the early stages of life compared to

later age groups, namely due to changes in the spatial relationship for the mandibular corpus and ascending ramus. The results of the study found significant differences between males and females in mandibular size and remodeling changes. Males had greater growth than females, even though there were similar patterns of change between males and females in the first postnatal year. Shape differences, particularly of the mandibular ramus, have also be seen for populations throughout ontogeny (Terhune et al., 2018). In a study comparing *H. sapiens* and Neanderthals, there were findings of mandibular ramal shape differences between modern human populations, as well as between species. This work suggested differences in ontogenetic trajectory as Neanderthals experienced shape changes much sooner than modern humans (Terhune et al., 2018).

Studies have found individual variations in the growth of the mandible for body length, ramus height, and mandibular length (Işeri & Solow, 2000; Gomes & Lima, 2006; Terhune et al., 2018). Mandibular growth for males and females is not significantly different until puberty (Gomes & Lima, 2006). For the authors, at puberty, mandibular growth rates increase, but these rates are variable and due to individual variation, rather than sex or age-related differences. While growth rate increases, dimensions of the mandibular body also increase (Işeri & Solow, 2000). Mechanisms for increased dimensions are likely due to the "V" principle and the mandibular dimensions adjusting based on the size of the cranium (Rosas & Martinez-Maza, 2010). Beyond this, Rosas et al. (2019) offers a more structuralist approach based on Enlow, where growth and compensatory mechanisms contribute to variability in facial pattern and mandibular shape. The mandibular symphysis ossifies soon after birth and an increase in mandibular body dimensions may allow for the mandibular body to bend with increased masticatory stresses (Işeri & Solow, 2000).

Coquerelle et al.'s (2011) study sought to investigate questions on the sexual dimorphism of the human mandible by analyzing CT scans of 159 modern humans. From birth to age four, sex differences can be seen. However, these differences decrease from age four until puberty. At the onset of puberty, males and females undergo allometric shape changes. Sexual dimorphism was found at the mandibular ramus and mental region as male mandibles had greater ramus breadth, and overall longer mandibles in the antero-posterior direction.

The human chin has been one of the most studied features of the mandible as it is one of the defining characteristics of modern humans. The development of the chin has been attributed to several factors, the first of which is biomechanical stresses. Though initially thought to be linked with mandibular biomechanics and architecture, Ichim et al. (2006) suggests the development of the chin is not related to functional stress placed on the mandible. By studying bite force and strain in the symphyseal region for both chinned and flat mandibles, the authors calculated the strain values for each type of mandible and found each to be in normal range. Other studies have argued for the development of the chin as the result of growth and spatial restraints (Holton et al., 2015). Symphyseal shape changes during ontogeny were explored as the result of symphyseal bending resistance. Results from this study displayed associations between the chin and increased vertical bending ability. This study also confirmed complexities in the interactions between the mandibular symphysis and jaw function during development. In a study using soft channel constructed CT scans, Coquerelle et al. (2013a) was able to map out how musculature changed as dentition developed. The authors suggested deciduous and permanent anterior tooth reorientation can influence bone modeling and is highly correlated with the appearance of the mental prominence.

Microscopic Studies on Mandibular Growth

Through the use of destructive histological methods, Hans et al. (1995) sought to identify patterns of bone resorption and formation in the *H. sapiens* mandibular ramus. The authors studied 30 juvenile human mandibles of unknown geographic origin for mandibular growth. This work did not show a consistent pattern in bone modeling during ontogeny but rather, three distinct patterns unrelated to age.

Kurihara et al. (1980) also conducted a study on 36 human skulls to determine the timing of resorption for both the anterior maxilla and mandible during ontogeny. The individuals were classified based on dental eruption. For both the first and second stages of dental eruption, the mandible did not display surface resorption. During the third stage, at the time after the eruption of the lower second molar, resorption was observed on mandible in the alveolar region and extended from the mandibular symphysis to the second deciduous molar. The fourth stage, categorized based on mixed dentition, displayed resorption more irregularly extending distally to the premolar region and in some cases, inferior towards the mental foramen. The authors found resorption of the maxilla to occur sooner than the mandible, possibly being attributed to earlier maxillary dental eruption and posteroanterior arch length increase. However, it was surmised that this was because the maxilla grows faster in conjunction with other cranial bones, and the mandible reacts to the growth and displacement of the maxilla.

Only one study has used surface histology to understand the bone modeling patterns of the mandible in *H. sapiens.* Martinez-Maza et al. (2013) sought to understand skull development through Enlow's principles by using microscopic analysis of replicas from the human facial skeleton and mandible to identify areas of deposition and resorption. Their sample consisted of 12 human skulls of known age and sex from a Western European population. The results were consistent with Enlow and Hans' (1996) work on the face. The mandible showed more

complexity in the posterior region of the corpus and ramus but was largely depository in the symphyseal region and anterior corpus.

Bone Modeling Studies on Fossil Hominins

Bone modeling studies have not only been performed on fossil species (See Table 1). Bromage (1989) used surface histology to investigate bone modeling patterns on *Australopithecus*, *Paranthropus*, and early *Homo*. He found bone deposition to characterize most of the face for all *Australopithecus* and the early *Homo* species in his sample, suggesting similarities to chimpanzees and an ancestral growth pattern. McCollum's (1999) work on the Australopith face supported Bromage (1989). The Australopith face is displayed in a forward and downward growth pattern (McCollum, 1999). Together, their results differed from previous findings on *H. sapiens* by Enlow & Moyers (1971) who found bone deposition could be seen on the posterior aspects of the face with resorption on the anterior face, leading to questions on species-specific bone modeling patterns as well as explaining differences in facial projection between *H. sapiens* and early fossil hominins.

A study on the bone modeling on the juvenile *H. erectus* (KNM-WT 15000) and *H. antecessor* (ANT6-69), fossils of similar age, was conducted by Lacruz et al. (2013). It was hypothesized that their differences of facial prognathism would reflect differences in bone growth patterns. According to the authors, *H. antecessor* contrasted *H. erectus* who displayed bone deposition in the area of the subnasal region, as opposed to resorption. This led the authors to conclude that KNM-WT 15000 was more similar to *Australopithecus* and ANT6-69 similar to *H. sapiens*, suggesting that the bone modeling pattern of *H. antecessor* is derived, rather than ancestral. However, these bone modeling patterns cannot be considered typical for their species as there are so few specimens available to study, and the polarity (i.e., ancestral versus derived)

of bone modeling patterns continues to be unknown. Generalized claims cannot be made for an entire species based on the bone modeling patterns of a few individuals. Lacruz et al. (2015), analyzed bone modeling growth patterns of the maxilla in Neanderthals and claimed that the Neanderthal fossils differ greatly from the bone modeling patterns seen in *H. sapiens*. However, while their sample size is better than the previous study (n=8), there is no quantification of the data.

A study on *H. heidelbergensis* mandibles from Sima de los Huesos, found the mandible displayed a unique growth pattern due to its distinct morphology (Rosas and Martinez-Maza, 2010). The authors created a color-coded map to represent mandibular bone modeling patterns. This work found two patterns of growth. The first was unique to *H. heidelbergensis* and the second more similar to *H. sapiens*. The *H. heidelbergensis* individuals displayed resorption on the lingual side of the mandibular corpus and showed the species' facial growth to be due to forward growth, with an inward displacement of the ramus, corresponding with the "Principle of the V" (Enlow & Bang, 1965). Very little is known about bone modeling patterns of species in the fossil record as sample sizes are historically small. There is no gauge of variability, and it is nearly impossible to discern if bone modeling patterns retain any commonalities or differences based on these findings.

Table 1 Publications using surface histology.

Publication	Authors	Year	Species Studied	Bone
Ontogeny of the early hominid face	Bromage, T.G.	1989	*Australopithecus*, Early Homo, *Paranthropus*	Mandible, frontal, zygomatic, maxilla, glabella, nasal
Surface Remodelling of the Facial Skeleton in Juvenile *Macaca mulatta:* Implications for Sexual Dimorphism	Wealthall, R.J.	2002	*Macaca mulatta*	Frontal, maxilla, zygomatic
Surface Bone Histology of the Occipital Bone in Humans and Chimpanzees	Mowbray, K.	2005	*Homo sapiens, Pan troglodytes*	Occipital
Remodeling Patterns of Occipital Growth: A Preliminary Report	Kranioti E.F., Rosas A., Garcia-Vargas S., Estalrrich A., Bastir M., Pena-Melian A.	2009	*Homo sapiens*	Occipital
Brief communication: Identification of bone formation and resorption surfaces by reflected light microscopy	Martinez-Maza C., Rosas, A., & Nieto-Diaz, M.	2010	*Homo sapiens, Pan troglodytes,* Neanderthals, *Homo heidelbergensis, Gorilla gorilla*	Mandible, frontal, zygomatic, maxilla, glabella, nasal
Bone remodeling of the *Homo heidelbergensis* mandible; the Atapuerca-SH sample.	Rosas A. & Martinez-Maza C.	2010	*Homo heidelbergensis*	Mandible
Bone remodelling in Neanderthal mandibles from the El Sidrón site (Asturias, Spain).	Martinez-Maza C., Rosas A, García-Vargas S., Estalrrich A., & de la Rasilla M.	2011	Neanderthal	Mandible
Facial Morphogenesis of the Earliest Europeans	Lacruz R.S., de Castro, J. M. B., Martinón-Torres, M., O'Higgins, P., Paine, M. L., Carbonell, E., Arsuaga, J. L., & Bromage, T. G.	2013	*Homo erectus, Homo antecessor*	Maxilla

Postnatal changes in the growth dynamics of the human face revealed from bone modelling patterns	Martinez-Maza C., Rosas, A., & Nieto-Díaz, M.	2013	*Homo sapiens*	Frontal, glabella, nasal, maxilla, zygomatic, mandible
Bone Modeling Patterns and Morphometric Craniofacial Variation in Individuals from Two Prehistoric Human Populations from Argentina	Brachetta-Aporta N, Martinez-Maza C., Gonzalez P.N., & Bernal V.	2014	*Homo sapiens*	Frontal, glabella, nasal, maxilla, zygomatic
Bone Growth Dynamics of the Facial Skeleton and Mandible in *Gorilla gorilla* and *Pan troglodytes*	Martinez-Maza C., Freidline S.E., Strauss A., Nieto-Diaz M.	2015	*Gorilla gorilla, Pan troglodytes*	Frontal, glabella, nasal, maxilla, zygomatic
Distinct growth of the nasomaxillary complex in Au. sediba	Lacruz R.S., Bromage T.G., O'Higgins P., Toro-Ibacache V., Warshaw J., & Berger L.R.	2015	*Australopithecus sediba*	Vomer, maxilla, nasal, palatine, frontal
Ontogeny of the maxilla in Neanderthals and their ancestors	Lacruz, R. S., Bromage, T. G., O'Higgins, P., Arsuaga, J. L., Stringer, C., Godinho, R. M., Warshaw, J., Martínez, I., Gracia-Tellez, A., de Castro, J. M. B., & Carbonell, E.	2015	Neanderthal	Maxilla
A quantitative approach for analyzing bone modelling patterns from craniofacial surfaces in hominins	Brachetta-Aporta N., Gonzalez P.N., & Bernal V.	2018	Hominins, *Homo sapiens*	Maxilla
Ontogeny of the human maxilla: a study of intra-population variability combining surface bone histology and geometric morphometrics	Schuh A., Kupczik, K., Gunz, P., Hublin, J., & Freidline, S. E.	2019	*Homo sapiens*	Maxilla
Variation in facial bone growth remodeling in prehistoric populations from southern South America	Brachetta-Aporta N., Gonzalez, P.N., & Bernal V.	2019	*Homo sapiens*	Frontal, maxilla. zygomatic
Intraspecific variability in human maxillary bone modeling patterns during ontogeny	Schuh A., Gunz P., Villa C., Kupczik K., Hublin J., & Freidline S.E.	2020	*Homo sapiens*	Maxilla

Additional Explanations for Mandibular Variation

Variation in bone modeling patterns in the mandible may also be caused by masticatory forces and diet. Two studies on rats and mice highlight differences attributed to mastication. For example, Anderson et al. (2014) sought to understand adaptation and plasticity in the mandible of mice and how bone modeling is affected by weaning. They separated mice into two distinct groups: those fed a hard pellet diet and those fed a soft ground pellet diet. The research primarily focused on mechanical advantage and musculature, and the mice fed the hard food diet had higher mechanical advantage. It is surmised this is due to hard pellet foods requiring a higher bite force compared to the ground pellet food. In another study conducted on rat mandibles, Yamada and Kimmel (1991) observed a relationship between diet and mastication. In this study, rats were once again split into two groups according to a diet of hard and soft food diets. At the conclusion of this study, the rats fed a soft diet had overall less bone volume, formation rate, and zones of condylar cartilage. However, by the time the rats reached 8 weeks of age, there was no difference in bone volume. Processes of mastication made a difference initially after weaning, but eventually evened out (Yamada and Kimmel, 1991). It is unknown why bone formation amounts remained consistent with a differing diet.

Subsistence strategy has also been attributed to the changes in mandibular morphology during ontogeny. Studies have shown there to be a distinction in morphology between populations who practice a hunter-gatherer subsistence practice and populations who are agriculturists (von Cramon-Taubadel, 2011; Noback & Harvati, 2014). Von Cramon-Taubadel's (2011) study found that shifting from one subsistence practice to another caused a considerable change to the shape of the mandible. However, as mentioned previously, numerous factors can contribute to mandibular shape change. The change found due to the shift in subsistence practice

was found to be a global phenomenon that led to insights on masticatory adaptation. Other studies contribute to this idea that mandibular and craniofacial shape change are caused by masticatory forces. A study conducted between farmers and hunter-gatherers of the Argentine Center-West reported a differentiation in craniofacial size caused by diet and masticatory factors (Sardi et al. 2006). Farmers reported a smaller craniofacial system compared to their hunter-gatherer counterparts. This was attributed to the low masticatory stress in the farmer diet and low protein diet. Mandibular shape changes with levels of bone strain associated with diet (Bouvier & Hylander, 1981; Corruccini and Beecher, 1982). Growth rates shift at the time of weaning, as children shift from a soft, liquid diet to a diet of hard food (Helm & German, 1996). In chewing hard food, an increased amount of force by the masticatory muscles is placed on bone, opposing the amount of force placed by suckling (German & Crompton, 1996).

In a study on *H. sapiens,* Brachetta-Aporta & Toro- Ibacache (2021) analyzed bone modeling patterns and the relationship of mechanical loading from bite in six skulls of juveniles and adults from archaeological sites in Argentina. The authors used both surface histology and finite element modeling to determine the extent bite can affect skull morphology. They found that the most strain from bite corresponded with areas containing high amounts of bone resorption. The authors suggested mechanical loading due to mastication and bite forces can enhance morphological differences in the mandible and areas of the cranium early in ontogeny. Von Cramon-Taubadel (2011) sought to identify mandibular variation through a cross-cultural comparative study of hunter-gatherer and agriculturalist populations. The results of this study showed that mandibular shape differences are present between hunter-gatherers and agriculturalists, and this was associated with diet, as other studies had shown the shift toward an agriculturalist lifestyle led to changes in mandibular shape. Hunter-gatherer populations

displayed shorter and more vertically oriented rami and coronoid processes on longer and more narrow mandibles. This study suggested that geographic location aside, morphological differences appeared as populations shifted subsistence economies.

Mandibular morphological changes can also be attributed to teeth being used as tools. For example, in a study conducted on craniofacial morphology of indigenous populations from Greenland, Hylander (1977) describes the numerous ways in which the indigenous people use their teeth and mouths as tools. Indigenous women were using their teeth to chew and soften sealskin for the fabrication of clothing and production of domestic items, while male individuals were using their teeth in hunting activities, such as holding a bow drill bit and shaping wood for a kayak. According to the author, the indigenous individuals are known for powerful jaws. Multiple factors acting concurrently on the mandible lead to differences in mandibular morphology.

Limitations of Current Literature

As stated previously, a common problem in analyzing bone modeling patterns is the lack of quantitative data, small sample size, and use of bone modeling maps that are vague and fail to accurately display the amount of bone resorption and formation that is occurring. For example, Brachetta-Aporta et al. (2014) studied the facial bones of 34 adult individuals from Argentina using surface histology and geometric morphometrics. The authors implemented surface histology; however, they created bone modeling maps simply based on the location of bone formation and resorption, rather than integrating the amount of bone resorption or formation that appeared.

Recently, Schuh et al. (2019, 2020 and 2021) developed methods to quantify and better visualize surface histology to understand the growth and development of the maxilla during

ontogeny. Schuh et al. (2020) studied the maxilla for three human populations: Western European, Inuit, and South African. As bone modeling patterns were scarcely quantified in the past, the authors combined surface histology and geometric morphometrics to visualize bone modeling patterns and shape differences through heat maps. According to the authors, the morphological differences in the maxilla reflected that found through microscopy. An overall similar pattern of bone modeling of the maxilla was seen between the populations that was reflected in facial morphology. Areas of bone resorption in adults were largely consistent with those observed in juveniles, although the amount of bone resorption is reduced.

Bone modeling patterns can be indicative of variability in modern human populations (Freidline et al., 2017). Though researchers such as Schuh et al. (2019, 2020) or Martinez-Maza et al. (2006, 2011) continue to make strides in this field, further research and standardization of bone modeling pattern research is necessary, such as the interpretation and identification of bone modeling patterns. Additionally, prior studies of the mandible were primarily looking at variation macroscopically or using destructive methods to analyze bone microscopically. Small sample sizes have been used due to time consuming histological methods and to preserve as many specimens as possible, creating a limit on the extent of the research. A broader, more diverse sample size is necessary to understand ontogenetic variability in bone modeling patterns of the human mandible. These patterns can then be used as a framework or guideline for interpreting fossil hominins. In case studies where the mandible is well-preserved and able to be replicated, studies fail to use a mixed-methods approach, such as integrating methods of geometric morphometrics and visualization of bone modeling patterns, instead opting for a qualitative, visual analysis (Martinez-Maza et al., 2013). There is no quantification of mandibular data, unlike that of the maxilla and other facial bones (Schuh et al., 2019; Brachetta-Aporta et al., 2014). Beyond the lack of quantified

data, there is no observer test for identifying bone formation or bone resorption. Thus, schematic map usage for analyzing bone modeling patterns is more prone to error (Martinez-Maza et al., 2013).

Aside from lack of quantitative data and sample size, discrepancies in the literature continue to persist as the terms "bone modeling" and "remodeling" are used interchangeably, although they describe two completely different processes. These contradictions in literature make it difficult to discern and differentiate between studies focusing on bone modeling and bone remodeling. For example, Barak (2020) and Allen & Burr (2019) state functional adaptation as resulting from bone modeling. However, studies such as that of Brachetta-Aporta & Toro-Ibacache (2021) fail to operationalize their terminology. Their study focused of the impact of masticatory load of facial bone surface remodeling (Brachetta-Aporta & Toro-Ibacache, 2021). Based on definitions in bone biology by Barak (2020), Allen & Burr (2019), and DiGangi & Moore (2013), the adaptation of facial bone surfaces due to processes of mastication should not be attributed to bone remodeling, but bone modeling. Problems in literature also persist as there are only two studies on surface histology of the *H. sapiens* mandible, only one of which is visualizing bone modeling maps (Martinez-Maza et al., 2013). This research will be a significant contribution to the literature by engaging in quantitative techniques and using a large sample of mandibles ranging throughout ontogeny.

The objective of this research is to quantify and visualize bone modeling patterns of the mandible in a diverse, ontogenetic sample of recent modern humans. This research will apply surface histological methods to explore bone modeling patterns between three diverse geographic groups, the Western European, Greenland Inuit, and South African (Khoe khoe and San) populations, to test for intra-populational and ontogenetic differences. The results will be

compared to Schuh et al.'s (2020) work on the maxilla for the same populations and age groups. Additionally, as the European sample is of known sex, this research will test for differences in bone modeling patterns between males and females in this sample.

CHAPTER 3 MATERIALS AND METHODS

Sample

The sample is comprised of a total of 48 individuals from three different geographic regions: Western European (Anatomical Institute of Strasbourg, France; University of Leipzig Anatomical Collection, Germany); Greenland Inuit (Laboratory of Biological Anthropology, University of Copenhagen, Denmark); and a South African sample composed of members of the Khoekhoe and San groups (Iziko South African Museum, Cape Town; McGregor Museum, Kimberley, South Africa, and Anthropological Collection of the Department of Human Biology, University of Cape Town). The ontogenetic age of the individuals ranges from birth to adulthood. The Western European sample is the only group in the study with known age and sex. The age for the Greenland Inuit and South African samples were previously estimated according to dental development by Schuh et al. (2019), following the criteria of AlQahtani, Hector, and Liversidge (2010). Sex for these two samples was never estimated due to juvenile sex estimation difficulties (Coquerelle et al., 2011). Following Schuh et al. (2019), each individual was placed in a dental age group according to dental eruption: AG1 consists of individuals with deciduous dentition in development; AG2, the first permanent molar in occlusion; AG3, the second permanent molar in occlusion; AG4, the third permanent molar has erupted. For each population, a total number of 16 individuals will be analyzed, with four in each age category (Table 1).

Table 2 Number of individuals in each population per age group. M represents males, F represents females, and U represents unknown sex.

Population	Age Group 1	Age Group 2	Age Group 3	Age Group 4	Total
Western European	4 (M 2, F 2)	4 (M 2, F 2)	4 (M 2, F 1, U1)	4 (M 1, U 3)	16
Greenlandic Inuit	4	4	4	4	16
South African	4	4	4	4	16
Total	12	12	12	12	48

Data Collection

Following previous studies (Schuh et al., 2019, 2020), casts of the mandibles were made and analyzed using the Zeiss Smartzoom 5 digital microscope. For each individual, four negative low-viscosity silicone (President Plus light body, Coltène/Whaledent AG, Switzerland) replicas of the periosteal surface of the mandibles were previously collected by Drs. Alexandra Schuh and Sarah Freidline. The negative replicas extend from the mandibular condyle to the mental symphysis on both the left and right sides, and buccal (exterior) and lingual (interior) surfaces (Figure 2).

Figure 2 Negative surface replica on the mandible.

A sample of positive replicas of the right side of the mandible were previously created from epoxy resin (5 Minute Epoxy Epoxidharz 2 L-Kleber transparent, Devcon) by injecting the resin into the mold created from the negative replica. The remaining casts were made using an alternative epoxy resin (J-B Weld Ultrarez UV-Resistant Coating & Casting Epoxy) and employing the same technique (see Appendix A for full instructions).

After drying, a grid was written on the positive replicas using a permanent marker. These grids are composed of 5x5 mm squares and 2.5x2.5 mm sub-squares (Figure 3). This effectively works as a coordinate plane. The positive replicas are viewed with the Smartzoom 5 microscope using the PlanApo D 1.6x/0.1, FWD = 36mm objective lens. Each aspect of the grid was investigated for evidence of bone resorption, i.e., the presence of Howship's lacunae, and bone formation, visible as collagen fiber bundles (Figure 4).

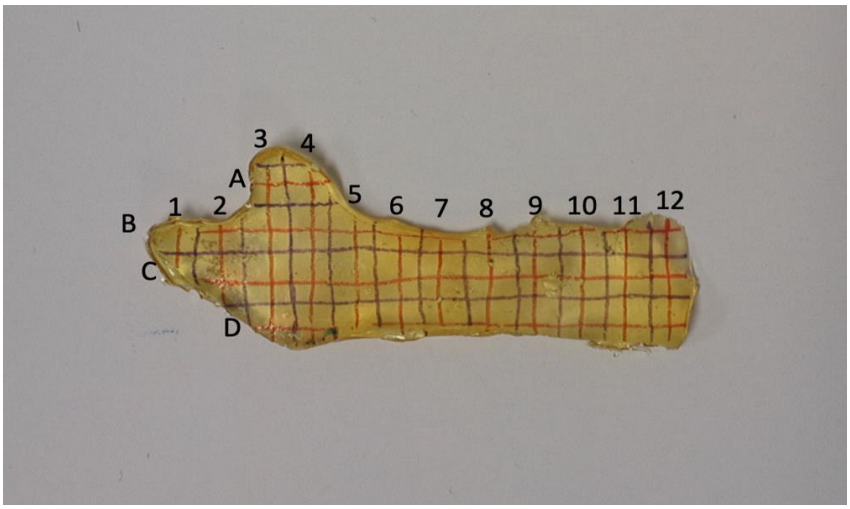


Figure 3 Positive replica of the right buccal side of the mandible with grid lines and labeling. Blue lines indicate 5 mm x 5 mm squares. Red lines divide squares into four 2.5 mm x 2.5 mm sub-squares.

Each grid is labeled to retain record of location of bone resorption. Alphabetical labels run vertically downward starting with the letter A, and numbered labels run horizontally beginning with the number 1. To standardize the work being conducted on both the lingual and buccal surfaces of the mandible, the first square (A1) will always start at the most upward and outward square on the mandibular condyle (Figure 3). Thus, the labeling of the lingual surface of the mandible should be the inverse of the buccal surface of the mandible.

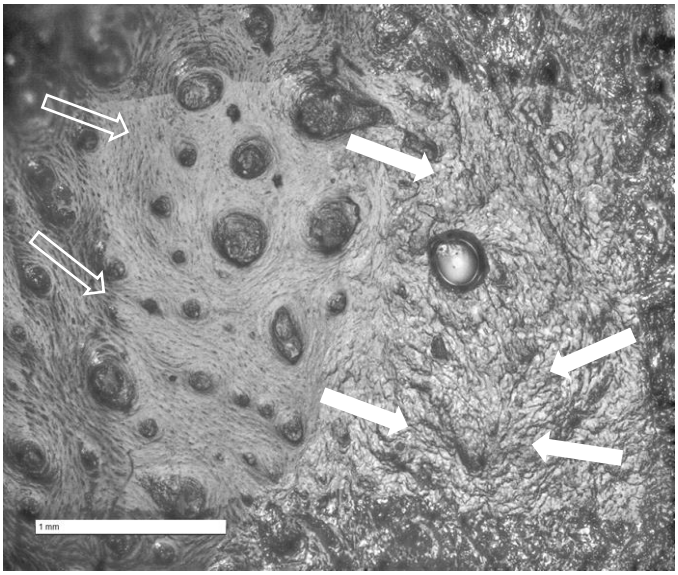


Figure 4 Bone modeling activity. Bone resorption indicated by Howship's lacunae (white arrows) and bone formation (open white arrows) .

Data Quantification and Qualification

Sub-squares (2.5 mm x 2.5 mm) with bone resorption were imaged. The percentage of bone resorption is calculated for each square of the grid using the software ImageJ (Schneider et al, 2012). Manually selected bone resorptive areas are quantified by using the "Polygon selection" tool and drawing a perimeter around each field of resorption. The surface area of each of these fields are recorded and a total percentage of bone resorption per square were calculated. The total percentage of bone resorption per individual is also calculated by the sum of bone resorption in each square and dividing this sum by the total surface of the specimen (Figure 5). A mean percentage of resorption is also calculated for each age group and population.

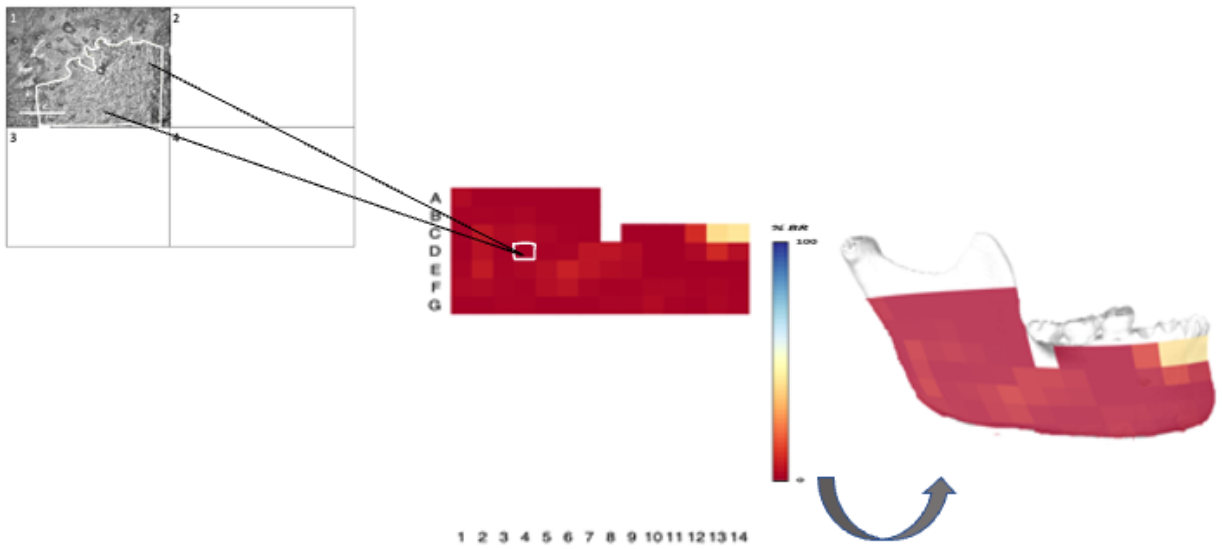


Figure 5 Bone resorption calculation and digital mapping (example on the mandible). Left: Bone resorption calculated per sub-square. Center: Digital map of a mandible with percentages of bone resorption (BR) per square. Right: Projection of the digital map onto 3D surface model

Bone modeling patterns for each age, population, and sex were visualized and compared through the use of digital maps (Figure 5). The maps were created and computed in RStudio (RStudio Team, 2020) by equating color and percentage of bone resorption for each 5 mm x 5 mm square. Computed tomography scans for the same individuals were previously taken at a 0.2-04 mm (BIR ACRIS 225/300) resolution and from these scans, three-dimensional surface models were created using the Avizo software (Thermo Fisher Scientific). Mean digital bone modeling maps were projected onto a corresponding 3D surface model for population and age group by using Photoshop (Adobe Inc., 2019). 3D surface models were edited using Geomagic Studios (Research Triangle Park, NC) to visualize the anatomical positions of bone formation and resorption for each surface of the mandible (Figure 5).

Following Schuh et al. (2019), the mean percentage of bone resorption for each square of the grid is used to create a bone modeling map where warmers colors indicate lower percentage of bone resorption and cooler colors indicate a higher percentage of bone resorption (Figure 5).

Inter- and Intraobserver Error Testing

In addition to analyses conducted on bone modeling patterns, an interobserver error test was conducted. The observers consisted of Dr. Schuh, experienced in identifying bone resorption, as well as an inexperienced undergraduate student and myself. This test is used to consider the subjectivity in identifying bone modeling patterns on bone surfaces. To do so, one individual from each age group and population was chosen (totaling 12 individuals) and a randomized selection of 10 total images of each individual taken from the digital microscope were scored. Five images came from the lingual surface of the mandible while the remaining five images are from the buccal surface. The observers selected areas of bone resorption as seen in the image. The areas of bone resorption were compared and an intraclass correlation was computed using RStudio to measure the degree of agreement between the raters.

An *intraobserver* error test was performed by myself at the start, midpoint, and conclusion of data collection. A two-way intraclass correlation (ICC) test from the package "irr" in RStudio was performed across the sample for each individual and surface (Gamer et al., 2019; Fisher, 1954). The 12 individuals from the interobserver error test were used for the intraobserver error test. A selection of five randomized images for the lingual surface of the mandible and five images for the buccal surface were analyzed, and the same images were used each time. Similarly, areas of bone resorption were selected and compared at each stage of the research. An intraclass correlation was computed to measure the agreement between the ratings at each stage of the research, and the results were statistically significant if $\alpha < 0.05$.

Data Analysis

Once the data was collected, the mean maps of bone resorption were computed for each of the four age groups and for each population to explore differences associated with ontogeny. A standardized grid was created in RStudio to compare mean bone modeling maps between the younger (smaller) and older (larger) individuals. Mean maps of each age group and population were projected onto a 3D surface model of a single individual that fits in the category for both population and age. Surface models are selected based on the above criteria as well as preservation. Statistical analyses are conducted to determine the variability of bone modeling pattern. To test if population differences and age differences can be found on the mandible, a PERMANOVA test was performed using the "adonis2" function from the package "vegan" (Schuh et al., 2020; Oksanen et al. 2022). PERMANOVA is a permutational nonparametric statistical test alternative to the MANOVA (Anderson, 2001). If the results were significant at $\alpha < 0.05$, a post-hoc pairwise test using the function "pairwise.adonis" was conducted in RStudio using the "pairwiseADONIS" package (Arbizu & Monteux, 2022). Mean and standard deviations for each age group and population were calculated. Certain results were expected based on the following hypotheses:

1. H_0*:* Bone modeling patterns are the same within a population from infancy to adulthood. *Analysis and Expectations:* For each population, an average percentage of bone resorption will be calculated and visualized for each age group and compared between age groups. There are little differences in bone modeling patterns between age groups as depicted in the bone modeling maps. Results of the PERMANOVA will show no significant difference between age groups

within a population. Alternatively, significant differences and/or differences in heat maps would suggest bone modeling patterns change throughout ontogeny.

2. H_o*:* Bone modeling patterns are the same between populations.

Analysis and Expectations: For each population, an average percentage of bone resorption will be calculated and visualized for each age group and compared between populations. There are little differences in bone modeling patterns between populations as depicted in the bone modeling maps. Results of the PERMANOVA will show no significant difference between populations. Alternatively, significant differences and/or differences in heat maps would suggest *H. sapiens* do not have a shared bone modeling pattern that persists throughout ontogeny.

3. H_o*:* Bone modeling patterns are the same between the individuals of known sex for the Western Europeans (the only group in the sample with known sex).

Analysis and Expectations: For each individual, bone resorption will be calculated and visualized, and compared between sexes. There are little differences in bone modeling patterns between sex as depicted in the bone modeling maps. Differences in heat maps would suggest sex differences can be seen in juvenile bone modeling patterns.

CHAPTER 4 ERROR TESTING RESULTS

Intraobserver Error Test

An intraobserver error test was conducted after taking a series of measurements from the lingual (interior) and buccal (exterior) surfaces of the mandible. Five images of each surface, selected at random, were observed for the presence of bone resorption for one individual from each population and age group, totaling 120 images. Areas of bone resorption were quantified using the software ImageJ and recorded (See Appendix B). The intraobserver error test was between three separate observations, over the course of data collection, for each individual and surface. These observations were taken at the beginning, middle, and end of data collection. A two-way intraclass correlation (ICC) was performed across the sample for each individual and surface (Gamer et al., 2019; Fisher, Oliver, & Boyd, 1954). The intraclass correlation operates based on groups of observations, in this case, five images per surface (Heagerty & Delong, 2022). For example, the intraclass correlation (ICC) takes the mean of the measurements (five for each surface and individual) of bone resorption for each observation of an individual and surface and measures the reliability between observations (Barcikowski, 1981). For the purposes of this research, an observation is the group of five images quantified for one surface and individual.

Reliability is measured through a correlation coefficient. The coefficient values can extend from -1 to 1. A negative intraclass correlation coefficient value occurs when the variability within a single group is more than the variability across groups of observations (Gonzalez, 1999; Norton et al., 1996; Zeger et al., 1988). For example, for the intraobserver error test, it would imply that the measurements for the first observation are more variable to one another than the measurements between all three observations. Positive ICC values from 0 to 1 represent the level of reliability where poor reliability is indicated by values less than 0.5, moderate reliability between 0.5 and 0.75, good reliability between 0.75 and 0.90 and excellent reliability indicated by a coefficient

greater than 0.90 with a 95% confidence interval (Koo & Li, 2016). Table 3 shows the ICC coefficient and p-value for the intraobserver test. The values of the coefficient range from 0.0805 to 0.97 with a majority of the values showing poor and moderate reliability.

Table 3 ICC coefficient and p-value of each individual for intraobserver test. * denotes a significant p-value

	UCT 195		**SAM AP 3027**		**KAL_0869**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.507	0.318	0.5	0.926	0.336	0.342
P-value	0.0179*	0.054	0.0272*	9.76E-05*	4.69E-02*	4.75E-02*
	SAM AP 6340		**SAM AP 4844**		**KAL 0674**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.211	0.494	0.571	0.58	0.63	0.0805
P-value	1.07E-01	0.0648	1.09E-02*	1.22E-02*	9.96E-03*	1.55E-01
	KAL 0707		**1906-07-37**		**1902-144-594**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.516	0.97	0.92	0.755	0.805	0.789
P-value	2.26E-02*	5.55E-08*	2.31E-05*	1.28E-03*	2.64E-03*	1.47E-03*
	KAL 0401		**1892-93-286-180**		**1879-73-21**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.693	0.868	0.607	0.762	0.519	0.722
P-value	0.00587*	6.32E-04*	1.18E-02*	1.84E-03*	1.74E-02*	2.60E-03*

Figure 6 displays the data gathered from the intraobserver test where each color represents a set of measurements taken at a different point during data collection (beginning, middle, and end). The mean measurement of bone resorption in the second (red) and third (green) observations, or middle and end of data collection, respectively, are more similar to one another than the first observation (blue). Using the same data, a second error test using ICC was

performed using only the second and third observations (Table 4). The intraclass correlation coefficient for this test showed improved levels of reliability. The lowest coefficient was equal to 0.874, an indicator of good reliability. The remainder of the coefficients were greater than 0.9, displaying an excellent reliability between the observations taken at the midpoint and end of data collection. The second intraobserver error test reiterated the information from Figure 6: the first observation caused a decrease in reliability of the first error test.

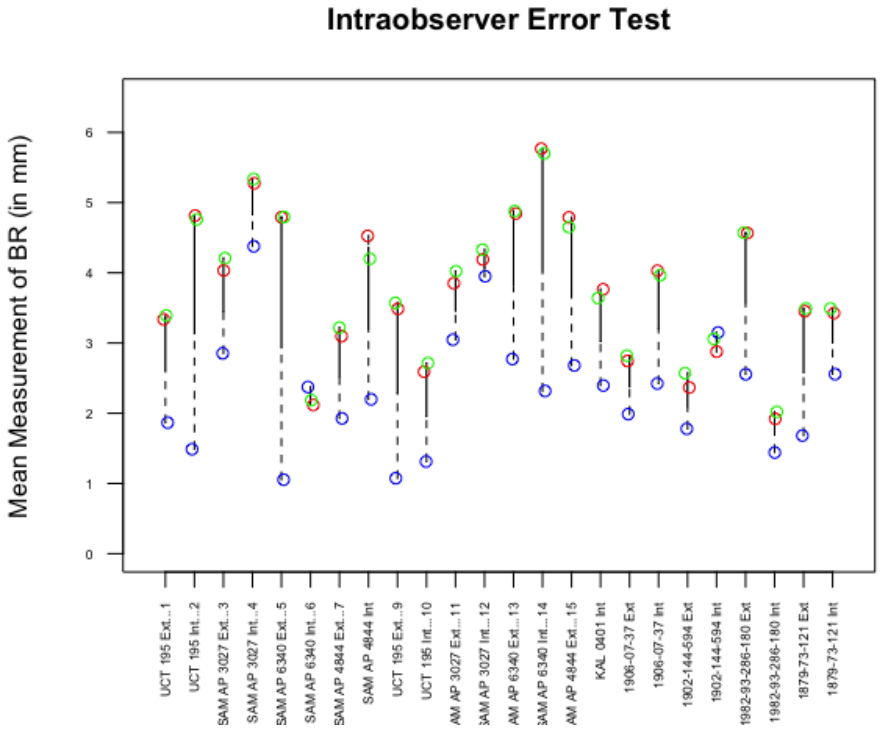


Figure 6 Intraobserver Error Test plot. Blue represents the first observation, red represents the second observation, and green represents the third observation.

Table 4 ICC coefficients and p-values of each individual for intraobserver error test with two observations. * indicates a significant p-value

	UCT 195		**SAM AP 3027**		**KAL_0869**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.999	0.999	0.989	0.999	0.998	0.992
P-value	1.17E-05*	2.52E-08*	2.46E-05*	5.81E-07*	1.25E-06*	2.47E-03*
	SAM AP 6340		**SAM AP 4844**		**KAL 0674**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.997	0.998	0.985	0.948	1	0.874
P-value	9.30E-06*	6.74E-05*	2.89E-05*	0.000631*	1.76E-08*	1.39E-02*
	KAL 0707		**1906-07-37**		**1902-144-594**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.99	0.996	0.999	0.996	0.974	0.976
P-value	4.75E-04*	4.45E-06*	8.99E-07*	5.39E-06*	1.35E-02*	8.32E-04*
	KAL 0401		**1892-93-286-180**		**1879-73-21**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.992	0.99	0.98	0.978	0.998	0.999
P-value	6.91E-06*	1.95E-05*	4.39E-04*	8.70E-05*	2.35E-07*	1.31E-06*

Interobserver Error Test

Two interobserver error tests were conducted. The first was between Dr. Schuh, an experienced observer and I, while the second included an undergraduate student, an inexperienced observer just learning how to identify bone resorption. The second observation from the intraobserver error test was used as my contribution to the interobserver error tests. The coefficient values ranged from -0.0418 to 0.99, with a majority of the results indicating a good to excellent reliability (Table 5). Lower values are indicated in bold. Negative coefficients are seen in this data,

unlike previous error testing. This means that for one of the observers, there was more variability between their own measurements than between observers.

Table 5. ICC coefficients and p-values of each individual for interobserver error test with two observations. Lower values are indicated in bold. * indicates a significant p-value

	UCT 195		**SAM AP 3027**		**KAL_0869**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	**0.287**	0.944	**-0.204**	0.952	0.938	**-0.0418**
P-value	0.207	0.000745*	0.617	0.0014*	0.0142*	0.543
	SAM AP 6340		**SAM AP 4844**		**KAL 0674**	
	Ext	**Int**	**Ext**	Int	**Ext**	**Int**
ICC	0.98	**0.0904**	0.947	0.97	0.971	0.881
P-value	0.000395 *	0.23	0.000625*	0.0146*	0.000942*	0.0123*
	KAL 0707		**1906-07-37**		**1902-144-594**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	**0.381**	0.771	0.821	0.99	0.659	0.694
P-value	0.138	0.019*	0.0114*	5.04E-05*	0.0746	0.0717
	KAL 0401		**1892-93-286-180**		**1879-73-21**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.986	0.954	0.888	0.766	0.918	0.902
P-value	0.000138 *	0.000757*	0.0325*	2.84E-02*	0.00777*	0.00276*

When integrating the inexperienced observer's data, it was found that overall, the coefficient values decreased. Coefficients ranged from -0.0941 to 0.993, with fewer individuals showing excellent reliability. Instead, many coefficients reflected poor to moderate reliability (Table 6). Visually, differences can be seen in Figure 7. The experienced observer in green, and I, red, show greater similarities in the measurements as seen in Table 5, compared to the inexperienced observer in blue. Measurements from the inexperienced observer varied more and

were often much lower than the experienced observer and me. This variation leading to the lower ICC coefficients seen in Table 6.

Table 6 ICC coefficients and p-values of each individual for interobserver error test with three observations. * indicates a significant p-value

	UCT 195		**SAM AP 3027**		**KAL_0869**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.177	0.104	-0.0793	0.118	0.277	0.355
P-value	0.191	0.225	0.566	0.207	1.24E-01	7.53E-02*
	SAM AP 6340		**SAM AP 4844**		**KAL 0674**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.917	0.171	0.957	0.967	0.665	-0.112
P-value	9.17E-05*	0.0626	4.09E-07*	6.32E-07*	4.22E-03*	8.58E-01
	KAL 0707		**1906-07-37**		**1902-144-594**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	-0.0941	0.463	0.889	0.769	0.271	-0.281
P-value	6.21E-01	3.73E-02*	3.06E-05*	1.08E-03*	1.24E-01	7.75E-01
	KAL 0401		**1892-93-286-180**		**1879-73-21**	
	Ext	**Int**	**Ext**	**Int**	**Ext**	**Int**
ICC	0.909	0.559	0.76	0.662	0.993	0.852
P-value	8.83E-05*	1.43E-02*	1.13E-03*	4.10E-03*	5.84E-06*	1.76E-04*

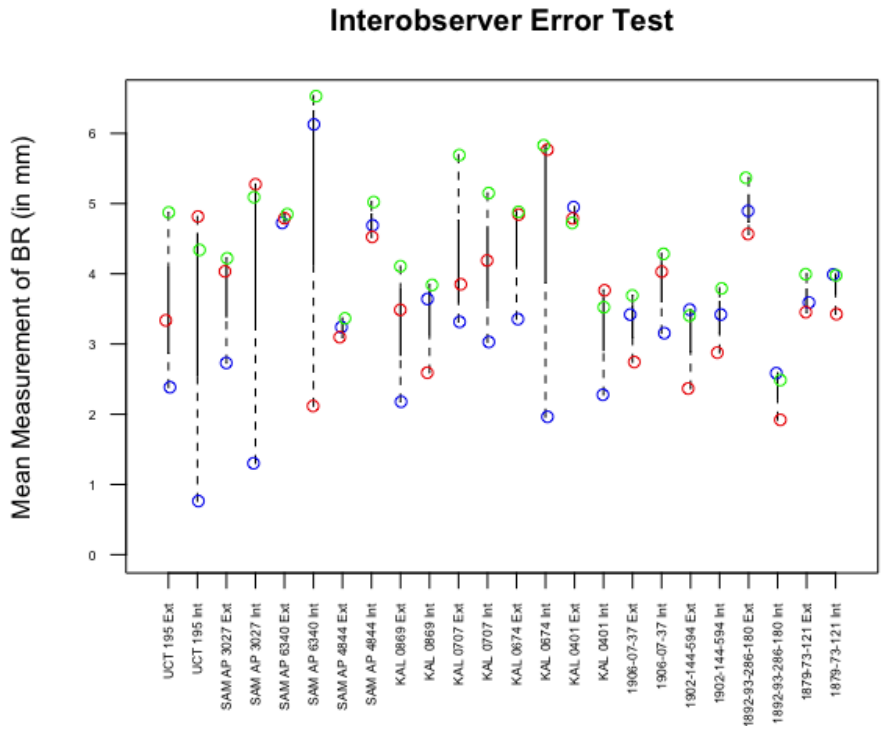


Figure 7 Interobserver Error Test plot. Blue represents the inexperienced observer, red represents the researcher, and green represents the experienced observer.

CHAPTER 5 HYPOTHESIS TESTING RESULTS

Descriptive Statistics

The results from the study display differences in the amount of total bone resorption for a population between age groups. Age groups are referred to in shorthand as AG and populations are referred to as WE for the Western European, GI for the Greenlandic Inuit, and SA for the South African. The buccal surface may be referred to as the exterior (ext) and the lingual surface may be referred to as the interior (int). The South African and Greenlandic Inuit populations exhibit a similar trend in the total percentage of bone resorption for both the lingual and buccal surfaces (Tables 7 and 8, Figure 8). After AG1, AG2 increases in the mean total percent of bone resorption. From here, the percentage decreases until AG4. This trend does not occur for the Western European population. The buccal surface of the Western European population starts off with the highest amount of bone resorption for AG1 and decreases continuously until the last age group (Table 7). The lingual surface for the Western European population has a higher amount of resorption for AG1 and decreases for AG2. The amount increases again for AG3 and decreases for AG4 (Table 8). AG1 Western European population has the highest mean of total percentage of bone resorption at 10.4% for the buccal surface and AG4 Greenlandic Inuit has the lowest percentage at 1.1% (Table 7). The AG1 Western European has the greatest standard deviation at 8.7 and the AG4 Greenlandic Inuit has the lowest at 0.7 for the buccal surface. For the lingual surface, AG2 Greenlandic Inuit population has the highest mean of total percentage of bone resorption at 7.3%. and AG4 Western European population has the lowest percentage at 2.0% (Figure 8, Table 8). The AG1 Western European has the greatest standard deviation at 7.1 and the AG4 Greenlandic Inuit has the lowest at 1.9 for the lingual surface (Figure 8, Table 9)

Table 7 Descriptive statistics, mean and standard deviation (SD) for each age group and population (buccal surface).

	Western European		Greenlandic Inuit		South African	
Age Group	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
1	10.4	8.7	6.9	5.5	4.5	4.7
2	4.7	1.8	7.1	9.5	6.1	5.7
3	4.6	1.8	2.0	1.6	3.7	4.8
4	1.7	2.6	1.1	0.7	3.5	3.1

Table 8 Descriptive statistics, mean and standard deviation (SD) for each age group and population (lingual surface).

	Western European		Greenlandic Inuit		South African	
Age Group	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
1	6.6	7.1	2.7	1.9	2.5	2.9
2	4.4	2.0	7.3	6.7	4.8	3.7
3	7.0	4.7	2.7	3.0	4.2	5.2
4	2.0	1.9	2.2	1.9	3.0	2.9

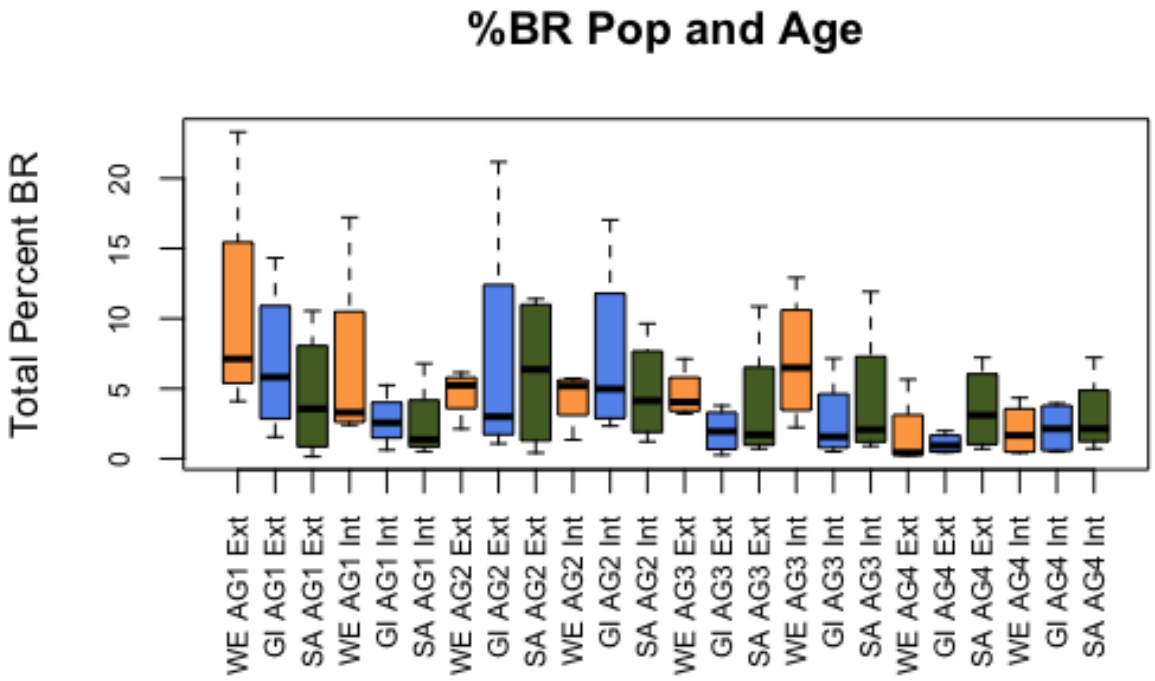


Figure 8 Box and whisker plot for total percentage of bone resorption for each population and age group. Tan indicates the Western European, blue represents the Greenlandic Inuit, and green represents the South African. The black bar in the box is the median, the whiskers are the minimum and maximums of the data, the area of the box above the median is the upper quartile, and the area of the box below the median is the lower quartile value.

Projections of Digital Bone Modeling Maps

The projections of the digital maps onto the 3D surface models gives a visual representation of the quantification of bone resorption (Figure 10 and 11). Overall, a shared pattern of bone resorption is observed for all populations for the first three age groups. For the buccal surface, AG1 has bone resorption that appears to be distributed between two areas on the mandible (Figure 10). Resorption is clustered along the mental symphysis and in the area where the ramus reaches the mandibular notch. AG2 has more bone resorption along the mandibular ramus in the area at the insertion site for the masseter muscle and a lesser amount of resorption continuing from the body to the mental symphysis (See Figure 9 for reference). AG3 has a

similar pattern to AG2 but with a lesser amount of resorption. AG4 shows a decreased amount of bone resorption compared to the previous age groups and the location of bone resorption is more variable. The Greenlandic Inuit population and South African population has resorption along the ramus and body. However, the Greenlandic Inuit displays more resorption beneath incisor and anterior dentition. The Western European population has resorption on the body.

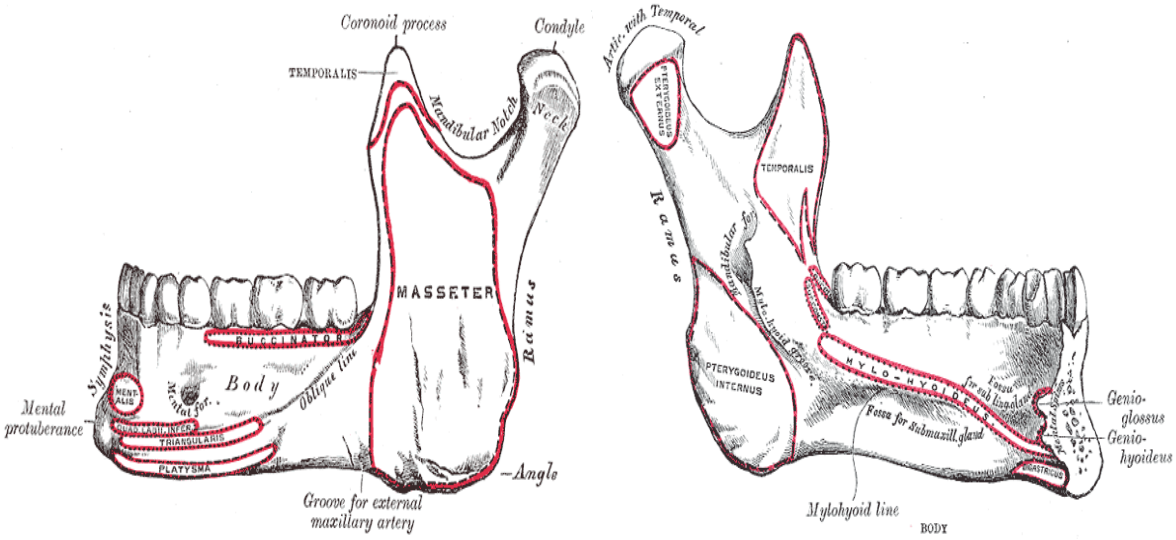


Figure 9 Mandible with marked muscle attachments. Left: lateral aspect of the mandible showing attachment site of the masseter. Right: medial aspect of the mandible showing attachment sites of the temporalis, pterygoid medialis, and pterygoid lateralis (Gray, 1918).

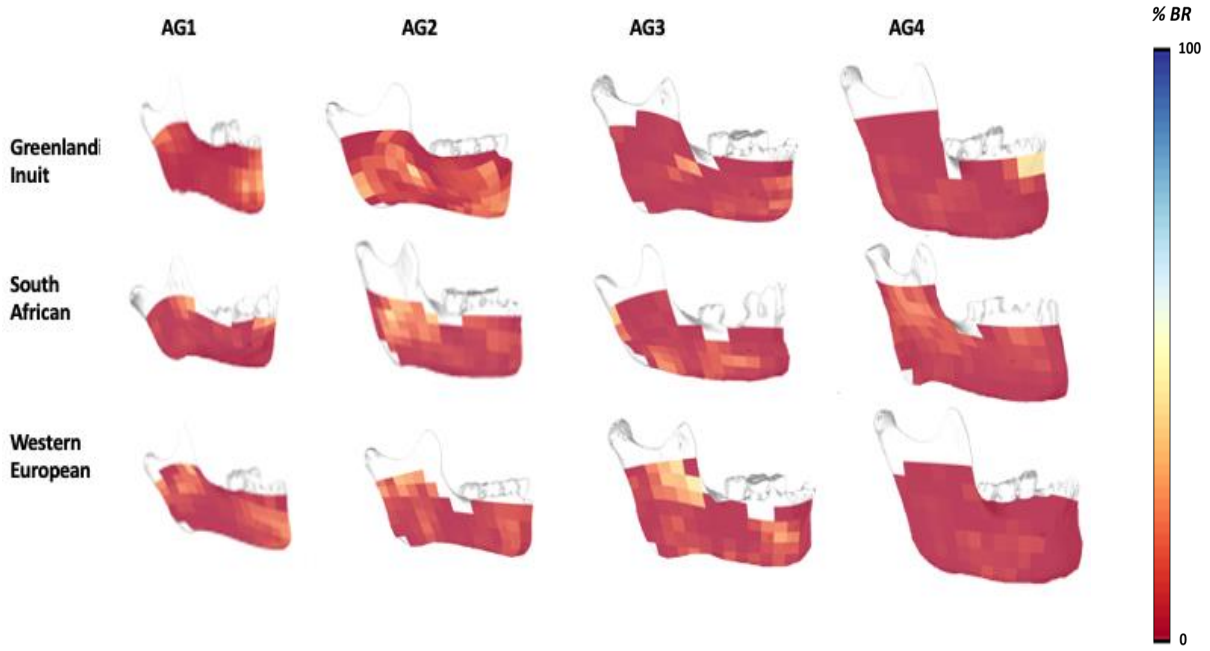


Figure 10 Projection of mean maps onto surface models (buccal surface). Left: Projection, right: scale bar. Warmer colors indicate a low percentage of bone resorption and cooler colors indicate a high percentage of bone resorption.

The results for the lingual, or interior surface, are similar to the buccal, or exterior mandibular surface. On the lingual surface, there an overall shared pattern of bone resorption for the four age groups (Figure 11). AG1 has a fairly distributed amount of bone resorption across the surface of the mandible with the Greenlandic Inuit having a lesser amount around the attachment site for the pterygoid medialis. AG1 South African and Western European resemble one another. Resorption for AG2 becomes more pronounced at the intersection of the body and ramus, and on the ramus. One main difference that can be seen is the presence of a blue square on the map of an AG2 individual from the Western European sample. This means that this square had a high amount of bone resorption. Being the only blue square, it was the highest amount of resorption for this study. AG3 has a similar pattern to that of AG2 but is more pronounced in some groups (e.g., Western European). AG4 is less variable for the lingual surface

of the mandible. Visually, the Greenlandic Inuit and South African populations more closely resemble one another with a low amount of resorption overall, but the highest concentration of resorption along the mylohyoid groove. The Western European population has a similar pattern but with less pronounced resorption.

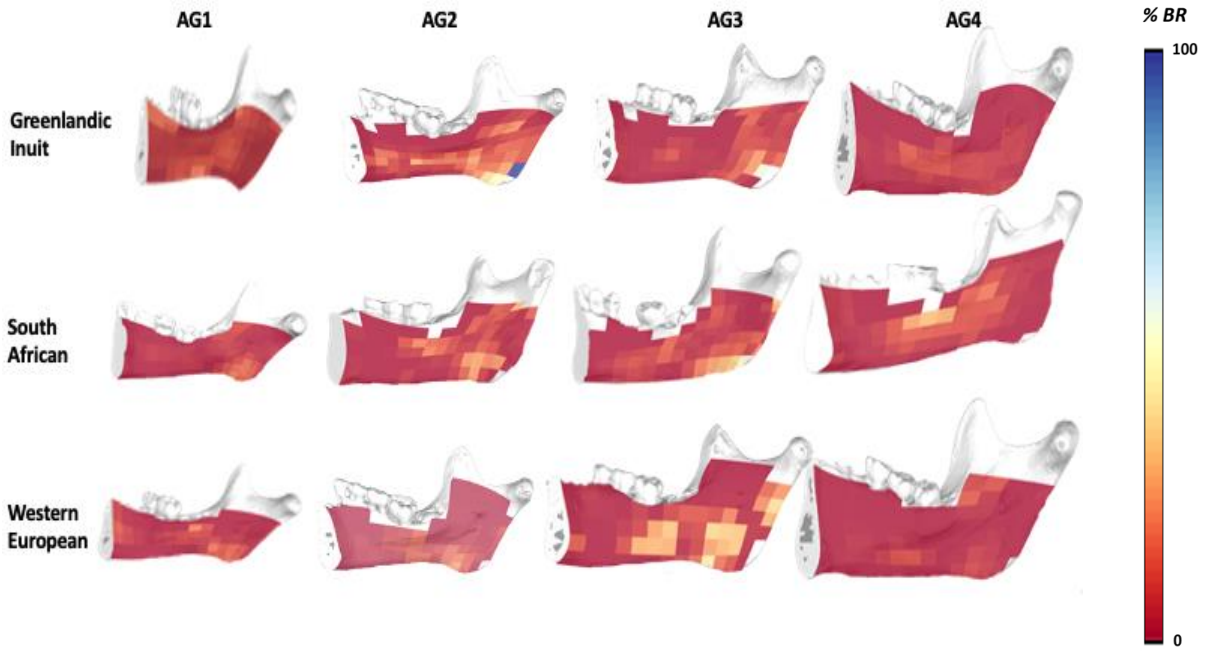


Figure 11 Projection of mean maps onto surface models (lingual surface). Left: Projection, right: scale bar. Warmer colors indicate a low percentage of bone resorption and cooler colors indicate a high percentage of bone resorption.

Hypotheses

1. Ho: Bone modeling patterns are the same within a population from infancy to adulthood

A PERMANOVA was conducted on each population to determine if there were statistically significant differences between the age groups within a population. Measurements of squares from the buccal and lingual surfaces were pooled to evaluate overall differences in percentages of resorption. The F-statistic for the South African and Greenlandic Inuit

populations each were 1.11 and 1.14, respectively (Table 9). Neither statistic for the South African and Greenlandic Inuit was significant, meaning there are no differences in the bone modeling pattern between age groups. The F-statistic for the Western European population was 1.55 and significantly different with a p-value of 0.003. A post-hoc pairwise test was used to determine which age groups were significantly different (Table 10). Statistically significant difference were found between AG1 and AG4, AG2 and AG4, as well as A3 and AG4. However, the adjusted p-values were not significant.

Table 9 Results of PERMANOVA testing. Degree of freedom(df), sum of squares (SumOfSqs), coefficient of determination (R^2), F-statistic (F), and p-value. An alpha level of 0.05 was used for all statistical tests * Indicates a significant p-value.

Population	df	SumOfSqs	R^2	F	p-value
WE	3	1.13	0.28	1.55	0.003*
GI	3	1.02	0.22	1.14	0.202
SA	3	0.91	0.22	1.11	0.265

Table 10 Results of post-hoc testing. Degree of freedom(df), sum of squares (SumOfSqs), coefficient of determination (R^2), F-statistic (F), and p-value. An alpha level of 0.05 was used for all statistical tests. * Indicates a significant p-value.

Pairs	df	SumOfSqs	R^2	F	p-value	p-value(adjusted)
1 vs 2	1	0.22	1.24	0.17	0.103	0.618
1 vs 3	1	0.24	1.04	0.15	0.389	1.000
1 vs 4	1	0.55	1.74	0.23	0.029*	0.174
2 vs 3	1	0.18	1.09	0.15	0.381	1.000
2 vs 4	1	0.53	2.07	0.26	0.019*	0.114
3 vs 4	1	0.53	1.73	0.22	0.025*	0.150

2. Ho: Bone modeling patterns are the same between populations.

A PERMANOVA was performed in RStudio to determine if there were significant differences between populations for each age group. The results of this test found no significant difference between the populations at each age group (Table 11). An additional PERMANOVA was conducted between all populations and pooled age groups (Table 12). The p-value was

significant at $p = 0.03$. As such, a post-hoc pairwise test was used to identify the populations for which there were differences (Table 13). The results from the post-hoc test found a significant difference between the Greenlandic Inuit and Western Europeans when age groups are pooled together. The adjusted p-values, however, did not reflect a significant difference between the three populations.

Table 11 Results of PERMANOVA testing. Degree of freedom(df), sum of squares (SumOfSqs), coefficient of determination (R^2), F-statistic (F), and p-value An alpha level of 0.05 was used for all statistical tests. * Indicates a significant p-value.

Age Group	df	SumOfSqs	R^2	F	p-value
1	2	0.66	0.20	1.13	0.246
2	2	0.57	0.23	1.39	0.065
3	2	0.63	0.21	1.16	0.209
4	2	0.65	0.18	1.00	0.453

Table 12 Results of PERMANOVA testing (age pooled). Pairs, Degree of freedom(df), sum of squares (SumOfSqs), coefficient of determination (R^2), F-statistic (F), p-value and p-value (adjusted). An alpha level of 0.05 was used for all statistical tests. * Indicates a significant p-value

Df	SumOfSqs	R^2	F	p-value
2	0.76	0.06	1.34	0.033*

Table 13 Results of post-hoc testing. Pairs, Degree of freedom(df), sum of squares (SumOfSqs), coefficient of determination (R^2), F-statistic (F), p-value and p-value (adjusted) An alpha level of 0.05 was used for all statistical tests. * Indicates a significant p-value.

Pairs	df	SumOfSqs	R^2	F	p-value	p-value(adjusted)
GI vs SA	1	0.39	1.33	0.04	0.107	0.321
GI vs WE	1	0.43	1.49	0.05	0.040*	0.120
SA vs WE	1	0.32	1.17	0.04	0.226	0.678

3. Ho: Bone modeling patterns are the same between the individuals of known sex for the Western Europeans (the only group in the sample with known sex).

Mean maps were computed by biological sex for the Western European population (Figure 12). The female buccal map and male buccal map had more resorption on the ramus and

in the direction of the mental symphysis. Visually, the lingual surface of the females had a decreased amount of resorption compared to the males. The male lingual surface had more resorption at the attachment site of the pterygoid medialis and mylohyoid groove.

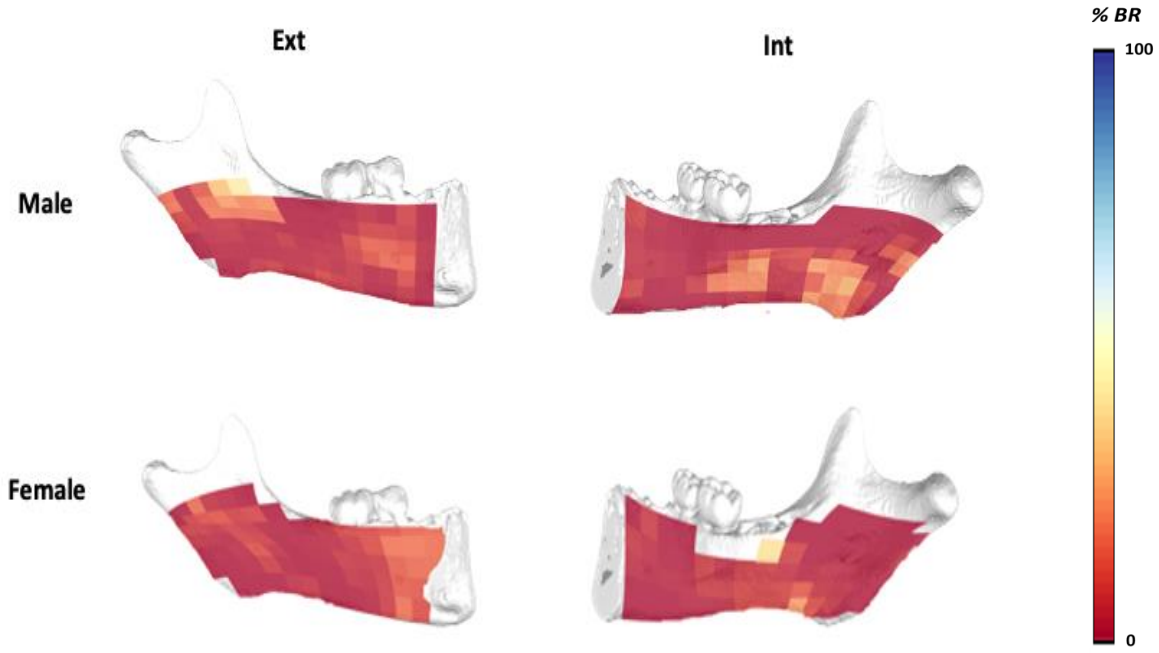


Figure 12 Projection of mean maps onto surface models (Pooled Age). Left: Projection, right: scale bar. Warmer colors indicate a low percentage of bone resorption and cooler colors indicate a high percentage of bone resorption.

To see if sex differences could be observed through bone modeling patterns during ontogeny, mean maps were created for the first and second age groups for both males and females. This was for both the lingual and buccal surfaces of the mandible. Maps were only made for the first two age groups because they had an equal number of males and females (Figure 13). AG1 males have more resorption closer to the mandibular notch on the buccal surface. AG1 females have resorption where the body meets the ramus, in the area of the oblique line. On the lingual surface, AG1 males have resorption along the body and at the attachment site

for the pterygoid medialis. AG1 females have resorption on the ramus and closer to the mandibular symphysis. The buccal surface of AG2 males and females are similar as resorption is in the area where the body meets the ramus. The lingual surface for AG2 males and females are also similar with resorption along the mylohyoid groove.

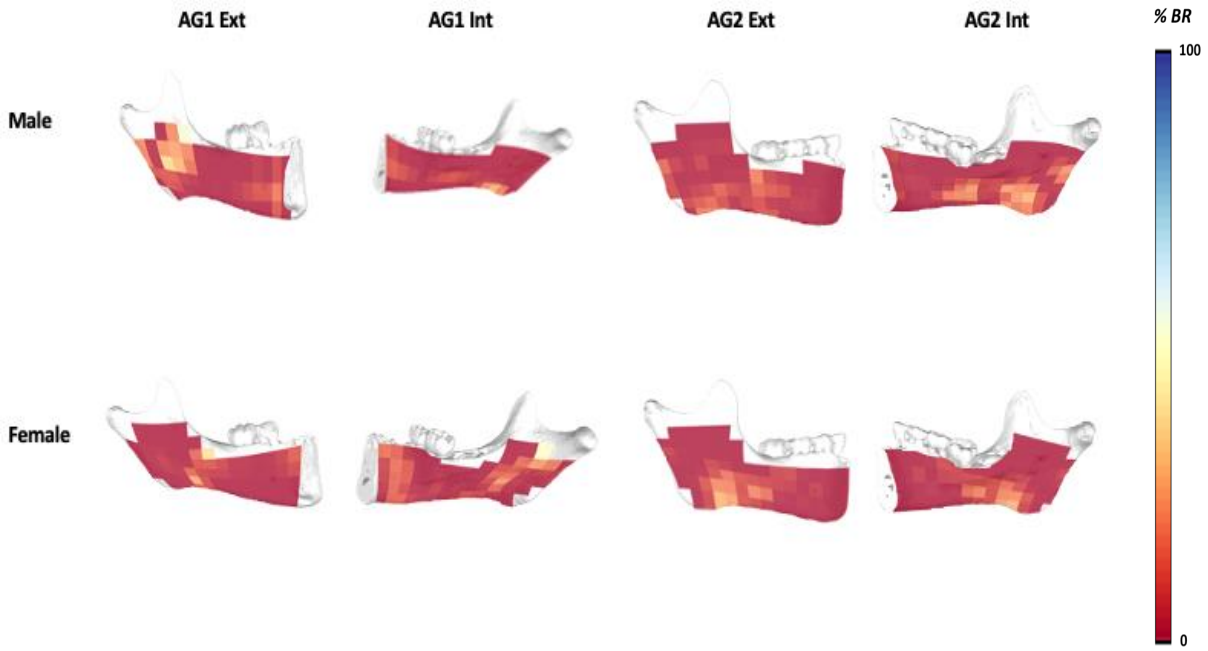


Figure 13 Projection of mean maps onto surface models (Sex for first two age groups). Left: Projection, right: scale bar. Warmer colors indicate a low percentage of bone resorption and cooler colors indicate a high percentage of bone resorption.

CHAPTER 6 DISCUSSION AND CONCLUSION

Error Tests

Intraobserver Error Test

An intraobserver error test was conducted to determine the agreement between observations (i.e., identification and quantification) of bone resorption throughout the research project. To recall, testing was conducted at the start, midpoint, and end of data collection on five images of the buccal surface and five images of the lingual surface. Overall, the values of the intraclass correlation indicated poor reliability between the three points in time. As such, this led to questions on the identification of bone resorption overtime. A second intraobserver error test using only the second and third observations for each individual and surface displayed vastly different results. A majority of the intraclass correlation coefficient scores were greater than 0.90, indicating excellent reliability. By conducting an error test on the second and third observations, it was clear that the first observation was causing the level of agreement to decrease. As such, the disagreement between the first and last observations of the intraobserver test is a representation of how interpretations change over time and with familiarity.

Interobserver Error Test

The results of the interobserver error test found greater reliability between the researcher and Dr. Schuh, however, reliability decreased when incorporating the results of the undergraduate student. The second observation in the intraobserver error test was the same observation used for the interobserver error test with Dr. Schuh, an experienced observer and the undergraduate student, an inexperienced observer. This showed an increase in the ability to identify bone resorption as we compare the results between the inexperienced and experienced observers. Three problems were identified that led to the differentiation and difficulties in

identifying bone resorption. The first was interpretation. As expected, the identification of bone resorption, due to human error, is subjective to the observer. When reviewing the results of the error test, differences between observers occurred primarily in areas adjacent to bubbles, caused by epoxy when casting, and abraded surfaces. For example, Figure 14 is an image of bone resorption with epoxy bubbles. The large bubble, outlined in red was a specific area of disagreement. My selection of bone resorption is in yellow, and Dr. Schuh's is in green. In our discussion, Dr. Schuh acknowledged that she was more apt to include larger bubbles if they were immediately surrounded by bone resorption. Conversely, I hesitate to include such bubbles as this data is unknown. Another area of disagreement included small areas of bone formation located within larger areas of bone resorption (i.e., islands of formation). If an observer were aiming for speed and attempting to get a general idea of the amount of resorption, these areas of bone formation may be quantified as well, if small and adjacent to the areas of resorption. Alternatively, this area could be excluded, but data collection would take longer. It can be argued that the areas of formation within resorption are often too small to make any significant difference, or that they are areas that will soon be resorbed. However, it depends on the goals of the researcher. While these were the two main issues with interpretation, there were also images where Dr. Schuh and I admitted missing or overestimating the amount of bone resorption.

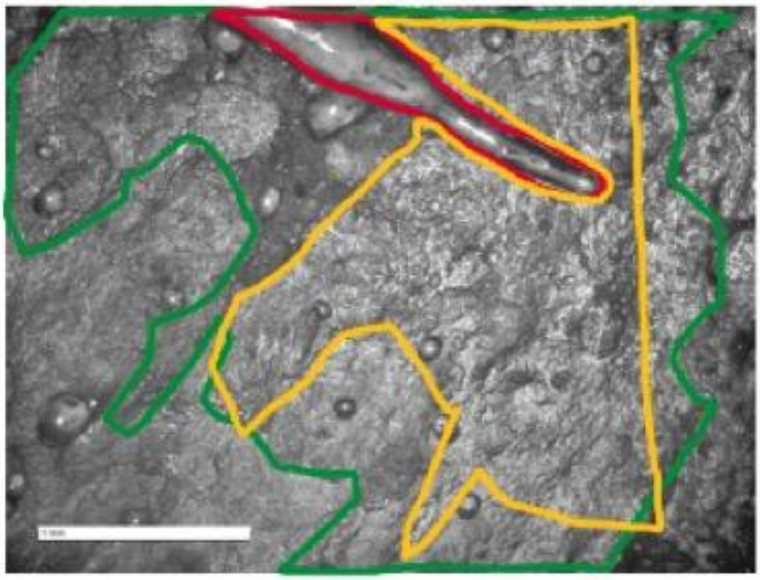


Figure 14. Sub-square showing differences in interpretation. Dr. Schuh's measurement in green, my measurement in yellow, and epoxy bubble in red.

The next problem we ran into was abrasion (Figure 14). Some individuals had bone surfaces containing very obvious areas of bone resorption alongside areas that were partially obscured by some type of abrasion or damage to the bone surface. This abrasion can result from taphonomy, replication, and residue left behind on the bone surface (Bromage, 1984). The silicone mold attaches to any substance directly on the bone, like bone dust and dirt.

The final issue to be addressed was that of image quality. When looking at specimens under the digital microscope, bone modeling was often clear and easy to see. The resulting images commonly resembled what was viewed on the screen. However, there were instances where, upon imaging, the resulting captured image was dark, blurry, and overall unclear. If one were to attempt to retake the image with the same or adjusted settings, the same problem would occur. It is unknown why this problem was occurring. It was chalked up to being a fallacy of technology. Images were taken with the expectation that the sub-square was noted for having resorption, though unclear.

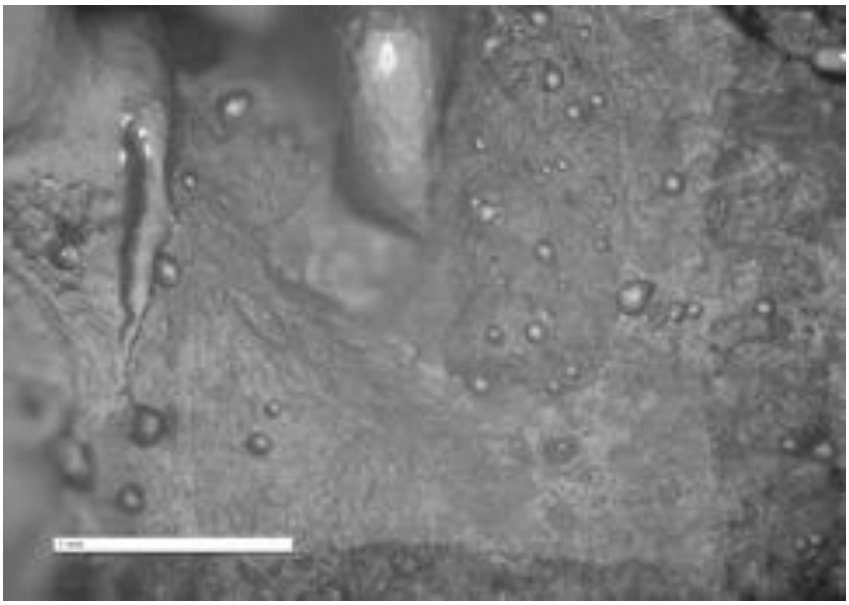


Figure 15 Image displaying poor quality and abrasion.

Before completing the statistics for the interobserver test between Dr. Schuh, the undergraduate, and myself, a trend was noticed for sub squares that Dr. Schuh and I considered completely resorptive. For the same images that we considered completely resorptive, the undergraduate reported the same areas as 0% bone resorption. Upon review of these images, it is likely the result of the three problems listed above. Figure 15 is a juvenile, South African individual. The image quality is poor and there is also a degree of abrasion. While Dr. Schuh and I interpreted this image to be completely resorptive, the undergraduate, just learning to identify bone modeling patterns, was hesitant to make this claim. This was a trend that carried throughout the rest of the data collection steps for the undergraduate, though to a lesser degree. In the images where resorption was less obvious, the undergraduate's results displayed much lower amounts of resorption compared to Dr. Schuh and me.

The results of both the interobserver and intraobserver error tests demonstrate how familiarity viewing bone modeling patterns is vital before beginning the process of data

collection. Though subjectivity may continue to appear, the gap in disagreement closes with increased experience. It is important to take notes on images that may have been difficult to capture or show limited data. Identification of bone modeling patterns becomes easier the more it is practiced. It is recommended for researchers to gain experience and training in identifying and quantifying the patterns prior to recording data. Additionally, it is necessary for the researcher to clarify the goals of their study and determine how important the researcher considers the smallest amounts of data.

Hypotheses

The primary purpose of this study was to answer the question: is there a shared bone modeling pattern in the mandible in a diverse sample of *H. sapiens*? This research evaluated visual and statistical differences throughout ontogeny, between populations, and between biological sexes. Results were evaluated for their adherence to three hypotheses and compared to the results of other studies on the mandible. Differences in bone modeling pattern may be indicative of culturally specific activities, diet, population history, and/or environmental factors.

1.H_o*: Bone modeling patterns are the same within a population from infancy to adulthood.

The expectation for the first hypothesis is that there are no visual differences in bone modeling patterns and no significant difference in the results of the PERMANOVA within a population throughout ontogeny. This hypothesis was only rejected statistically for the Western European population; however, the adjusted p-value using the Bonferroni correction for multiple comparisons was not statistically significant. This correction lowers the alpha value to account for the number of comparisons being performed. There were no significant differences between the age groups for the Greenlandic Inuit and South African.

The adult (AG4) Western European bone modeling pattern is statistically significantly different from the other age groups when evaluating the non-adjusted p-value. This may be related to the composition of this sample, the Western European sample included some of the youngest individuals in the study, and unlike the other groups, the juvenile and adults in the Western European sample were from different countries, France and Germany, respectively. Visually, the adult (AG4) Western European displays less bone resorption compared to the other age groups, which may account for the statistical difference that is seen. However, these results need to be interpreted with caution give the overall small sample size.

For each population, visual patterns for each population appeared similar during ontogeny (AG1, AG2, and AG3), but different for adults. Muscle attachment sites become increasingly rugose as more stress and wear is placed upon them. Rugosity on bone is an indicator of strengthened muscle. Muscle attachment sites on bone typically experience a higher amount of bone turnover as bone adapts to the stresses placed upon it. The resulting turnover influences bone modeling. Musculature changes as dentition develops in the mandible, also affecting the bone modeling patterns that may be seen (Coquerelle et al., 2013). As these are processes that influence all individuals, this may be why we see a similar pattern visually during ontogeny for each population.

Bone resorption is seen on the mental symphysis ramus for AG1, AG2, and AG3 for all populations. The mental symphysis fuses during the first postnatal year (Schaefer & Scheuer, 2009). In this study, increased resorption is seen at the mandibular symphysis as the suture begins to close and the symphysis fuses. A decrease in the amount of bone resorption in the region appears from AG2 onward for the buccal surface. The development of the chin occurs between 1 and 5.5 years, corresponding with AG1 (Coquerelle et al., 2013b). The chin develops

as the result of the relationship between bone formation and forward movement of the mental symphysis (Enlow, 1966; Pampush & Daegling, 2016). The results of this study show resorption along the mental symphysis in AG1, with variability in bone resorption in the area of the chin. The variability seen is likely related to the development of the chin. By AG3, the chin is developed, and this region displays more bone formation.

AG2 and AG3 in each population have fairly distributed amounts of resorption but much of it is on the anterior and posterior areas of the ramus. For the anterior, the masseter muscle attachment site makes up most of the area of the mandible. The masseter muscle is heavily involved in the mastication process. On the posterior ramus, the resorptive areas are around the pterygoid medialis, another masticatory muscle attachment site. As diet changes around the time of weaning, we may see increased bone resorption in the areas of masticatory muscles as increased stress is placed on those areas of bone (Helm & German, 1996). AG1 consists of individuals younger than six years old who retain deciduous dentition (AlQahtani, Hector, & Liversidge, 2010). Though weaning ages may be different for the groups in this sample, weaning generally occurs between the ages of one and three (Sellen & Smay, 2001). There is an increase in total bone resorption between AG1 and AG2 for all populations, possibly related to the more regular eating of solid foods in the juvenile diet.

Mandibular Growth and Comparison to Other Studies

Kurihara and co-authors' (1980) research studied maxillary and mandibular bone modeling patterns for 36 individuals during ontogeny, corresponding to AG1, AG2, and AG3 of this study. The authors found resorption in the alveolar region directly above the chin. The mandible had deposition on the chin, or mental prominence. The results of this study show

similarities to Kurihara et al., where there is bone resorption along the mental symphysis. This study, however, is more variable in the amount of bone resorption on the chin.

A study by Martinez-Maza and co-authors in 2013 compared the bone modeling patterns of human juveniles and adults in the face. The specimens came from the Anthropological Collection of the University of Coimbra (Portugal), a Western European sample. The sample was composed of 12 individuals of known age and sex, though sex differences were not researched. This is the one of the few studies to compare bone modeling patterns of both the lingual and buccal surfaces of the mandible. As in this study, Martinez-Maza and co-authors see a pattern in the adults that can be distinguished from the juveniles, suggesting that cellular changes occur with the development of the craniofacial system. They also reported that juveniles have resorption on the ramus for both the lingual and buccal surfaces. The adult pattern for Martinez-Maza et al. (2013) is also similar to the adult bone modeling pattern in this study, displaying resorption that approaches the condylar neck on the buccal surface and resorption in the area of the buccinator attachment site on the lingual surface. This study's mandibular results consisted of an overall younger ontogenetic age, which may be attributed to the differences seen on the buccal surfaces.

2.H_o*:* Bone modeling patterns are the same between populations.

Based on the hypothesis, it was expected there would be no significant differences in the digital bone modeling maps and PERMANOVA between populations and ages groups, and the bone modeling maps would not be different visually. The hypothesis was rejected statistically as there are statistically significant differences and visual differences between the Western European and Greenlandic Inuit, although the adjusted p-value is not significant. This suggests that there are differences between populations, although subtle.

Diet and Cultural Practices

Numerous studies have shown the relationship between diet and masticatory stresses for a variety of species including humans, nonhuman primates, and rats (Anderson et al., 2014; Yamada & Kimmel, 1991; Hylander, 1977; Churchill & Morris, 1998). The mandible has been shown to better reflect diet than population history (von Cramon-Taubadel, 2011), and that mandibular shapes more closely resemble one another in groups with similar subsistence strategies (Hoover & Williams, 2016; Sardi et al., 2006). Masticatory stresses in conjunction with cultural activities may be the cause of this difference. Mandibular and overall craniofacial size have been found to decrease in agriculturalists (von Cramon-Taubadel, 2011).

There are three distinct subsistence strategies that the groups in this study engages in. The Western Europeans are postindustrial agriculturalists while the Greenlandic Inuit are primarily hunter-gatherer-fishers, and the South African are terrestrial hunter-gatherers (Hylander, 1977; Churchill & Morris, 1998). An agrarian diet consists of softer foods such as grains, vegetables, and fruits, supplemented by meat from animals (Milton, 2000). Hunter-gatherers eat more hard foods such as tough meats and raw plants, roots, and tubers (Iannotti et al., 2022). The fisher diet consists of aquatic mammals, shellfish, and is supplemented by terrestrial plants.

The periosteal surface has been shown to be less responsive to diets that are less harsh on the bone and surrounding musculature (Yamada & Kimmel, 1991). High mechanical loading from mastication can be why we see differences in adults between the hunter-gatherers and agriculturalists. In this study, the agriculturalist diet is low stress, reporting formation in the areas of musculature. Conversely, the hunter-gatherer diet is higher stress, and in the same muscular regions, report resorption. These findings are consistent with other work conducted on the maxilla (Brachetta-Aporta et al., 2019).

We can see on the bone modeling maps that the South African and Greenlandic Inuit populations differ from Western European in the areas of masticatory muscles, although statistically, differences are only seen between the Western European and Greenlandic Inuit (Figures 10 and 11, Table 13). The Greenlandic Inuit population are known for their extensive use of teeth as tools (Hylander, 1977; Scott. 2020). In this study, they exhibit more resorption in the incisor region, although it is surprising that the pattern is not more resorptive along the ramus and in areas of masticatory muscle attachment sites. The incisor teeth act as a third hand, holding bow drill bits, as well as being used to shape wood for hunting purposes and prepare seal hides (Hylander, 1977; Pederson, 1947). The buccal surface of the mandible for the adult (AG4) Greenlandic Inuit population displays a higher percentage of resorption directly beneath the incisors. Though their overall bone modeling pattern is similar to the South African population, the differences in the incisal region may be due to culture-specific activities.

3. H_0*:* Bone modeling patterns are the same between the individuals of known sex for the Western Europeans (the only group in the sample with known sex).

One aspect of this study was to compare bone modeling patterns between biological males and females to determine if it could be an indicator of sex, as biological sex in juvenile skeletal remains continues to be an ongoing problem in biological anthropology. Based on the third hypothesis, it was expected that there would be no visual differences between males and females of the Western European sample. This hypothesis was rejected due to visual differences on the digital bone modeling maps between males and females. Age groups were pooled for each sex and analyzed visually. There was no visual difference between males and females for the buccal surface of the mandible. Differences in the pattern of bone resorption appeared on the lingual surface of the mandible between the sexes. Males had a greater amount of bone resorption along the ramus, while females were characterized by formation in this area. A

possible explanation for the differences seen in the maps with ages pooled is the presence of sex hormones (Baab et al., 2018). Sex hormones such as testosterone and other androgens have been used to explain increased bone deposition in males at the onset of puberty. Alternatively, estrogen stimulates bone formation by increasing the number of osteoblasts while simultaneously decreasing bone resorption (Kini & Nandeesh, 2012). Estrogen levels rise during puberty and have an active role in the menstrual cycle. Thus, with an increased amount of estrogen, females exhibit higher levels of bone formation.

Due to a small number of individuals and uneven distribution of the sexes in the later age groups, additional maps were created for only AG1 and AG2. These maps were created to determine how early a sex-specific bone modeling pattern could be identified in ontogeny. The lingual and buccal surfaces resemble one another for males and females in AG2, whereas AG1 displayed greater visual differences (Figure 11). These differences could be indicative of sex differences at an early age. However, these mean maps were created from only two individuals, and statistics were not performed because of small sample sizes. Thus, these results should be interpreted with caution, as the variation that is seen in these bone modeling maps could be due to a single individual significantly impacting the mean total percentage of bone resorption.

Mandibular growth in the context of craniofacial growth

The craniofacial system grows with an increase in brain size, influencing both the size and shape of the face. The upper face holds the oro-naso-pharyngeal cavities and while the mandible is not directly related to these cavities and the airway, mandibular growth is transitively related to these airways through the maxilla and possibly the integration of the airway and lungs. The pattern seen during ontogeny (AG1-3) for each population corresponds with Enlow & Bang's (1965) Principle of the V and the fusion of the mental symphysis. The

ramus, at the end of the "V" is elongating and growing at the widest point of the "V" shaped bone, largely due to the growth of the maxilla.

To get a more comprehensive understanding of human facial growth, it is interesting to compare the results of this study with Schuh et al. (2020), which evaluated the ontogenetic bone modeling patterns in the maxilla for the same individuals, allowing for direct comparisons between anatomical regions. Schuh and co-authors' research found no significant difference between populations and age groups. Schuh and co-authors found maxillary bone modeling patterns to be similar from AG1 to AG3 within a population, and by adulthood, AG4, there was an overall decrease in the mean total percentage of bone resorption and subtle population differences emerge.

A similar pattern is seen in this study on the mandible. Although there are slight differences in the bone modeling patterns in AG1, populations retain a similar pattern in AG2 and AG3 on the mandible. On the mandible at AG4, each population has the lowest mean percent of bone resorption. The visual bone modeling patterns of AG4, although similar for the Greenlandic Inuit and South African populations, are unique in certain areas and differ from the age groups that proceed them, just as in Schuh and co-author's study. Similar to findings on the maxilla, the greatest amount of bone resorption for the adult Greenlandic Inuit mandible was in the incisor region, supporting the idea that differences in bone modeling pattern could be related to differing cultural practices.

The main difference between Schuh and co-author's study on the maxilla and this study is the amount of bone resorption. The maxilla is much more resorptive than the mandible. High levels of resorption in the modern human maxilla have been related to our uniquely flat, or orthognathic faces (Bromage, 1989; McCollum, 2008; Martinez-Maza et al., 2010, Martinez-

Maza et al., 2015; Schuh et al., 2021). Other hominins, with more prognathic faces, have higher levels of deposition in the maxilla (Enlow, 1966; Bromage, 1989; Schuh et al., 2021). Resorption has been associated with the posterior displacement of bone during ontogeny and deposition is associated with anterior displacement (Enlow, 1966). The mandible has higher levels of resorption along the ramus, which shifts posteriorly during ontogeny (Enlow & Moyers, 1971). The mandible, as a whole unit, displaces in an anterior direction to match overall maxillary growth. This may be why we see less resorption on the mandibular body.

The mandible is the last bone in the facial skeleton to finish growing, and its development is dependent on the rest of the craniofacial system as it articulates with the cranial base and aligns with the maxilla (Enlow & Bang, 1965; Stansfield, Evteev, & O'Higgins, 2018). Since the maxilla develops prior to the mandible, it undergoes more bony changes due to its integration with the remainder of the face. This may be why we see greater amounts of bone resorption in the maxilla The mandible adjusts to the maxilla. However, the mandible has a loose integration with the skull as it only needs to adapt to one bone and surrounding musculature, leading to lower amounts of bone resorption.

Schuh and co-authors (2019, 2020) suggest that slight differences of bone modeling patterns in the maxilla are not due to difference in the location of bone resorption, but the result of rate changes of bone resorption and formation. Together these studies suggest that growth of the maxilla is constrained because of the integration of the skull. The mandible is less constrained as it only articulates with the temporal bone at the joint and grows in conjunction with the maxilla due to dentition and soft tissue.

Facial bone modeling in hominins

To better understand the evolution of *H. sapiens* facial morphology, the results of this study can be compared to published data for fossil hominins, the most relevant being our most recent ancestors *H. heidelbergensis* and *H. neanderthalensis*. Compared to *H. heidelbergensis* and *H. neanderthalensis*, *H. sapiens* mandibular morphology is distinct. The human chin is a key characteristic of modern humans (Pampush & Daegling, 2016). The development of the chin has been argued to be the result of several factors including growth and spatial restraints, functional stress, and the development of anterior dentition (Coquerelle et al., 2013a; Holton et al., 2015; Ichim et al., 2006).

Rosas & Martinez-Maza (2010) studied the bone modeling pattern in the mandibles of *H. heidelbergensis*. According to the published bone modeling maps, juvenile *H. heidelbergensis* specimen AT-3888 and the *H. sapiens* in this study exhibit resorption on the ramus, in the areas of muscle attachments sites on the buccal and lingual surfaces. On the juvenile AT-3888, there is formation on the buccal surface in the region of the mental symphysis whereas *H. sapiens* displayed bone resorption. The chin develops in the same space as the mental symphysis and the lack of resorption in this area for *H. heidelbergensis* could be indicative of a species-specific bone modeling pattern related to the lack of a chin.

Published bone modeling maps on Neanderthal mandibles show that there are few similarities between them and *H. sapiens* (Martinez-Maza et al., 2011). The buccal surface on Neanderthals demonstrate formation in the areas of the ramus on the buccal side and formation around the mental symphysis. The modern human bone modeling pattern directly opposes that of Neanderthals with resorption along the ramus and region of the mental symphysis. The lingual surface of Neanderthals and the humans in this study is more similar. Differences in bone modeling patterns of the maxilla between Neanderthals and *H. sapiens* have also been identified

(Lacruz et al., 2015). Together, these studies suggest species-specific patterns. However, comparisons have only been made between fossil hominins and modern humans from the Holocene epoch. More robust *Homo sapiens* of the Late Stone Age or Upper Paleolithic would be a more appropriate comparison to the fossil hominins discussed above, however, this relationship has not been studied. Poor preservation of fossil hominins makes the identification and comparison of bone modeling difficult.

Challenges and Limitations

The biggest challenge in this research is that of time. Many aspects of the research needed to be repeated or replicated due to problems in data collection. As previously mentioned, there were challenges not only in the identification of bone resorption, but in the use of the digital microscope. Multiple images had to be retaken for each individual because the images were dark or unclear. It is unknown why this problem occurred, but it was noticed that this issue was commonly seen when imaging areas with holes and ridges, likely an issue with the depth of the bone. With increased time, sample sizes can be expanded. The sample size for this study is small and statistical analyses should not be overinterpreted.

Another major difficulty of this research was the projection of the digital bone modeling maps onto the three-dimensional surface models. Originally, the plan was to project the results using the software Geomagic (Research Triangle Park, NC). However due to the irregular shape of the mandible, technological difficulties occurred. It was difficult to project the vertical ramus while simultaneously projecting the horizontal body. Attempts were made to subdivide the maps into their vertical and horizontal parts separately, but the program would not allow for an alignment of the two aspects of the map. Due to this issue, the maps were projected using Photoshop and Illustrator (Adobe Inc., 2019). It was much easier to project the maps onto an

image of the three-dimensional surface model, though in this case, square size could be warped, and it was difficult to accurately align the map with the model.

Perhaps the most difficult aspect of this research was that of supply chain issues. Due to residual problems from the COVID-19 pandemic, there was trouble in obtaining materials needed for data collection. For the purposes of this research, epoxy was needed to create the positive replicas of bone. Many casts had been previously made using one brand of epoxy (5 Minute Epoxy Epoxidharz 2 L-Kleber transparent, Devcon). However, this brand took several months to arrive as it had previously been out of stock or unavailable. Numerous other brands of epoxy had to be trialed. While the best alternative was determined to be the J-B Weld Ultrarez UV-Resistant Coating & Casting Epoxy, there were other types of epoxies that were also tested for image clarity and their ability to retain data.

Conclusion and Future Studies

This study analyzed the surface of the human mandible for evidence of a shared bone modeling pattern that can be identified for *H. sapiens*. Using replicas of bone surfaces, bone resorption was identified and quantified through digital microscopy. The data was computed into digital mean bone modeling maps for each individual, as well as mean maps created for age, population, and sex. Bone modeling patterns are similar between age groups and populations throughout ontogeny until adulthood, where we can see the obvious changes in bone modeling patterns for all three populations. Visual bone modeling pattern differences are seen between populations with differing subsistence practices, where hunter-gatherer populations are more similar to one another than they are to agricultural populations. The integration of other populations with diverse diets would aid in determining the validity of the dietary signal seen in this study.

More research needs to be conducted to determine whether bone modeling patterns may be used to determine the biological sex of skeletal juveniles. This study looked at mean bone modeling maps for the youngest individuals in the study (i.e., AG1 and AG2) for biological sex as there two males and two females in each age group. The preliminary results suggest greater sex differences for males and females in the youngest age group, AG1, and that the patterns are more similar in the subsequent age group, AG2. Mean maps by sex were also created with age groups pooled. Males displayed more bone resorption compared to females. By including more individuals of known sex and an understanding of the effect of sex hormones on bone growth, we may be able to interpret sex differences in bone modeling patterns (Baab et al., 2018; Kini & Nandeesh, 2012).

This research, along with the work of previous research, used bone modeling data from the maxilla and mandible from three geographically diverse populations to look at bone modeling patterns. However, to fully understand bone modeling patterns in the face, more data from other anatomical regions needs to be integrated. Currently, studies on the zygomatic and brow ridge are underway using the same methodology and individuals. With additional research on these facial components, we can compare this data to fossil hominins to better identify species-specific differences and adaptations. We can also use geometric morphometric shape analysis and additional statistical analyses, to understand the relationship between bone modeling patterns and shape differences and disentangle population history from dietary signal.

This is the first bone modeling study to implement an error test. Future work on bone modeling patterns should include an error test early in the research process to accustom researchers to the identification of bone resorption and formation. Additionally, due to the time-consuming nature of this research, the development of an automated approach would increase the

speed and accuracy. This type of machine learning would allow for cellular activity (i.e., bone resorption and formation) to be both identified and quantified, potentially allowing for less subjectivity.

APPENDIX A: PROCEDURE FOR DATA COLLECTION

Walls

1. Preparation

To begin creating walls, you will need to have the following materials:

- Silicone cartridge or alternative molding material
- Silicone gun
- Box of gun tips
- Gloves
- Tray
- Plastic covering (bubble or saran wrap)
- Logbook

Put gloves and lay the plastic covering on the counter. You can place the tray overtop the plastic covering as this is the surface you will be working on. Take the silicone gun and pull the black slider all the way out and insert the cartridge into its slot, then push the slider back in. Unscrew the cap from the cartridge and replace it with one of the plastic tips. Alternative molding materials, such as the Pixiss, do not require the silicone cartridge and instead can be made by mixing equal portions of Part A and Part B until a uniform color is reached.

2. Creating Walls

The silicone cartridge gun is recommended for creating your initial base and wall, with alternatives used for filling in gaps and holes. To create a wall, you will simply push down on the trigger of the silicone gun and begin dispensing it onto the tray, creating a flat, even surface for your base. The silicone dries quicky so you will need to have your individual selected and placed with the data side up on the base. From here, you can begin building up walls around the cast on the edges of the base you have made. These walls need to be high enough that the epoxy will not leak out. After your silicone has dried, check for any holes that need to be covered. This can be done with the silicone gun or the alternative molding material. The alternative material has a longer work time and will allow you time to fill the gaps completely. Once materials are dry, you can place the mold back into its proper bag, until time for casting. The silicone will dry in the tips rather quickly. Change the tips and cartridges as needed. Be sure to keep a running list of the individual and bone you created walls for.

3. Cleanup

Dispose of your gloves and all unusable plastic. Cap your silicone guns and throw away all tips that have dried up. Return all materials to where you found them. Do not forget to write your information in the logbook. Place the bags with your completed individual in its respective box and make sure the correct individual goes into the correct bag if working with multiple at one time. The trays can be reused, try your best to scrape off all dried silicone into the trash.

Casting

1. Safety

Before working with the materials for the epoxy, make sure you are wearing the proper safety equipment. This includes gloves, face covering, long sleeves, and pants. For extra precaution, wear a lab coat. It is important to avoid direct contact of the epoxy and skin. Wash the area with soap and water immediately if contact is made.

2. Making the Resin

To make the resin you will need:

- Plastic covering
- Plastic stirring stick
- Two plastic beakers marked “A” and “B”
- Plastic syringe
- Logbook
- Epo-Tek 301 (Epoxy Technology) bottles “A” and B” (preferred)

The Epo-Tek 301 is recommend, but other alternatives can work such, as the J-B Weld UltraRez. It is important the epoxy dries clear. Lay out the plastic wrap on the tray and counter before beginning. This will make the cleanup process easier and prevent epoxy from ending up the counter. The proportion of Part A to Part B is 3:1. It is recommended that you work with a small amount at first so as not to waste. For example, you will pour 30ml of Part A into plastic beaker “A” and pour 10ml of Part B into plastic beaker “B”. Pour Part B into Part A. Use the plastic stirring stick to mix the materials. Stir in a clockwise motion for 3-5 minutes, then pour the mixture back into beaker B. Mix counterclockwise for another 1-2 minutes. Then let the solution sit for another 2 minutes. Epo-Tek instructions list a shorter mixing time, but we found this to be best.

3. Creating the Cast

After you have mixed your materials, you have two options. You can either use the syringe to draw the epoxy from the beak OR pour directly from the beaker. Remember, the epoxy is being used to create casts of materials that already have walls. Be sure to look at your materials before adding epoxy to avoid holes in the silicone or walls that are too short or uneven. Holes and other wall issues will cause the epoxy to leak out and be unusable. As you pour the epoxy onto the mold, you want a thin layer with all parts of the mold covered. An incomplete epoxy pour leads to missing data and another cast will need to be made. It is best to leave the tray with your casts in a fume hood to fully harden. Please keep track of which individual goes in which bag.

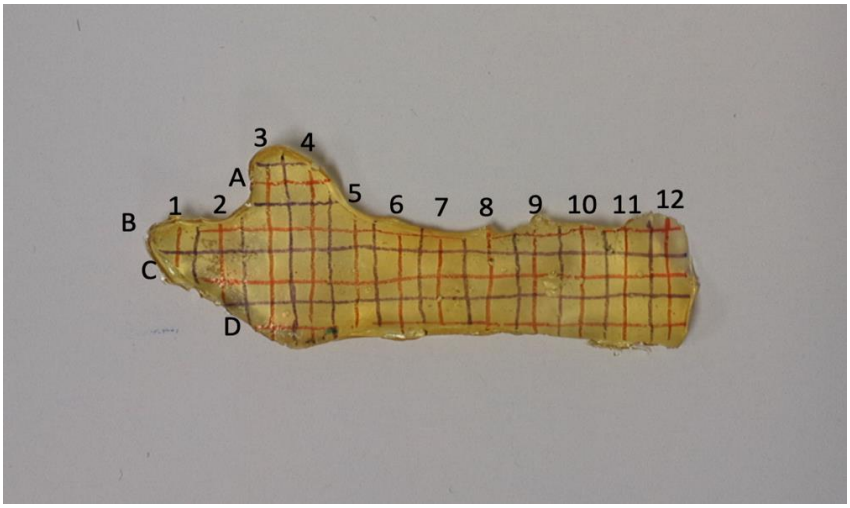
4. Cleanup

After handling the epoxy, be sure to wash your hands. Make sure your beakers are clean and if the syringe is no longer useable, throw it away. Plastic beakers can usually be reused. The epoxy resin, while wet can be cleaned up with isopropyl alcohol. If the plastic wrap cannot be used again, throw it away. Dispose of gloves and face coverings and return casting materials where you found. Mark in your lab book which individuals were casted.

Microscope

1. **Prepare your sample**

On your DRY cast, draw a grid. Place the cast on grid paper, use double stick tape if it slides too much, and follow the lines to draw a grid on your cast. The resulting squares should be 5x5mm. After all squares are drawn, sub-divide your squares into 2.5x2.5mm sub-squares. See below:



2. **Starting with the microscope**

A. Turn on the computer and create a new folder. Name it the specimen name:

Documents-> Lab Data-> SpecimenName_Side_Bone (e.g., KAL_0707_LZyg)

B. Open the software first: Smartzoom 5. Then, click on "System Administrator". Now turn on the microscope and wait until the connection is made between the microscope and software. Once it is connected, select "Free Examination".

3. **Create Overview Image**

A. Place the white piece of cardboard under the microscope and position the sample. This will prevent reflection.
B. Go to "Setup" and take an initial picture of the mold using "Acquire Overview Image". Do not use "Overview Illumination". Then, give the job a name. This should be your specimen's name, SpecimenName_Side_Bone (e.g., KAL_0707_LZyg), and click "Apply".
C. Go to "Edit" and press "Acquire".
D. Go to "Result" and the overview image will be in the upper right corner. Export this image by pressing the middle button with the three horizontal lines and select export

image. Save it to the folder you created and name it “Overview” (e.g., KAL_0707_LZyg_Overview).

E. Copy the Overview to a USB flash drive and connect it to the workstation to print.
F. Paste the image in your logbook and label your grid.
    - Label alphabetically starting at the top row and list vertically until you run out of bone. Label numerically starting from the furthest left column for the buccal surface of bone and label numerically starting from the furthest right column for the lingual surface. Sub-squares should be labeled as such:

1	**2**
3	**4**

Buccal Surface

2	**1**
4	**3**

Lingual Surface

You will mark what you see in the sub-squares, resorption, formation, or NA. You may use whatever system you prefer, just make sure it is noted and consistent. See the above image for assistance. If you are unable to print your image, OneNote will allow you to mark your images digitally.

4. Data Acquisition

After taking an overview image, you will move on to the “Edit” tab. From here, you have to adjust the parameters to make your observations better suitable for data collection:

A. Stage/Tilt: The slider for “Stage” will need dragged all the way up while the “Focus” slider will need dragged down. This will bring the stage up closer to the lens and bring the view into focus. If you look in the top right corner, you will see the numbers for the focus change. The focus is commonly in the 100-115. Once you get close, you can use the dials on the pad to adjust and get a clearer image.
B. You can choose your starting square and zoom in and out with the mouse. Zoom out with the middle mouse button and choose your starting square by double clicking on the overview image.
C. Focus image. To focus, the small, top knob is to zoom in and out and the bottom knob is to focus.
D. Lighting/Aperture: In “Illumination”, change the lighting to “Coaxial bright”. You can adjust the brightness depending on the lighting of the room, but it is unlikely you will need to do so.
E. Image Enhancement: click on “Real Time HDR” (the red dot should turn green once it is selected and on) and reduce to “Color Saturation” to 0. Then adjust the “Texture” as much as you want for a clearer picture. This will increase the contrast. We commonly only move it until it is just above 0. Sometimes if the texture is too high, the images do not come out good. It is better to reduce to texture to 0, even if the frame looks blurry to start. A higher contrast is typically better to really see bone resorption. You may need to adjust the “Brightness” too.

5. Taking an Image

A. Select “Image Processing” to take an image of your current field of view. Using the knob, you adjust the zoom as well as the focus. The zoom needs to be at 101x magnification with the 5x objective lens. You can double check this in the upper right corner.
B. Click on “EDF” (the red dot should turn green once it is selected and on). Adjust the focal plane (larger, bottom knob) to the lowest and highest points of your field of view before taking the image. These points will be just barely in focus. Press the “Assign” button to identify the lowest and highest planes.
C. Press “Acquire” to capture the image. It is important not to touch the table or microscope at this stage. If you need to retake the image, select “Live” and repeat B.
D. Before saving the image, make sure to put a scale in “mm” on each picture you take. This will be on the top of the screen, usually next to “Last used:”. The line color and text color will need to be adjusted. Double check the rest are at the proper settings:

Line Color: White
Text Color: Black
Line Thickness: 5
Font Size: 14
Label and color opacity: 100

E. Saving image: Select middle button on the image (the three horizontal lines) and export as a TIFF or .tif. This should be default. Save the file in the specimen folder specifying the square and sub-square:

SpecimenName_Side_Bone _Square_Subsquare (e.g., KAL_0707_LZyg_A5_1)

APPENDIX B PROTOCOL FOR THE QUANTIFICATION OF BONE RESORPTION

You will need the following programs downloaded to your device: R, RStudio, and ImageJ. Additionally, make sure all images for a specimen are in their correct and individualized folders.

Step 1: Prepare Excel Files

Create an Excel file for each individual. You will need a list of all the images you have taken for a specimen. To load all the names of the pictures, apply the "List_pictures" R code to your specimen's folder in RStudio. You will get a data frame that you can copy and paste in the column "1/4 square". Remove your overview image from this list.

Step 2: Sharpening

Use the ImageJ script "Sharpen" to sharpen all pictures. First, create a file where the sharpened pictures will go and indicate the new path in the code. Then go to File -> Open and open the "Sharpen" script in the console. Select the whole script and click "Run". Normally it will open each picture one by one in ImageJ*.

*We had difficulties doing using these codes on both Mac and Windows.

Step 3: Calibration

Open your first sharpened picture, and select the tool called "straight" (the straight line). Then draw a straight line on your scale bar, go to Analyze -> Set scale, and write in "Known distance: 1" and "Unit of length: mm". Report ***all*** the information in the "Calibration_1" script for ImageJ (distance/known/pixel/unit). This is the one you will use each time you open a new individual.

Tip: to see if your calibration worked, try it on the scale bar again, to make sure it gives you the right amount.

*If this does not work, you can try setting the calibration through Analyze-> Set Scale and once you give known distance and length, select global. This should allow for you to keep the same scale. However, you will get a pop-up. Do NOT allow for the disabling of the Global Calibration or you will have to repeat steps. I recommend you leave one image open, and the pop-up won't appear every time you select a new image.

Step 4: Selecting and measuring bone resorption

To start selecting bone resorption, use the tool "Polygon selections". Click one time at the beginning of the area you want to select, and then continue clicking and forming the shape you want. Join the first and last points to close the selection. Then press CTRL+M to do the first measurement. A window called "Results" should open, showing different things. Report the "Area" in the Excel file (column "Measurement_1 (BR ImageJ)"). To open the next pic, press CTRL+Shift+O.

* CTRL+Shift+O may not work. You can click and drag images into the ImageJ window

Step 5: Calibration 2

Once you took all your measurements on the pictures, open the picture of the Overview and change the calibration. Do all the steps as in Step 3 but calibrate it to "Known distance = 5". Similarly, report all this new info in the Calibration_2 script. Next time you open an "Overview" picture, just run the script on this picture.

*Again, if this does not work, you can try setting the calibration through Analyze-> Set Scale and once you give known distance and length, select global. This should allow for you to keep the same scale. However, you will get a pop-up. Do NOT allow for the disabling of the Global Calibration or you will have to repeat steps. I recommend you leave one image open, and the pop-up won't appear every time you select a new image.

Step 6: Getting the percentages of BR for each square

First, write all the names of each square that you wish to digitize in the "Mapping" column. Not all of them need to be reported. This is your preference. If you think they are too small, you don't necessarily have to include them. I included everything to be safe.

In the squares that just have bone formation, write "0" in "Percentage". For all the squares that have BR:

- "Complete" squares (= those who are a full square of resorption):
    - Write "**100**" if the square has only BR.
    - Write: **=(%BR Square*100)/25** if it has both formation and resorption
        - It is easiest if you use the formula =SUM(**select the sub squares**)/25*100

- "Incomplete" squares (those that you decide to keep but are not 25 mm^2; mostly found around the borders of the mold): go to your overview and measure with the polygon tool the area of that square (called here "Area"). Then do: **=(%BR Square * 100)/ Area**

  This allows you to get the percentage of incomplete squares so that on the map, they will be considered as "complete" squares (because the mapping does not draw incomplete ones) but with their original percentage of BR for that size square. When finished, select the Mapping and Percentage columns (with the names) and paste them in a txt file that you will save in a folder (see example) for each individual*. Save the "Results" document from ImageJ in a folder with the Excel file.

*When I did it, I saved everything to one Excel file so it was in one place. Then I created separate txt files and saved them to a separate folder.

Step 7: Calculating the %BR for each square and the total BR for each individual

Once all your measurements are reported in the Excel sheet, you need to add all the data of bone resorption and report this sum in the column "Total BR/mm^2" for each square. For example,

if you have “A5_A”, “A5_B” and “A5_C”, you need to do A5_A+A5_B+A5_C in the row for A5. This will calculate the total area of bone resorption for the square A5. Do the same for all the squares that have bone resorption**.** The goal here is to get a count of all the areas of bone resorption on the bone.

Go back to your overview and use the tool “Polygon selection”. Do not forget to change the calibration with Calibration_2 if it is not done. To be sure, select the tool “Rectangle” and draw a rectangle on one of the squares of your grid that is complete. Measure this area and make sure that it is around 25 mm^2.

Once you have reported all the areas of bone resorption for each square, you can calculate the total amount of BR by adding all these data and reporting it in the column “Total BR (mm2)”. A helpful formula:

=SUM(**squares with BR**)

Step 8: Calculate the total area of bone

Go to your overview and with the tool “Polygon selection”, select the contour of the overview to measure its total surface. Report this number in “Total area specimen”. Do it two times and take the mean of the two measurements.

Step 9: Calculate the percentages of BR and NAs

To calculate the total percentage of BR, you just need to write in the column “% BR TOT SPE”:

= (Tot BR*100)/ Tot area specimen

Click on the cells that contain the numbers (do not write them by hand!).

APPENDIX C PROTOCOL FOR CREATING PROJECTIONS

1. **Segment your 3D model**
    a. Open up Geomagic Wrap on your workstation and open the ply file you want to be working with.
    b. Right click the name of your file and select duplicate.
    c. On the right-hand side of your screen, select the “Rectangle” tool and “Select Through” tool.
    d. Select the side of your bone that you are not wanting to use and hit the “Delete” key. You should be left with only one side of the bone.
2. **Take a picture of your 3D model**
    a. At the top left of your screen, right click the “Capture” button, select “Advanced” and click “Capture” again in the pop-up.
    b. Give your duplicate a name and save as a .tif. Select “Ok”
    c. Open the duplicate TIFF in Adobe Photoshop
3. **Prep your digital map for projection**
    a. Open the pdf of your mean map and click “Edit PDF”
    b. Right click and “Select all”
    c. Right click again and click “Edit Using”. Select Adobe Illustrator.
    d. Once Illustrator opens, copy the map you are wanting to use without labels.
    e. Paste into Photoshop as a Smart Object.
4. **Project your digital map**
    a. Now that both your 3D model and map are in Photoshop, you should be able to move it around to fit your needs.
    b. Make sure the map is a “Smart Object”. To double check this, right click the layer that your map is, and select “Convert to Smart Object”. In the Edit tab, you can you the “Transform button” to warp your layer and move it around to fit the bone.
    c. If you need to erase any areas where the map spills over the edge, you can clean it up using the “Eraser” tool. A pop-up will appear asking if you want to rasterize the image. Select “yes”.
    d. Adjust the fill of the layer to 75% and save your new image as a TIFF.

APPENDIX D IRB CORRESPONDENCE

Institutional Review Board
FWA00000351
IRB00001138, IRB00012110
Office of Research
12201 Research Parkway Orlando, FL 32826-3246

UNIVERSITY OF CENTRAL FLORIDA

NOT HUMAN RESEARCH DETERMINATION

February 2, 2023

Dear Madison Hubbart:

On 2/2/2023, the IRB reviewed the following protocol:

Type of Review:	Initial Study
Title of Study:	Bone Modeling Patterns in the Modern Human Mandible
Investigator:	Madison Hubbart
IRB ID:	STUDY00004787
Funding:	None
Grant ID:	None
Documents Reviewed:	• HRP-251- FORM - Faculty Advisor Scientific-Scholarly Review.pdf, Category: Faculty Research Approval; • HRP-250-FORM- Request for NHSR (9).docx, Category: IRB Protocol; • MPI_BoneModelingData_CompleteList (1).xlsx, Category: Other

The IRB determined that the proposed activity is not research involving human subjects as defined by DHHS and FDA regulations.

IRB review and approval by this organization is not required. This determination applies only to the activities described in the IRB submission and does not apply should any changes be made. If changes are made and there are questions about whether these activities are research involving human in which the organization is engaged, please submit a new request to the IRB for a determination. You can create a modification by clicking **Create Modification / CR** within the study.

If you have any questions, please contact the UCF IRB at 407-823-2901 or irb@ucf.edu. Please include your project title and IRB number in all correspondence with this office.

Sincerely,

Jonathan Coker
Designated Reviewer

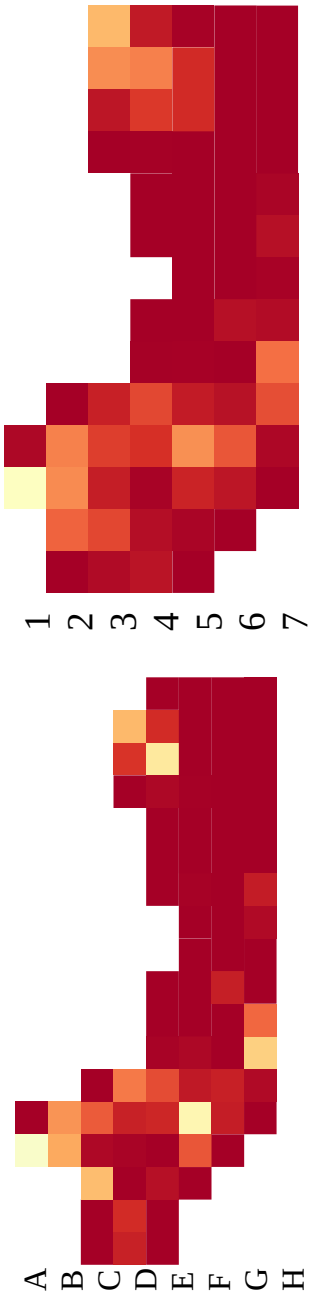
APPENDIX E LIST OF INDIVIDUALS

INDIVIDUAL	POPULATION	DENTAL AGE	ONTOLOGICAL AGE	KNOWN SEX
1879-73-121	Western European	N/A	5	F
1886-87-99	Western European	N/A	6	M
1890-91-21	Western European	N/A	10	M
1892-93-285-179	Western European	N/A	10	F
1892-93-286-180	Western European	N/A	0.833	F
1892-93-307-197	Western European	N/A	0.5	M
1893-94-8-210	Western European	N/A	4.5	F
1893-94-86	Western European	N/A	4	M
1894-95-142-277	Western European	N/A	0.5	M
1898-99-232-476	Western European	N/A	0.83	F
1899-288-512	Western European	N/A	4.5	M
1902-144-594	Western European	N/A	12	M
1906-07-37	Western European	N/A	7	M
KAL-0028	Inuit	M1	N/A	N/A
KAL-0109	Inuit	N/A	adult	N/A
KAL-0122	Inuit	M2	N/A	N/A
KAL-0168	Inuit	dm2	N/A	N/A
KAL-0169	Inuit	dm2	N/A	N/A
KAL-0401	Inuit	M1	N/A	N/A
KAL-0674	Inuit	M1/M2 in eruption	N/A	N/A
KAL-0707/08	Inuit	dm2/	N/A	N/A
KAL-0722	Inuit	dm2/M1 in eruption	N/A	N/A
KAL-0869	Inuit	M2	N/A	N/A
KAL-0877	Inuit	M2 not fully erupted	N/A	N/A
KAL-1310	Inuit	M2 not fully erupted	N/A	N/A
KAL-1412	Inuit	N/A	adult	N/A
KAL-1418	Inuit	neonate	N/A	N/A
KAL-1459	Inuit	M1/M2 in eruption	N/A	N/A
KAL-1800	Inuit	N/A	Few months old	N/A
MMK 146	South African	dm2	N/A	N/A
MMK 238	South African	dm2	N/A	N/A
SAM-AP-1273	South African	N/A	>15	N/A
SAM-AP-1448	South African	N/A	7	N/A
SAM-AP- 1879	South African	N/A	adult	
SAM-AP-3027	South African	N/A	7	
SAM-AP-34	South African	N/A	adult	N/A
SAM-AP-3691	South African	N/A	11	N/A
SAM-AP-3737A	South African	N/A	10	N/A

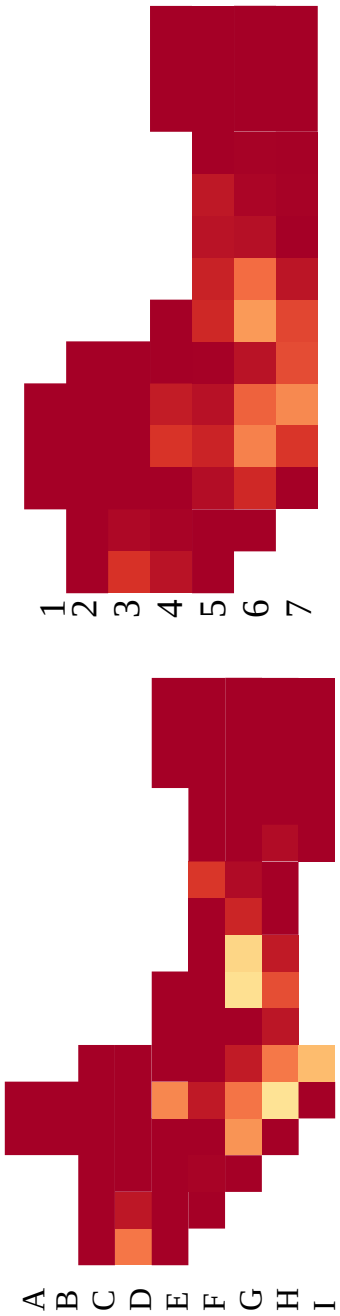
SAM-AP-4790	South African	N/A	adult	N/A
SAM-AP-4844	South African	N/A	adult	N/A
SAM-AP-6052	South African	N/A	5	N/A
SAM-AP-6340	South African	N/A	3	N/A
SAM-AP-6348A	South African	N/A	>15	N/A
UCT-189	South African	N/A	0.5	N/A
UCT-195	South African	N/A	.2	N/A
ULAC 131/199	Western European	N/A	adult	N/A
ULAC 186/201	Western European	N/A	adult	N/A
ULAC 201/12	Western European	N/A	adult	N/A

APPENDIX F INDIVIDUAL MAPS

1879_73_121_RMand_Ext

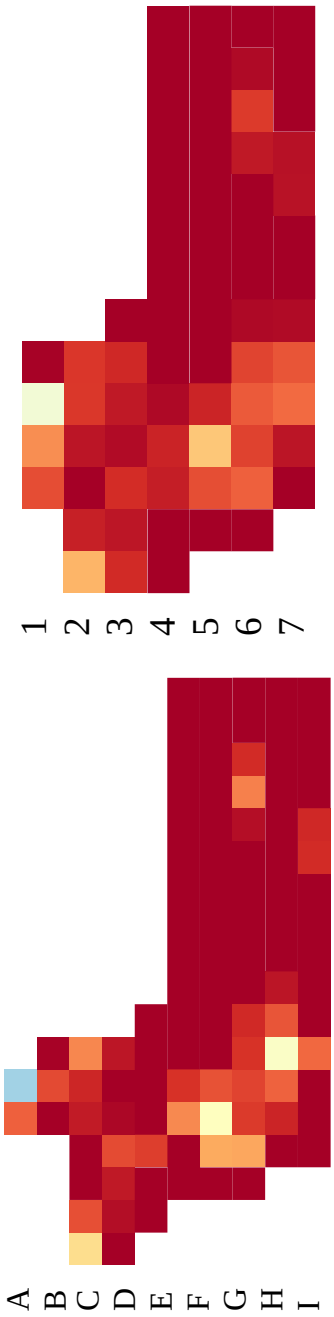


1879_73_121_RMand_Int

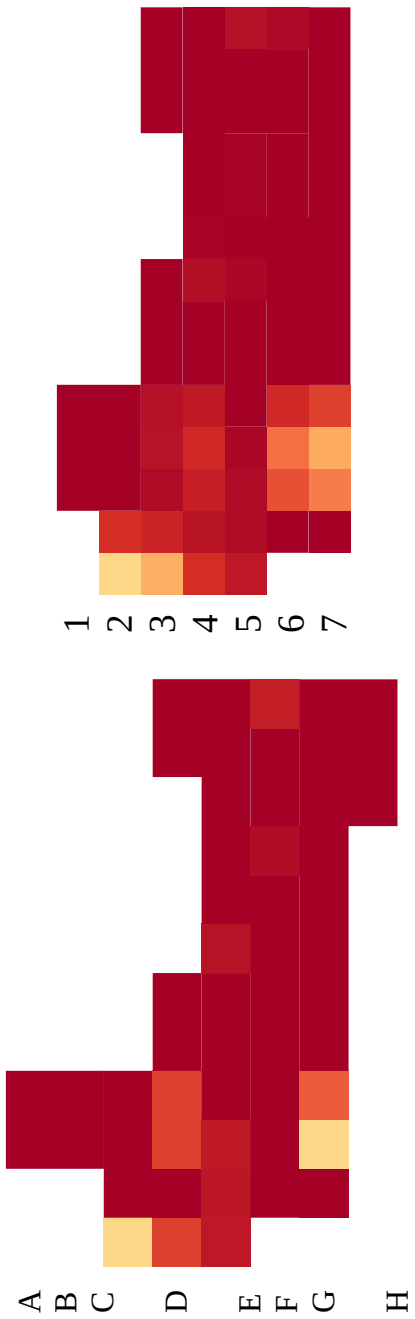


1886_87_99_RMand_Ext

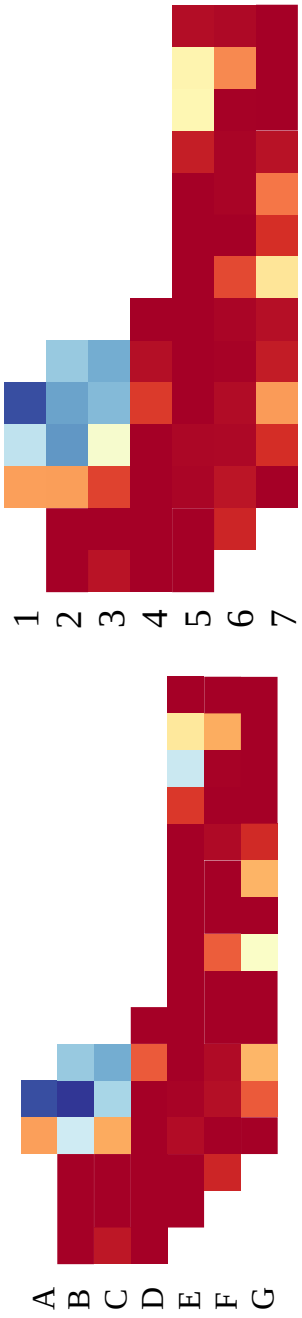
J



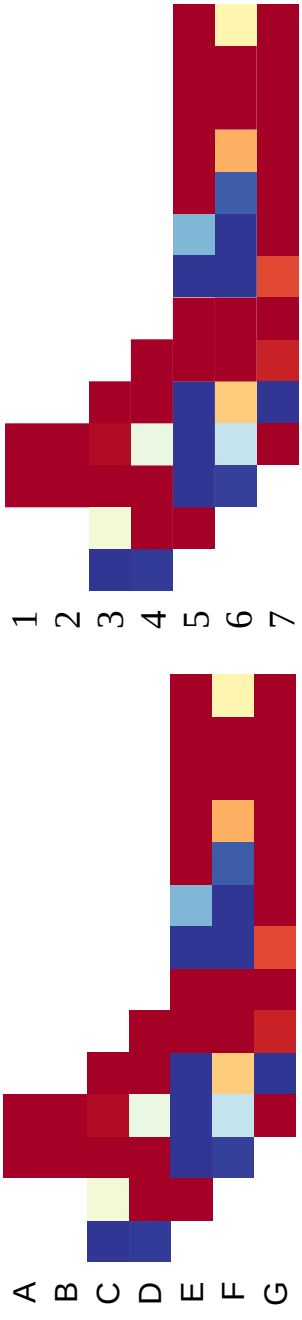
1886_87_99_RMand_Int



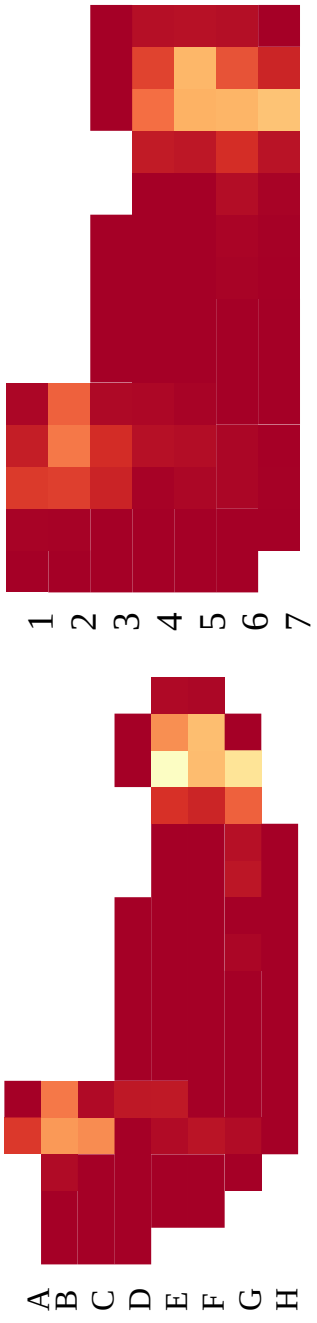
1890_91_21_129_RMand_Ext



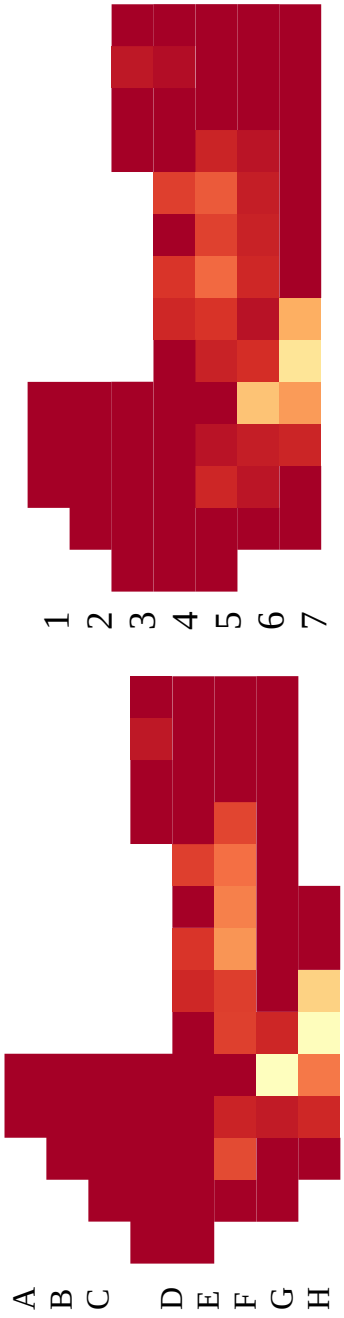
1890_91_21_129_RMand_Int



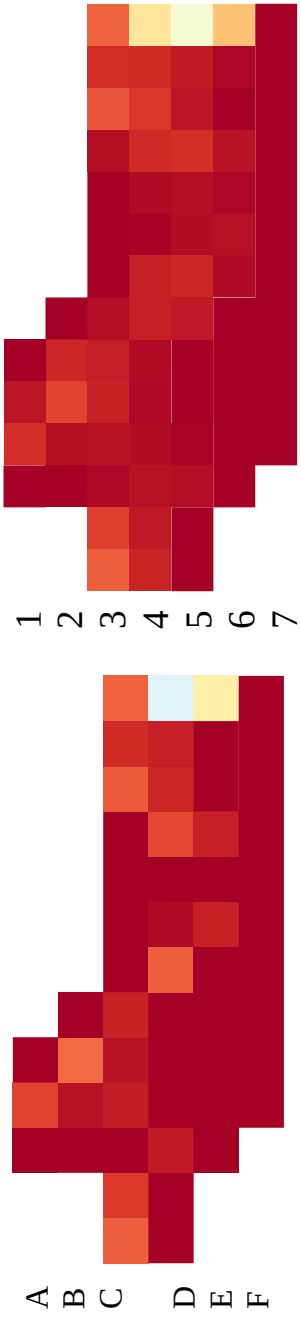
1892_93_285_179_RMand_Ext



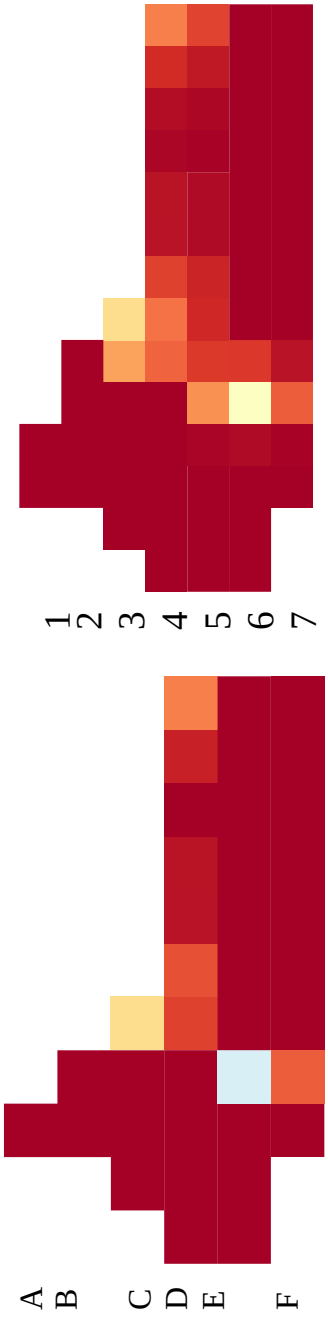
1892_93_285_179_RMand_Int



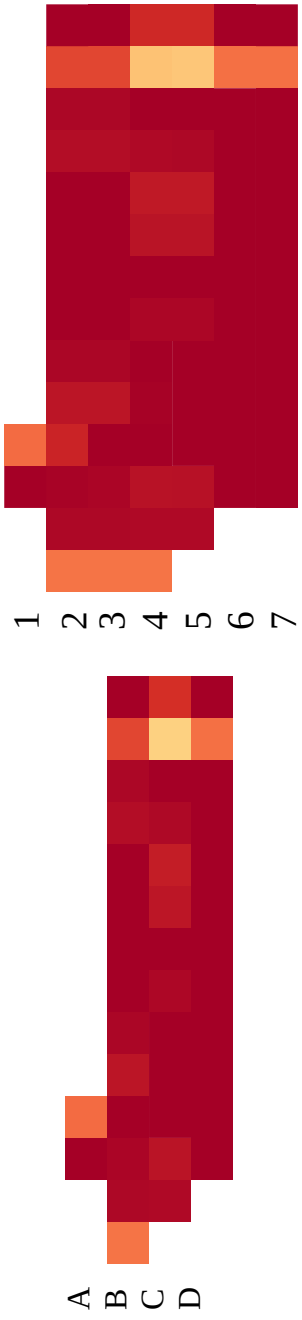
1892_93_286_180_RMand_Ext



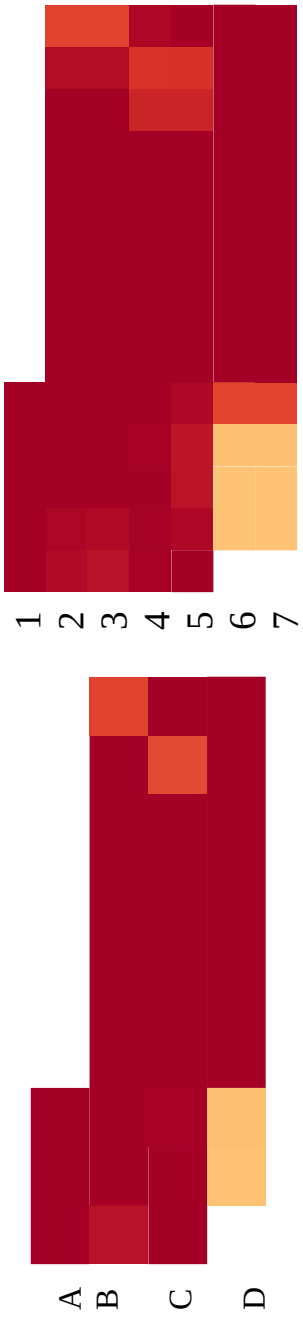
1892_93_286_180_RMand_Int



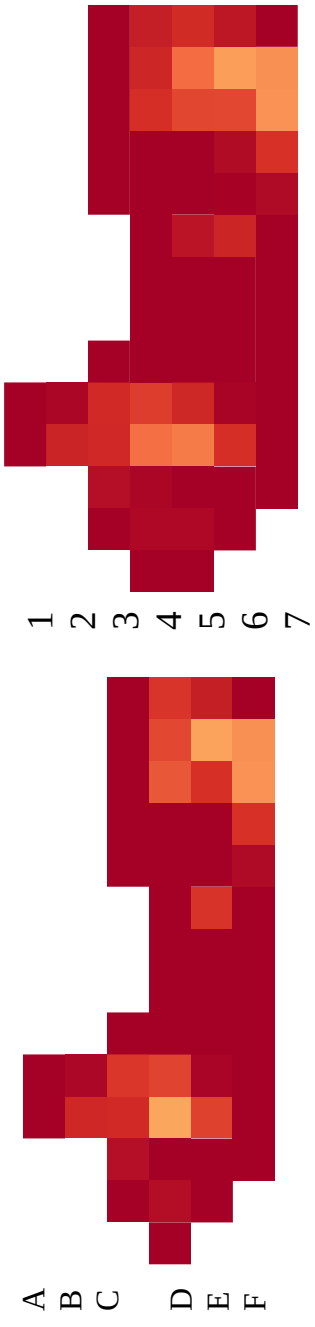
1892_93_307_197_RMand_Ext



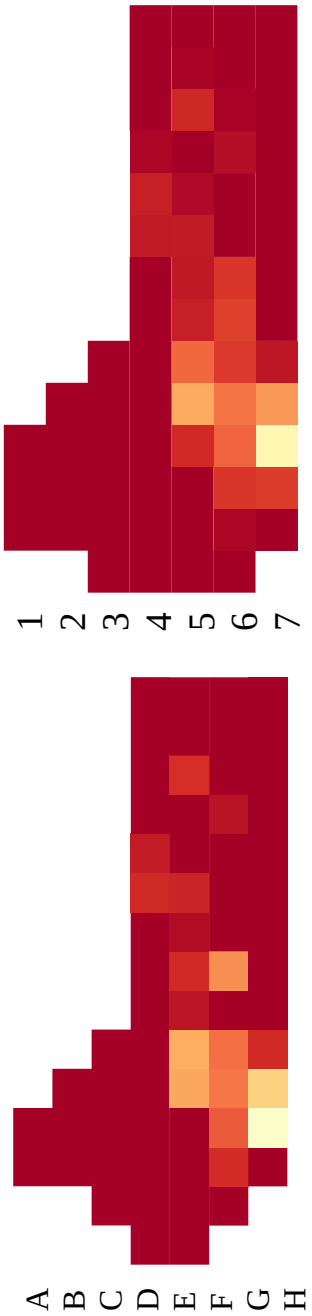
1892_93_307_197_RMand_Int



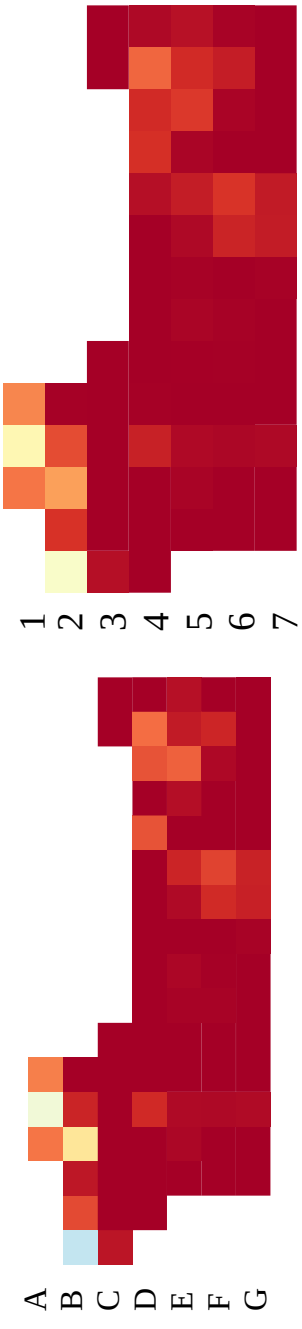
1893_94_8_RMand_Ext



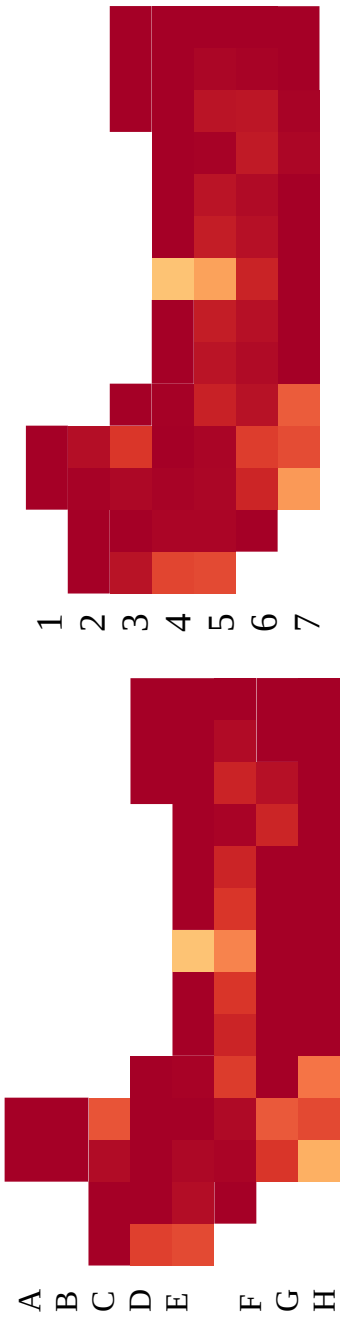
1893_94_8_RMand_Int



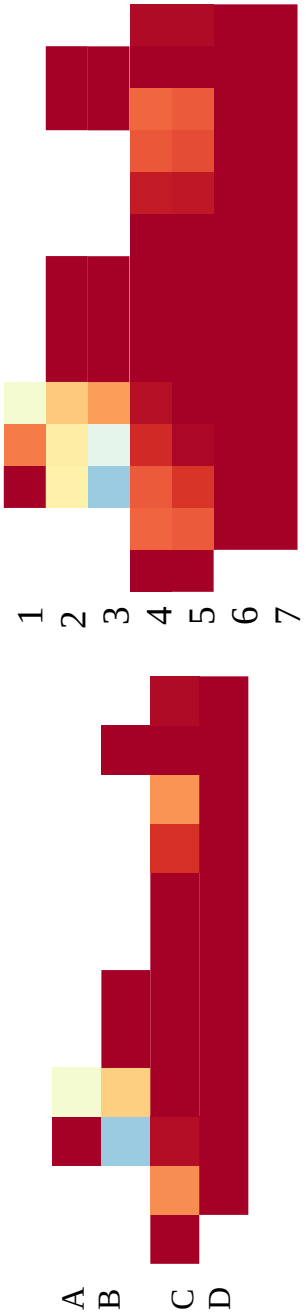
1893_94_86_RMand_Ext



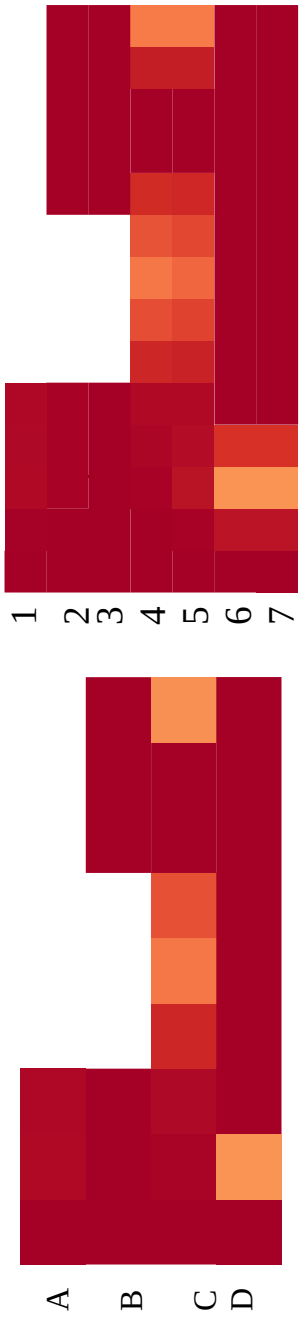
1893_94_86_RMand_Int



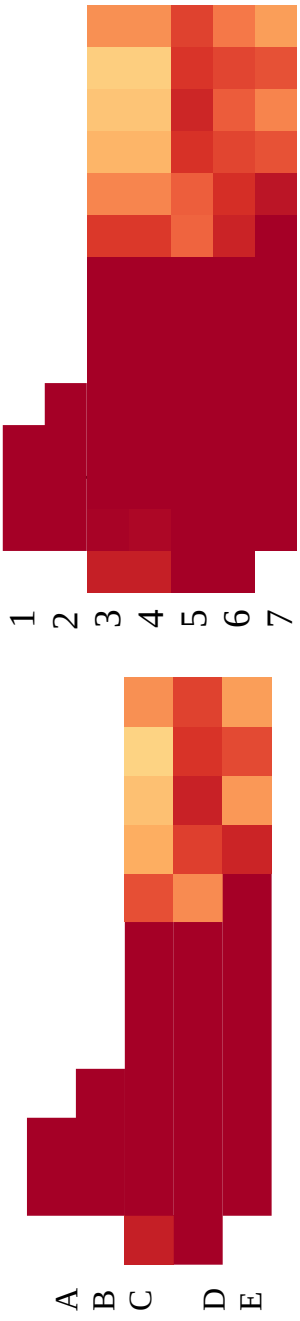
1894_95_142_277_RMand_Ext



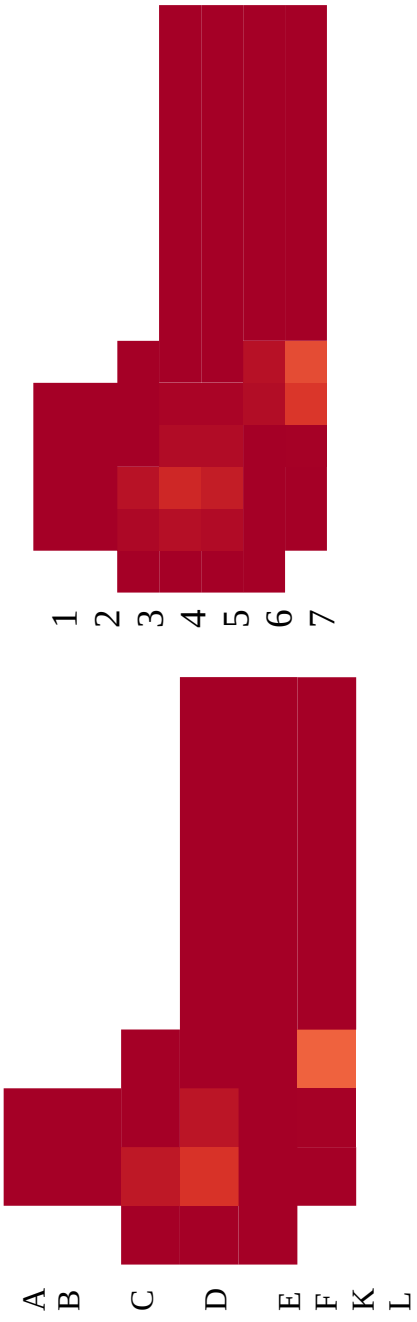
1894_95_142_277_RMand_Int



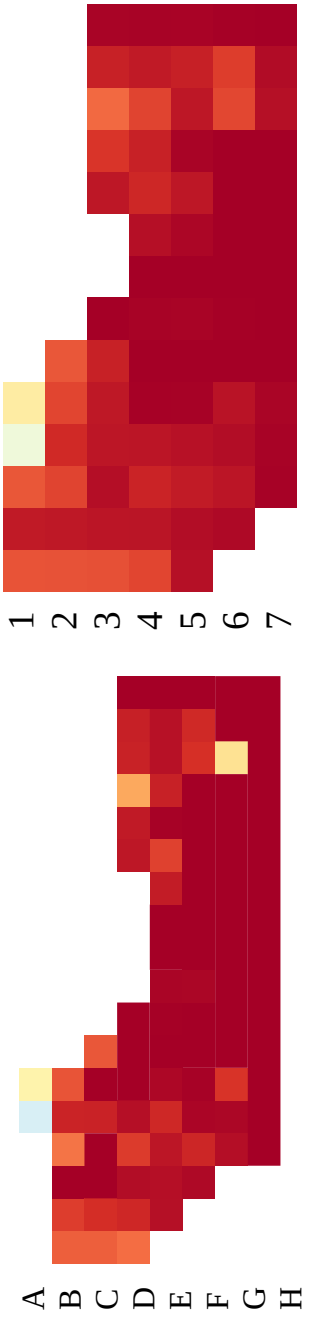
1898_99_232_476_RMand_Ext



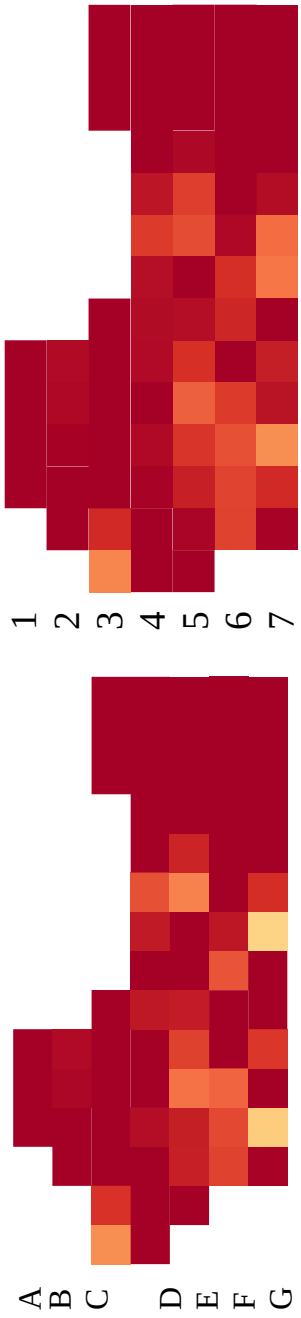
1898_99_232_476_RMand_Int



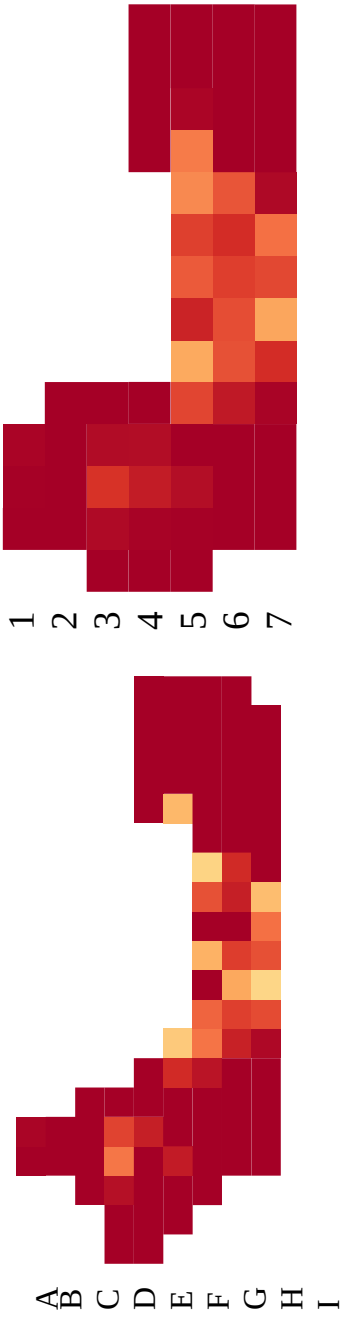
1899_288_512_RMand_Ext



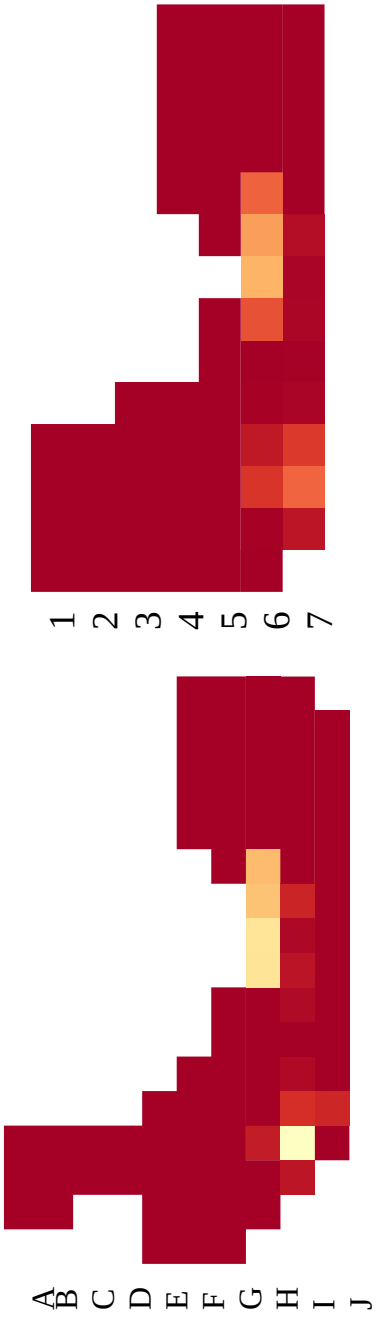
1899_288_512_RMand_Int



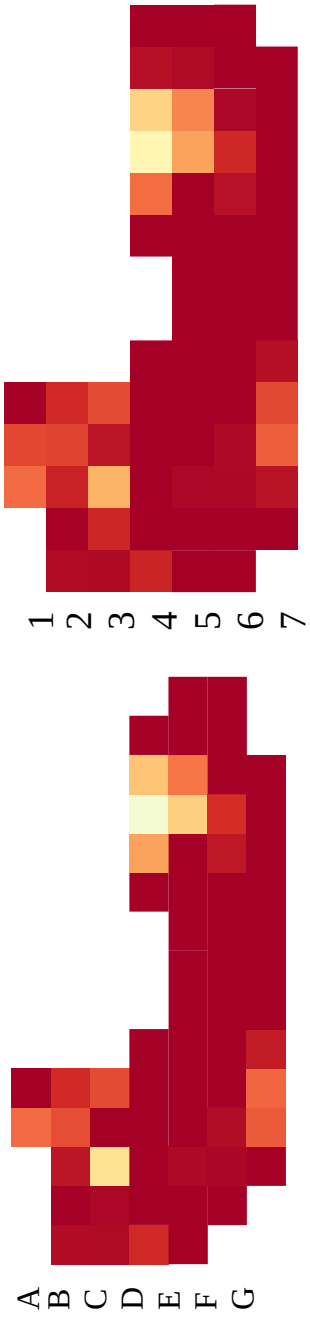
1902_144_594_RMand_Ext



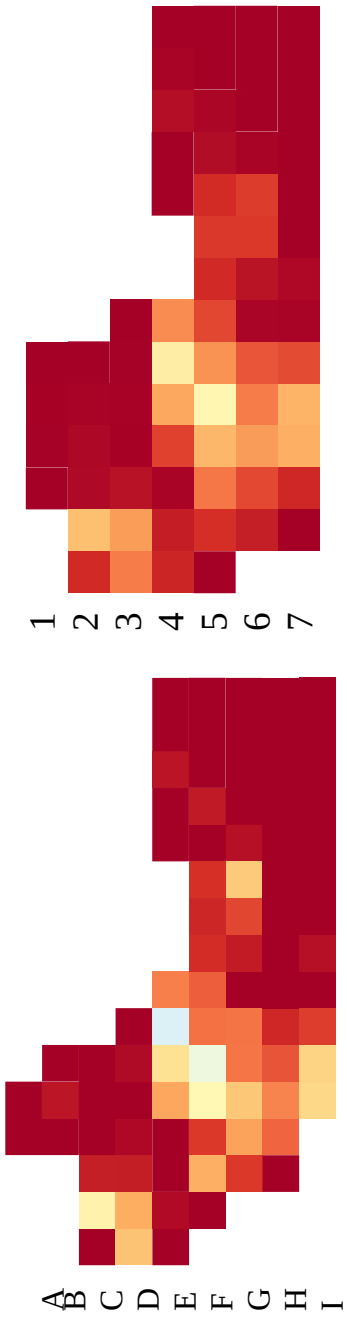
1902_144_594_RMand_Int



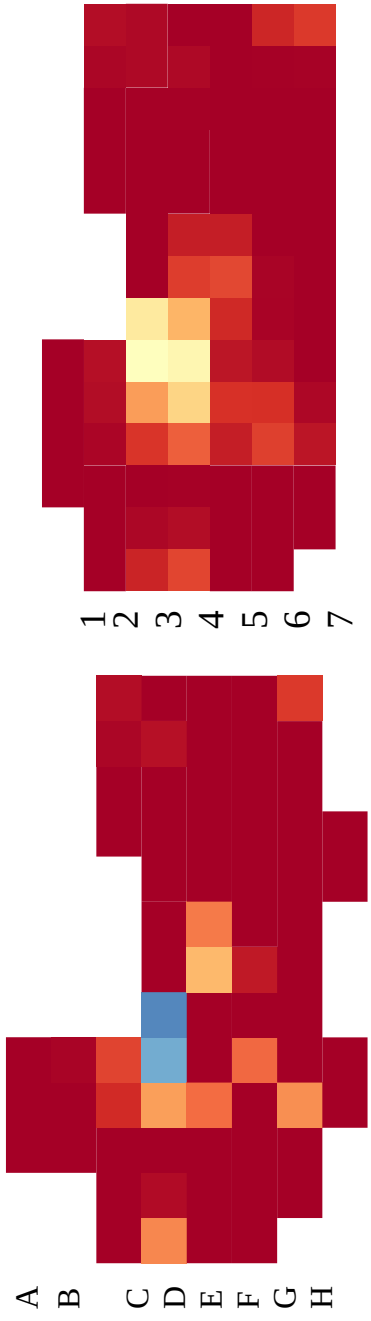
1906_07_37_RMand_Ext



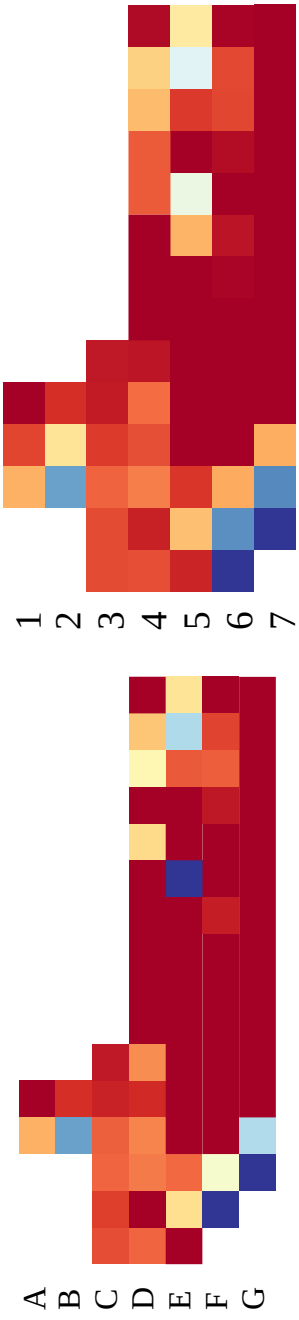
1906_07_37_RMand_Int



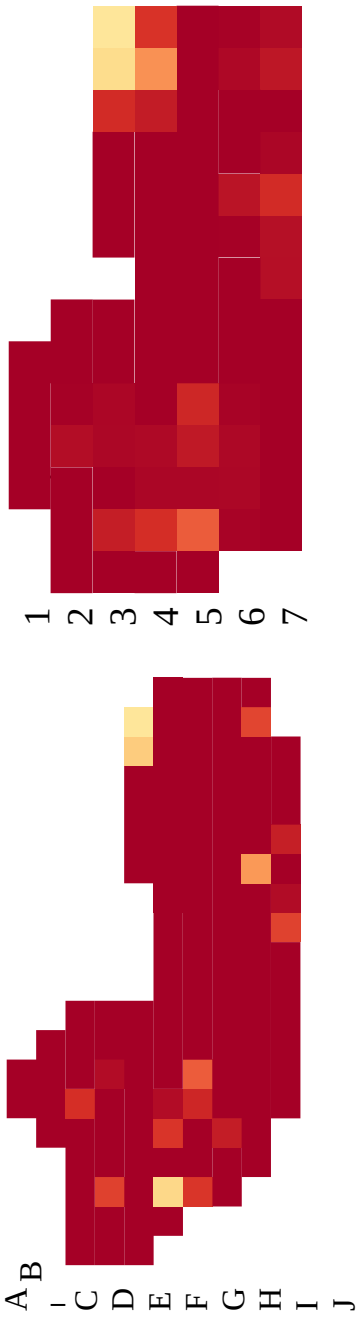
KAL_0028_RMand_Ext



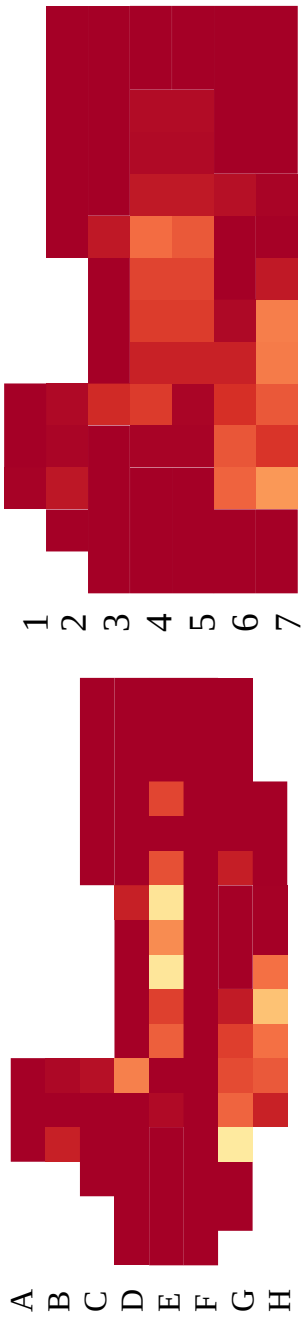
KAL_0028_RMand_Int



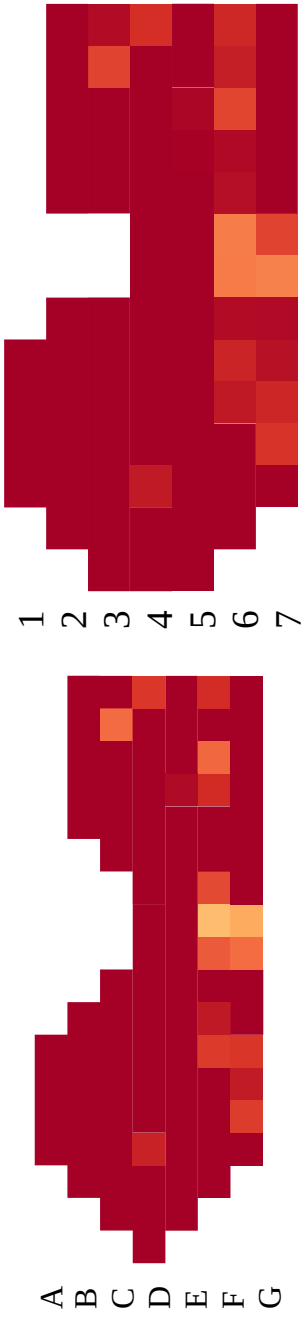
KAL_0109_RMand_Ext



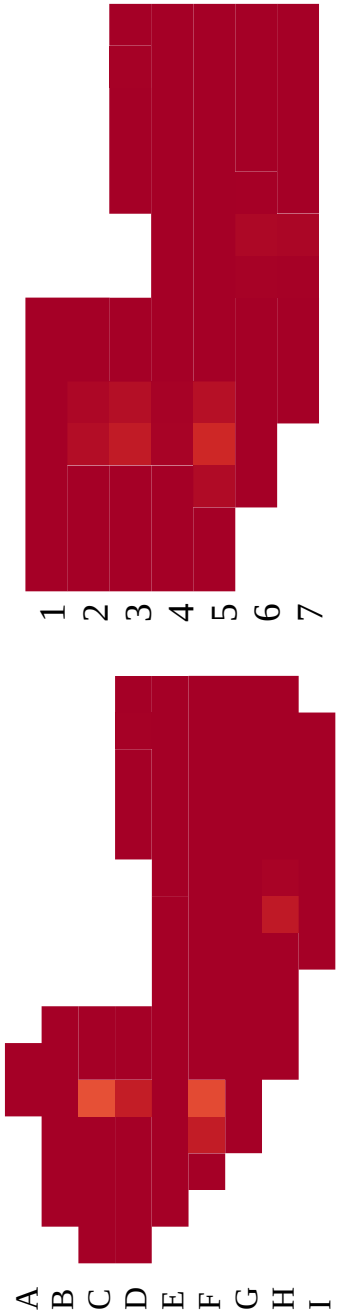
KAL_0109_RMand_Int



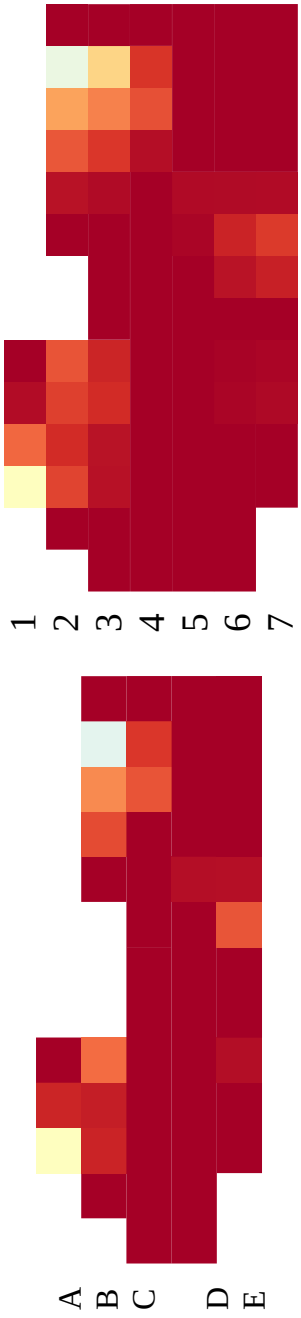
KAL_0122_RMand_Ext



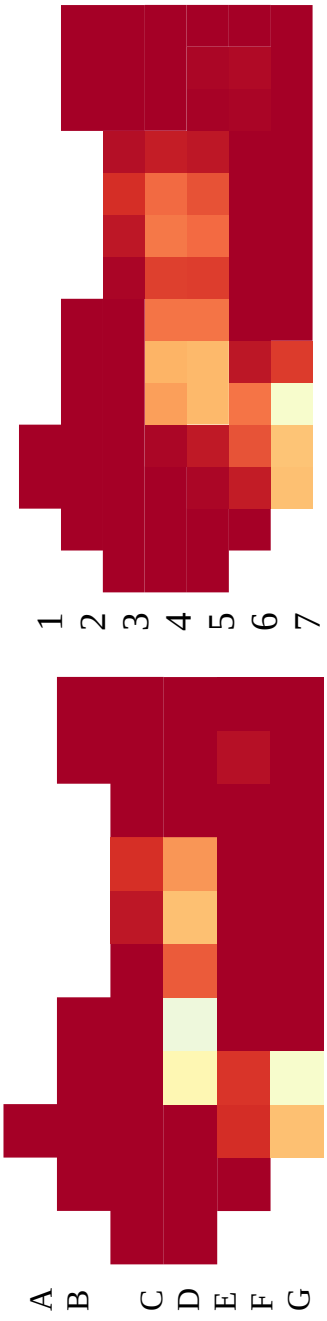
KAL_0122_RMand_Int



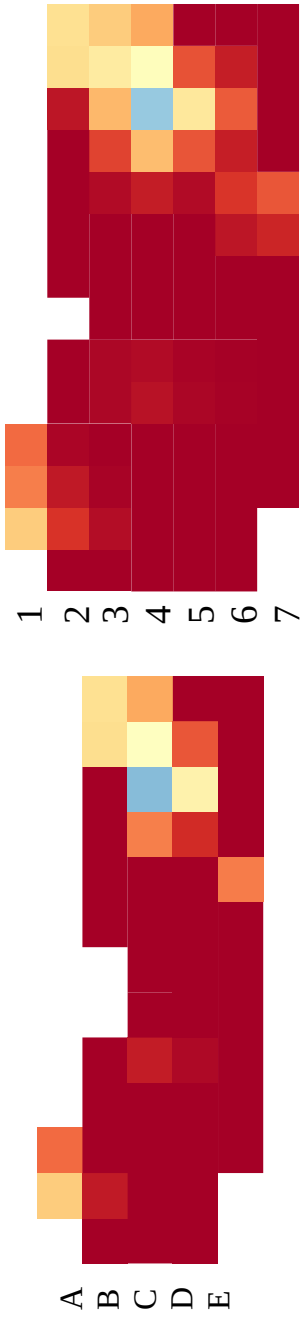
KAL_0168_RMand_Ext



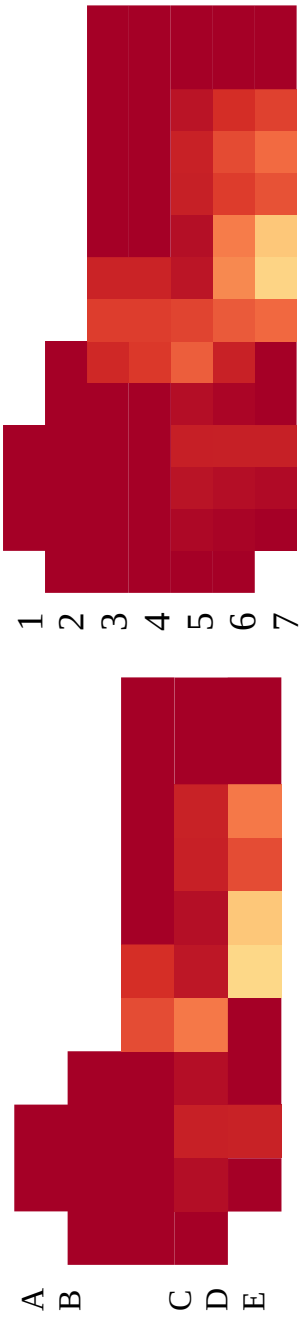
KAL_0168_RMand_Int



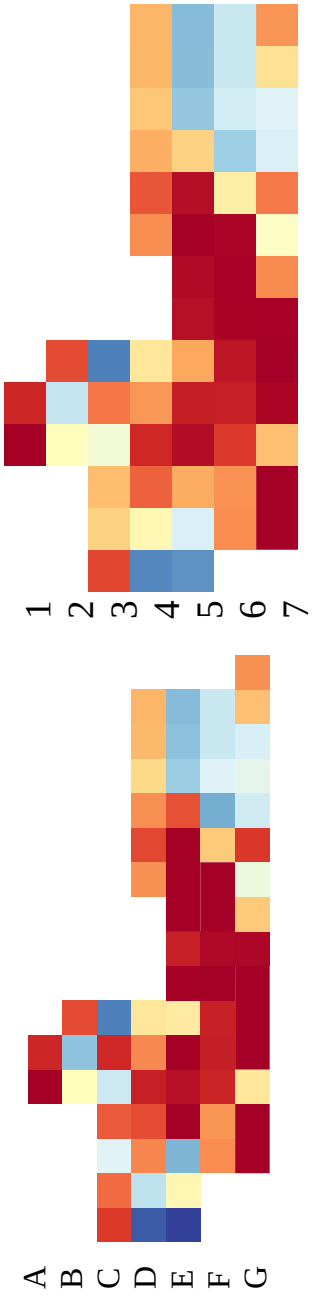
KAL_0169_RMand_Ext



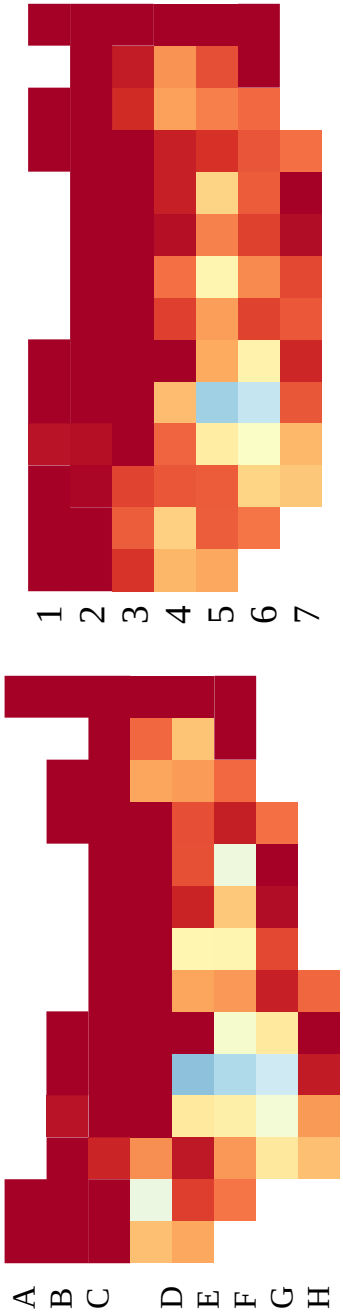
KAL_0169_RMand_Int



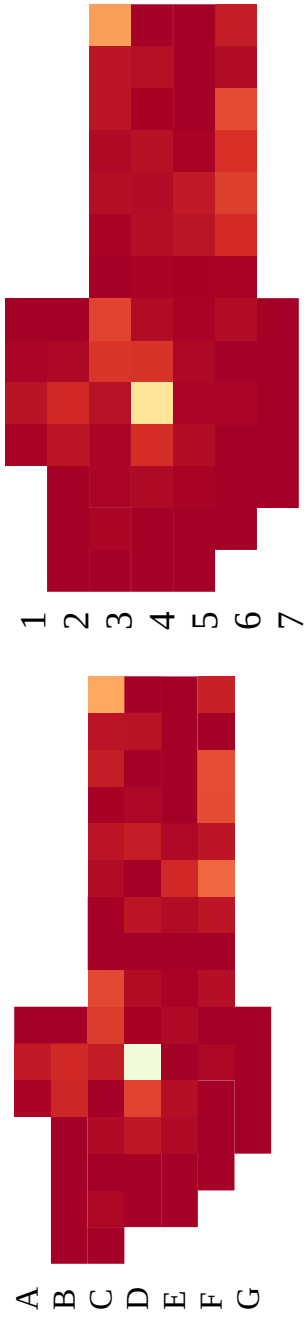
KAL_0401_RMand_Ext



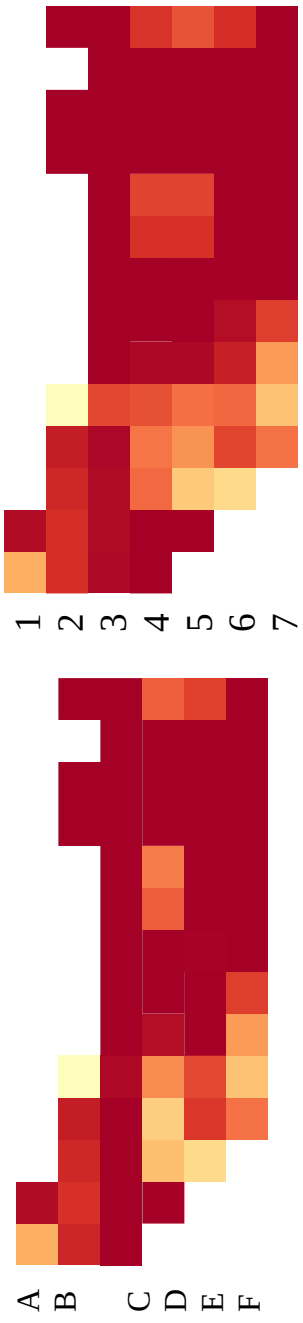
KAL_0401_RMand_Int



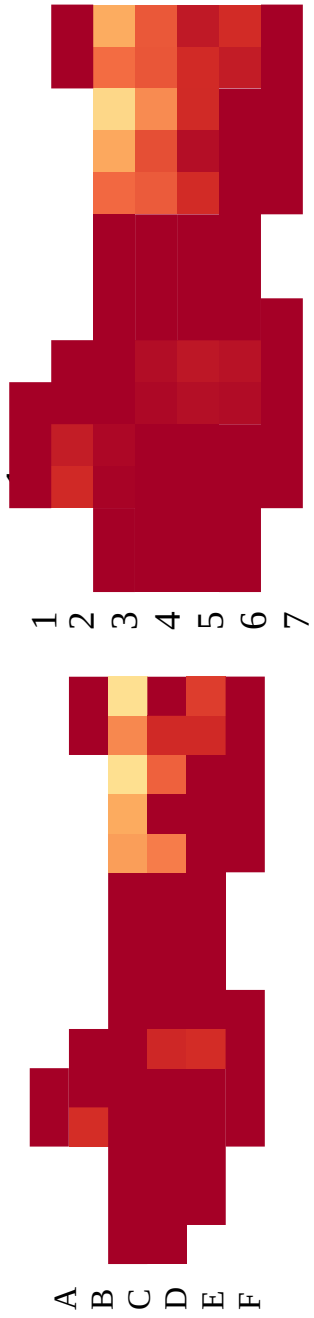
KAL_0674_RMand_Ext



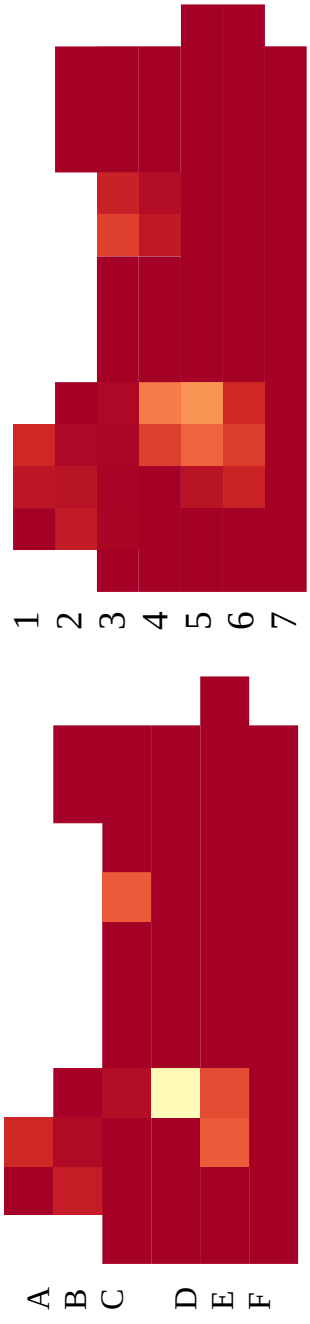
KAL_0674_RMand_Int



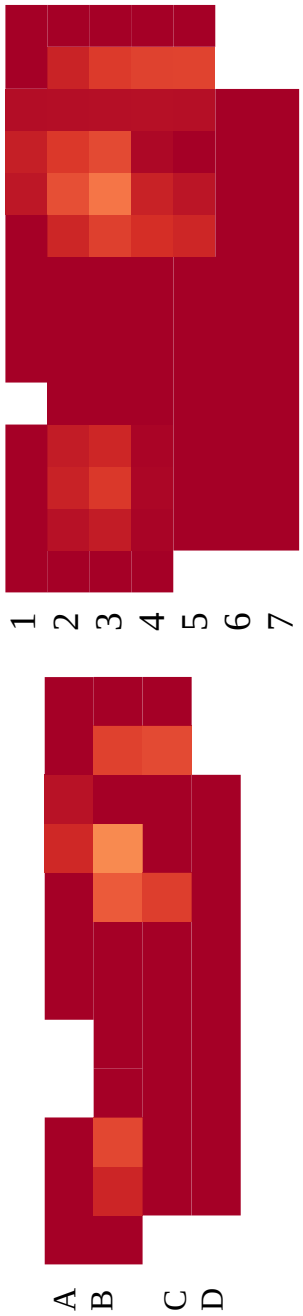
KAL_0707_RMand_Ext



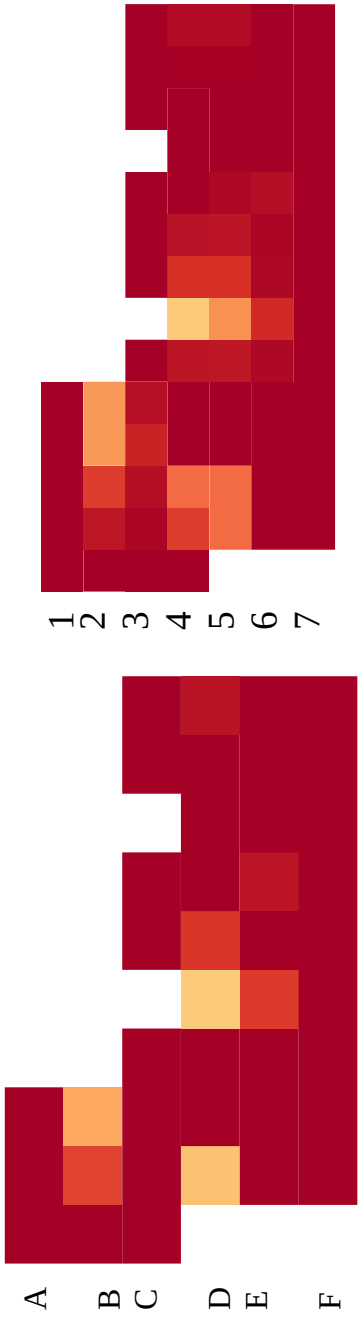
KAL_0707_RMand_Int



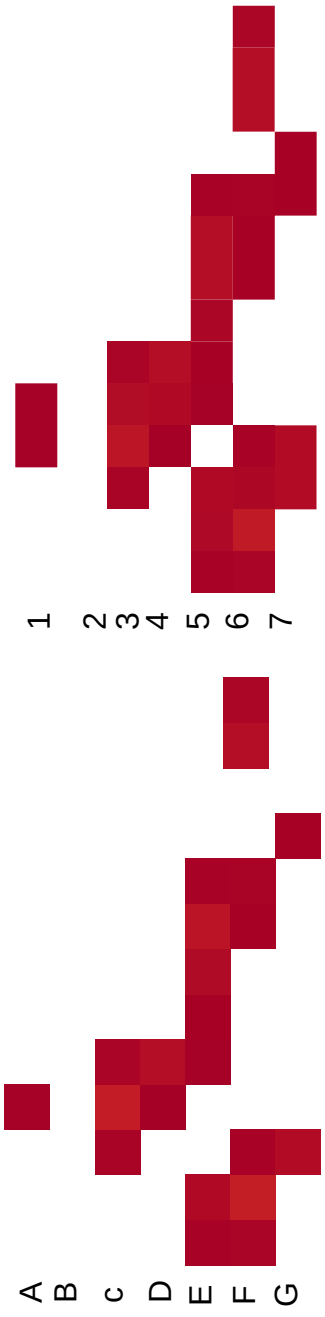
KAL_0722_RMand_Ext



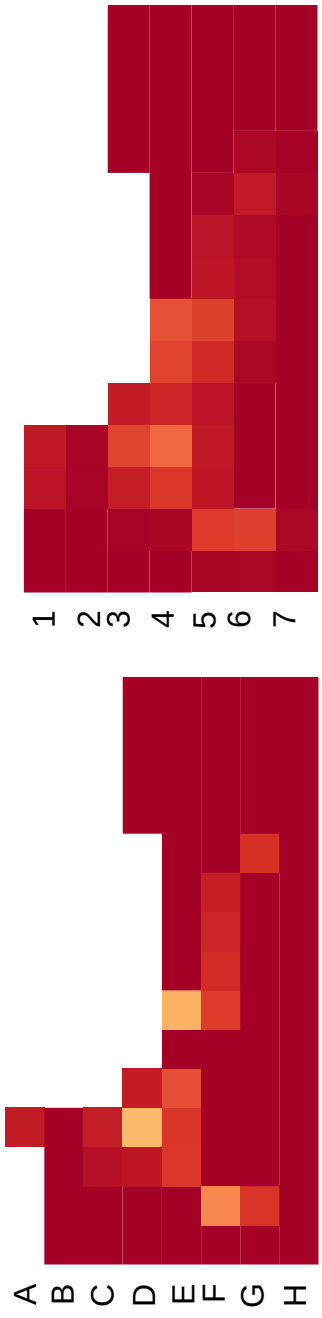
KAL_0722_RMand_Int



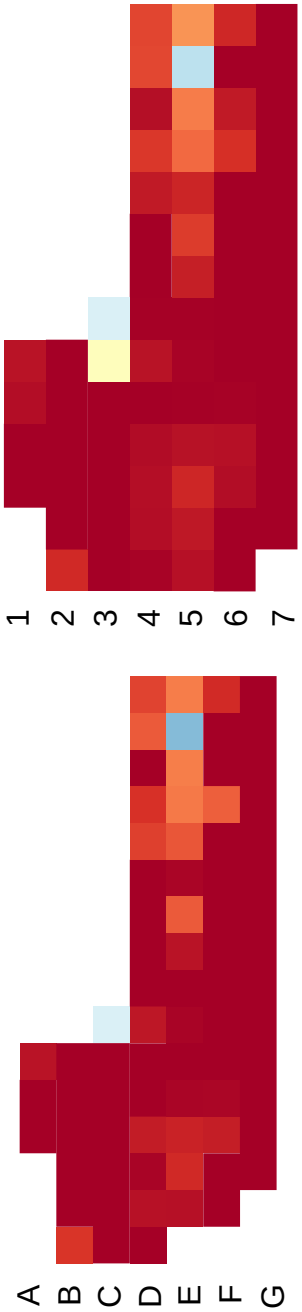
KAL_0869_RMand_Ext



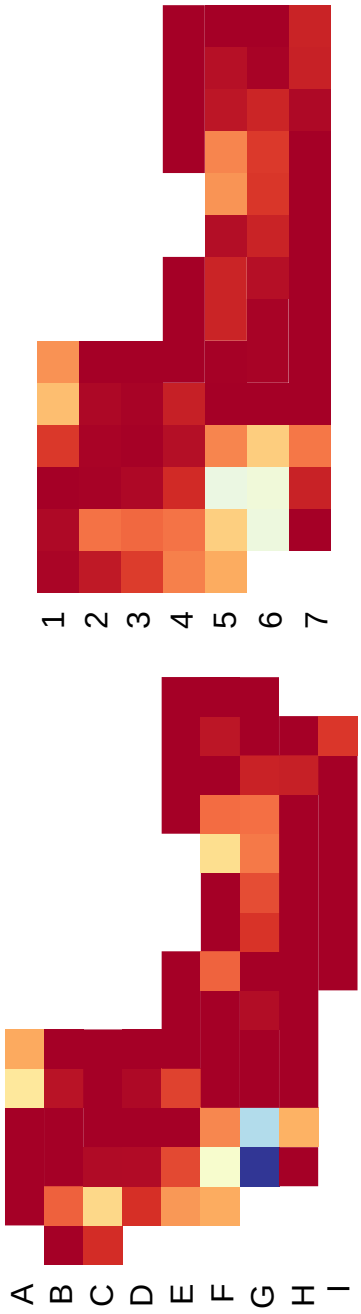
KAL_0869_RMand_Int



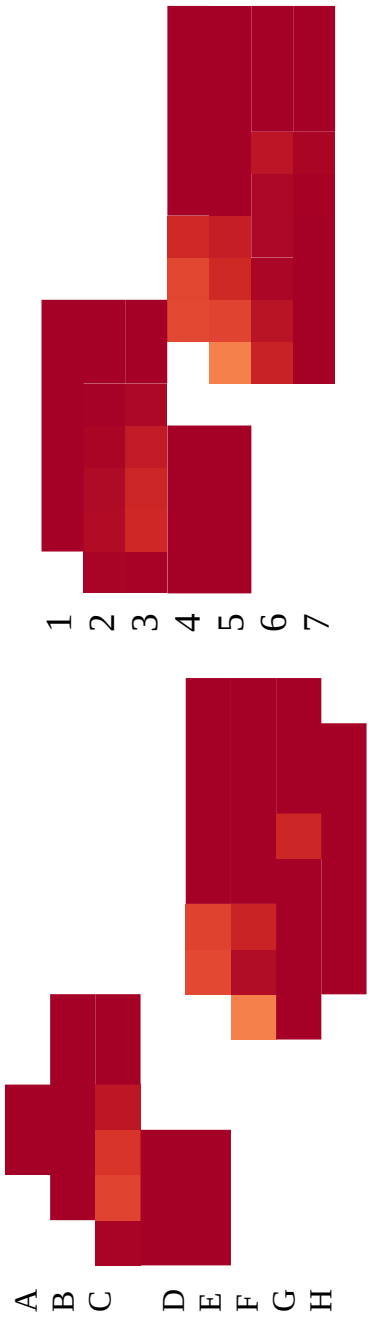
KAL_0877_RMand_Ext



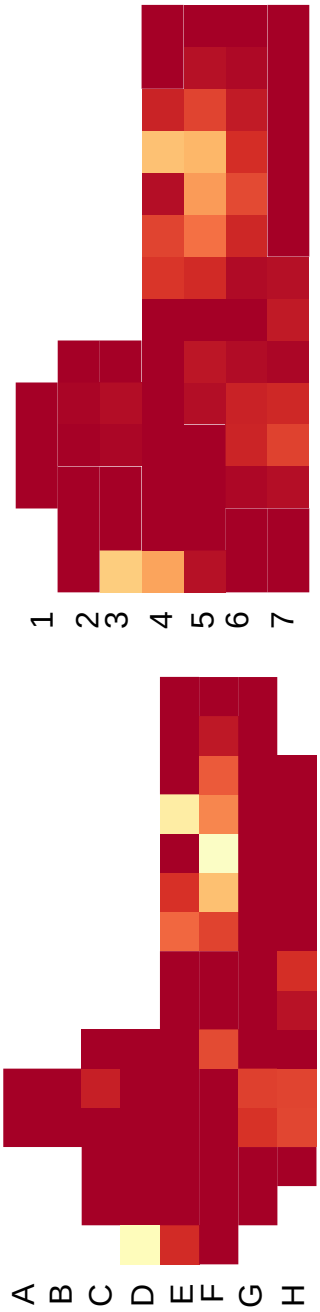
KAL_0877_RMand_Int



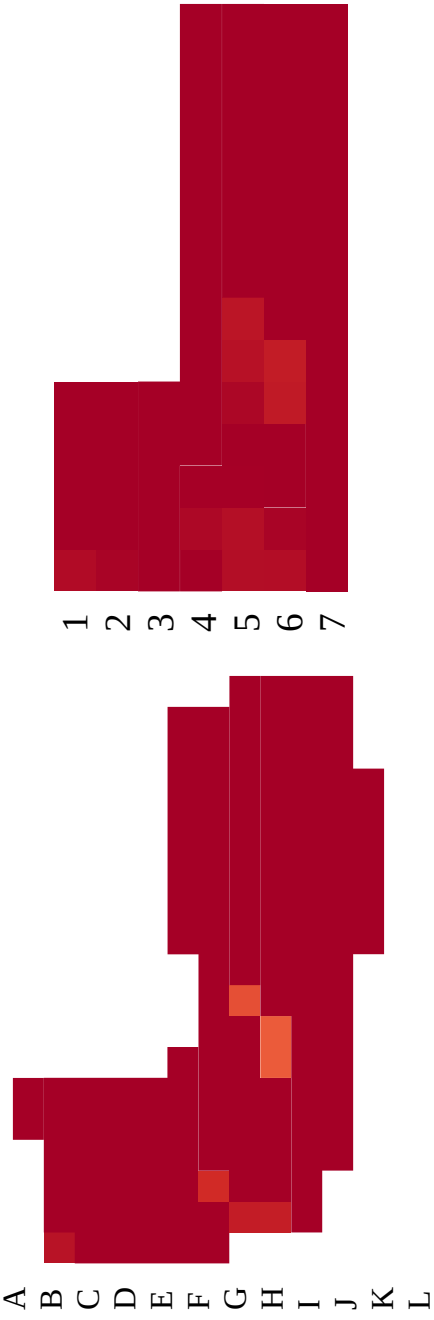
KAL_1310_RMand_Ext



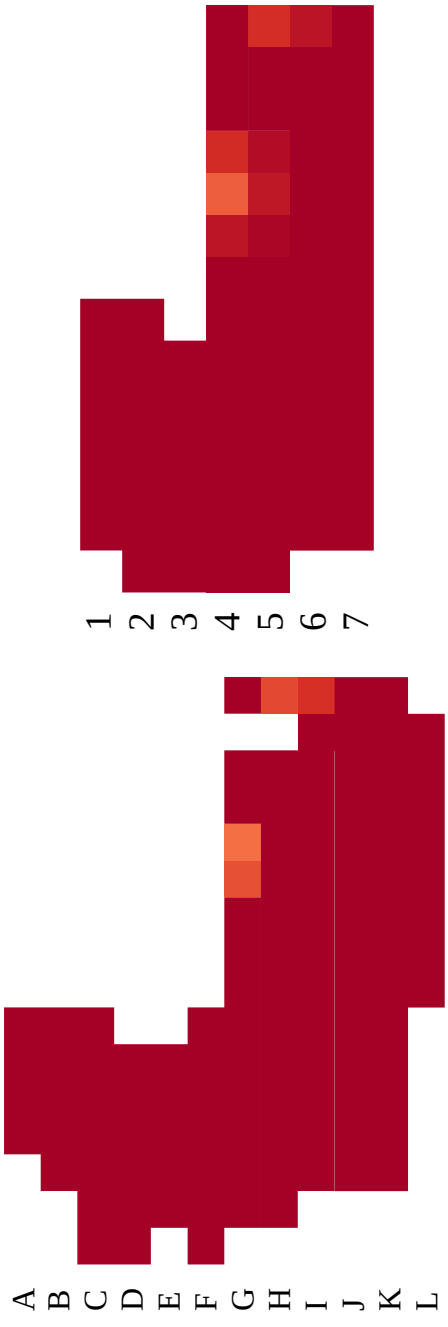
KAL_1310_RMand_Int



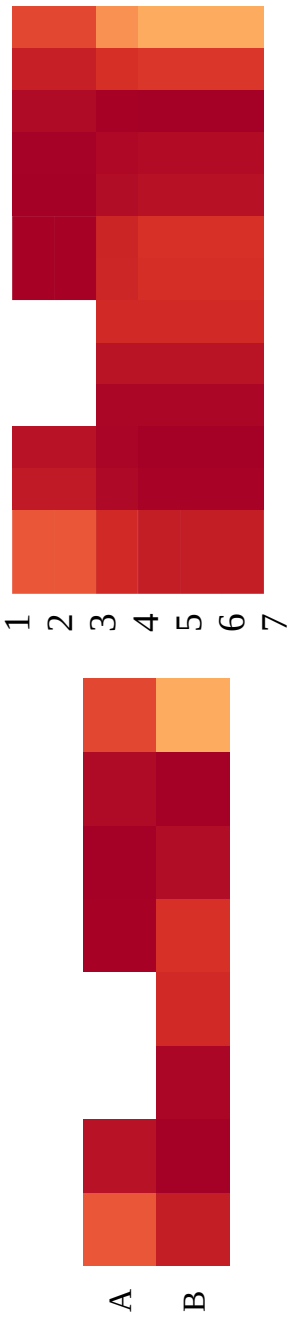
KAL_1412_RMand_Ext



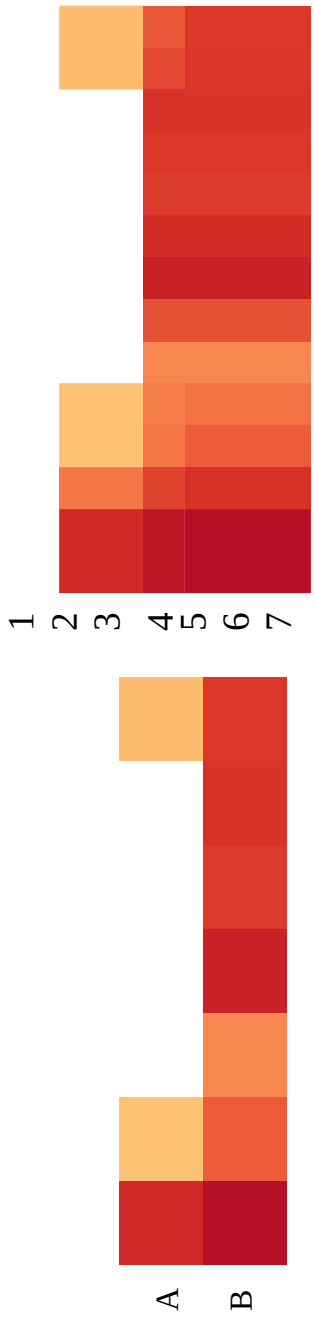
KAL_1412_RMand_Int



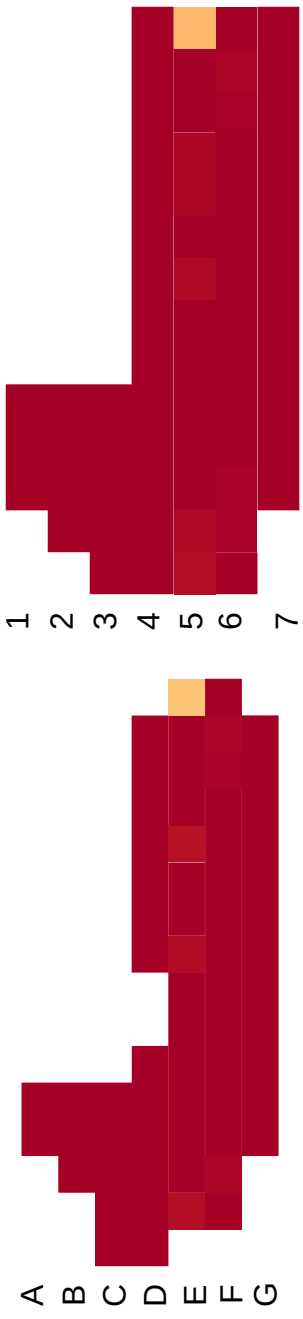
KAL_1418_RMand_Ext



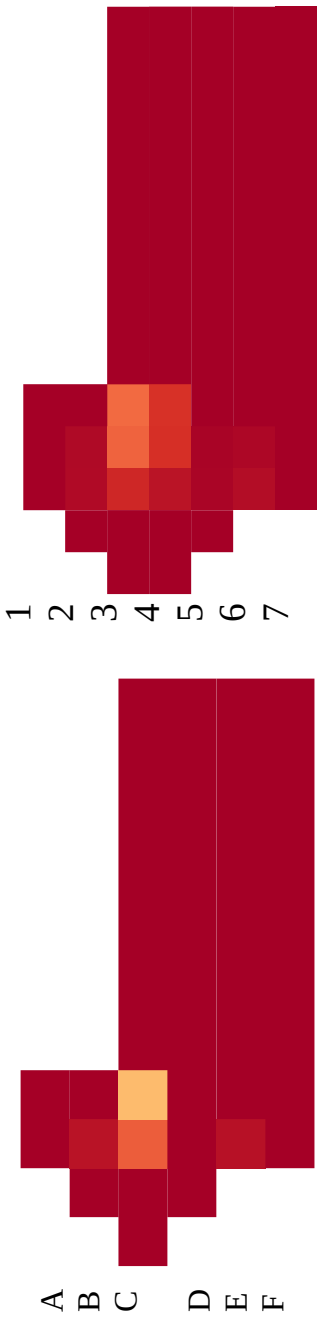
KAL_1418_RMand_Int

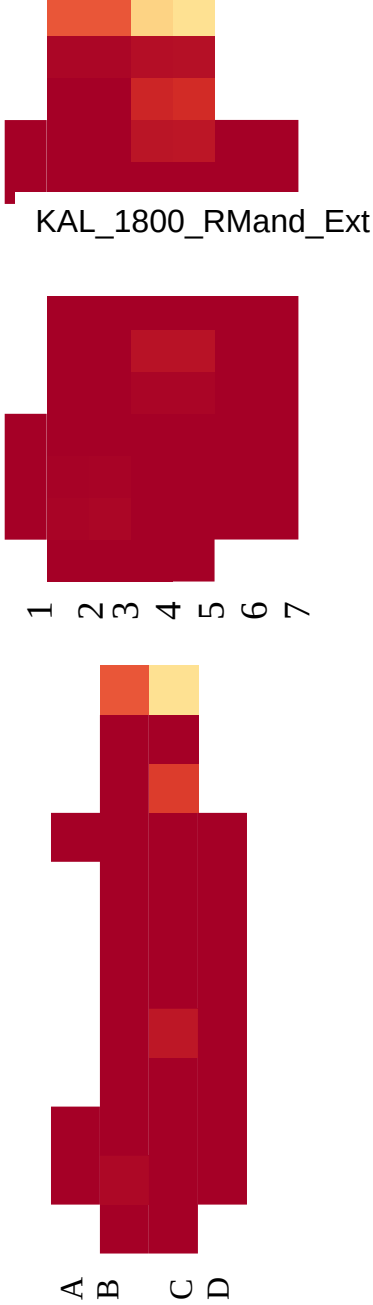


KAL_1459_RMand_Ext

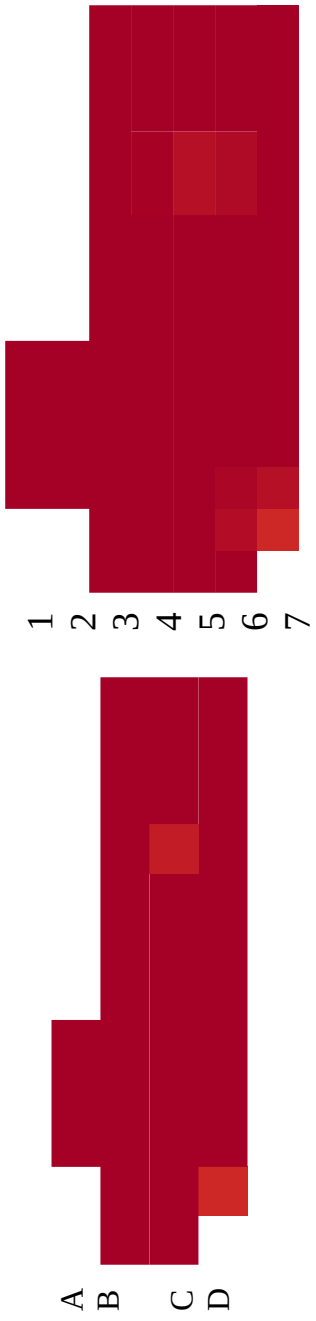


KAL_1459_RMand_Int

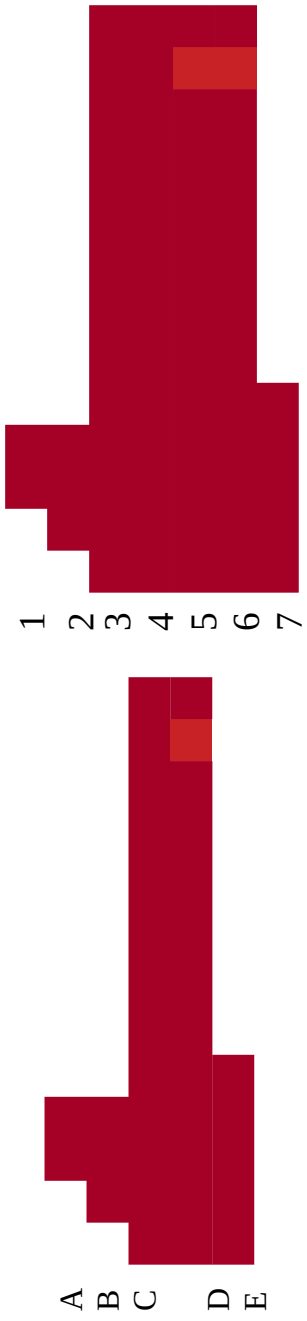



KAL_1800_RMand_Ext
1
2
3
4
5
6
7
A
B
C
D

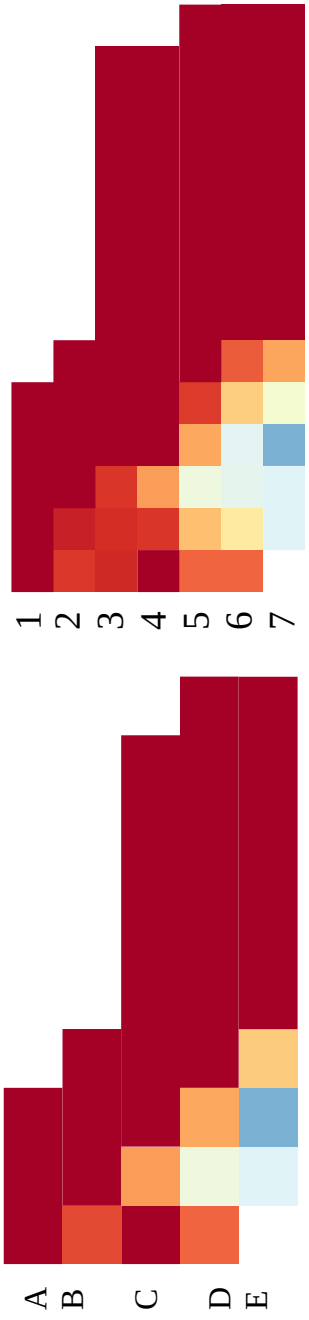
KAL_1800_RMand_Int



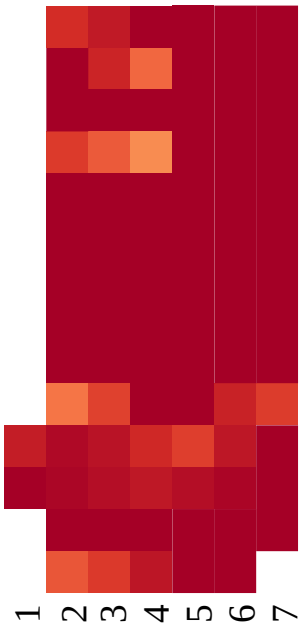
MMK_146_RMand_Ext

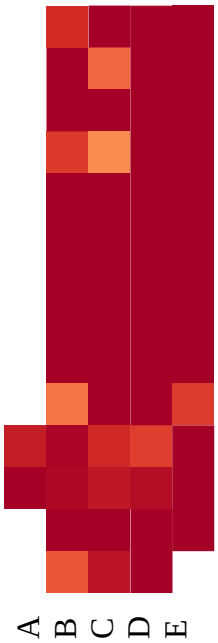


MMK_146_RMand_Int

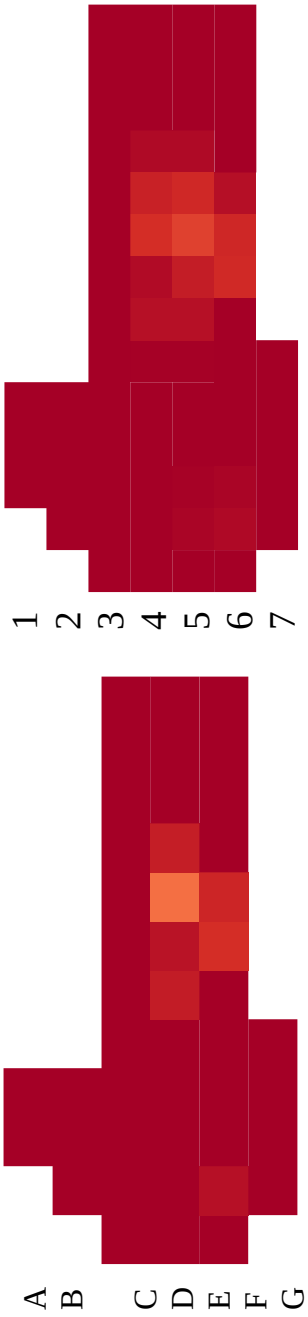


MMK_238_RMand_Ext

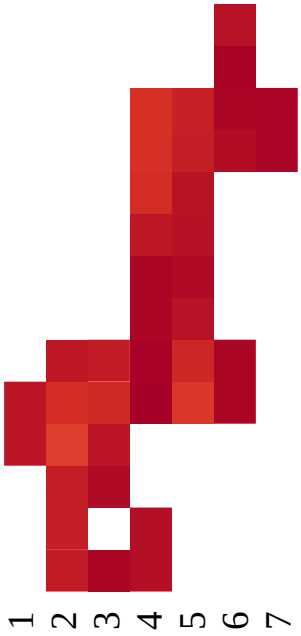


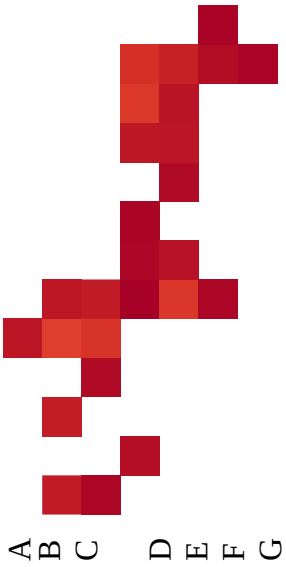


MMK_238_RMand_Int

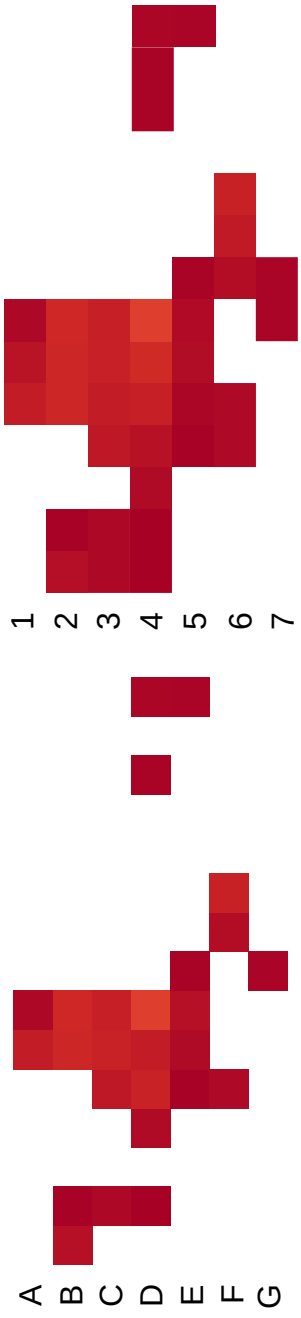


SAM_AP_34_RMand_Int

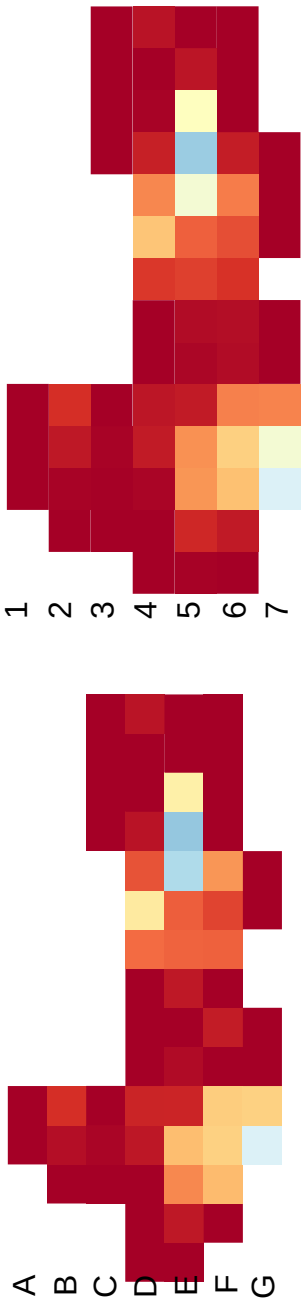




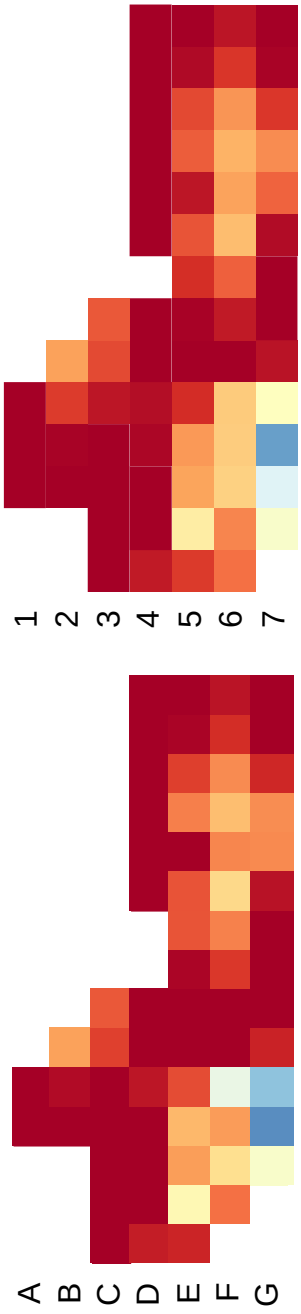
SAM_AP_34_RMand_Ext



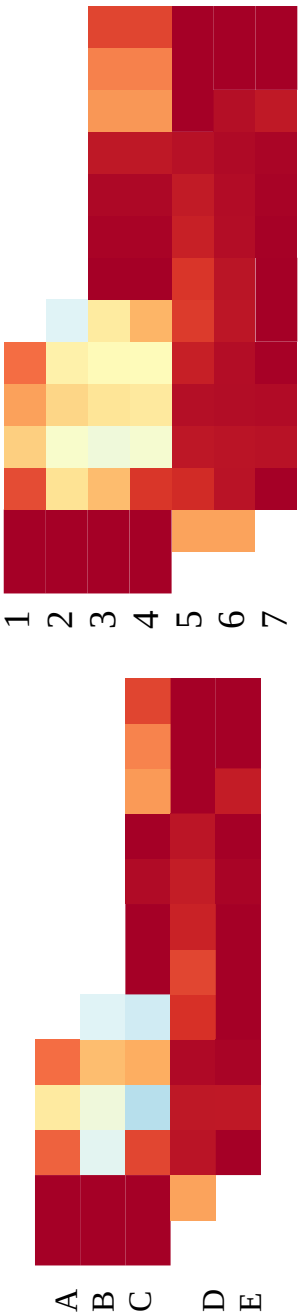
SAM_AP_1273_RMand_Ext



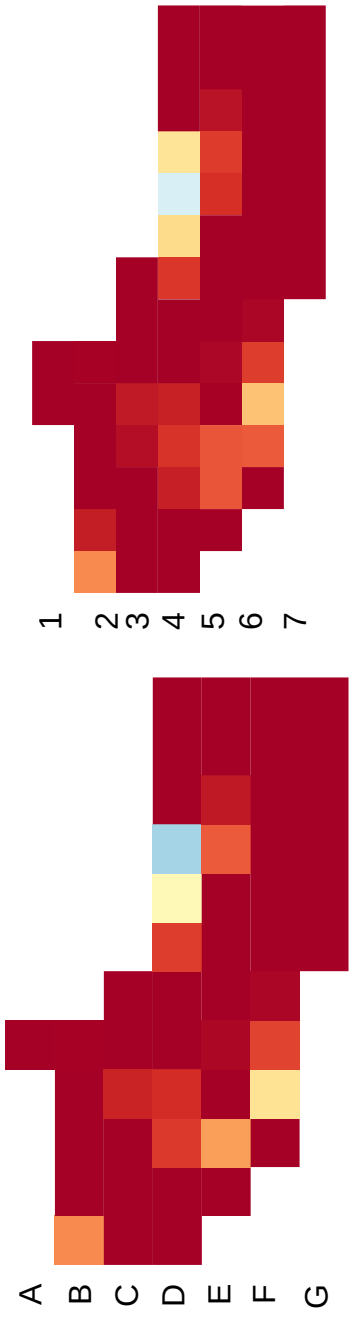
SAM_AP_1273_RMand_Int



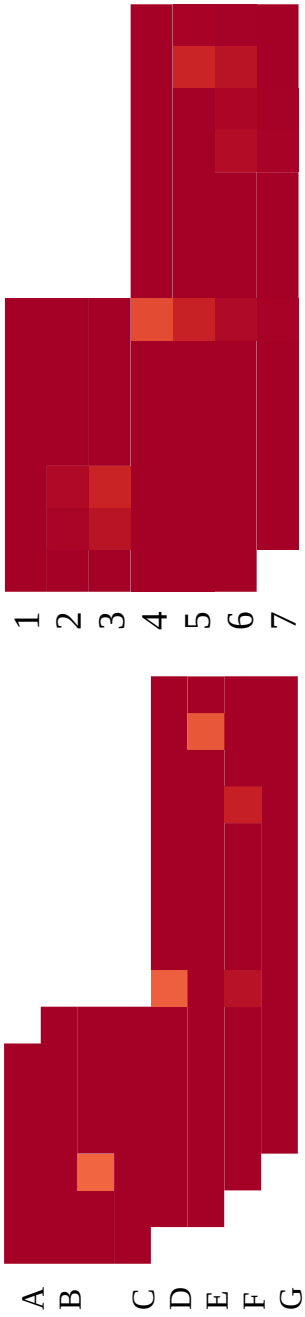
SAM_AP_1448_RMand_Ext



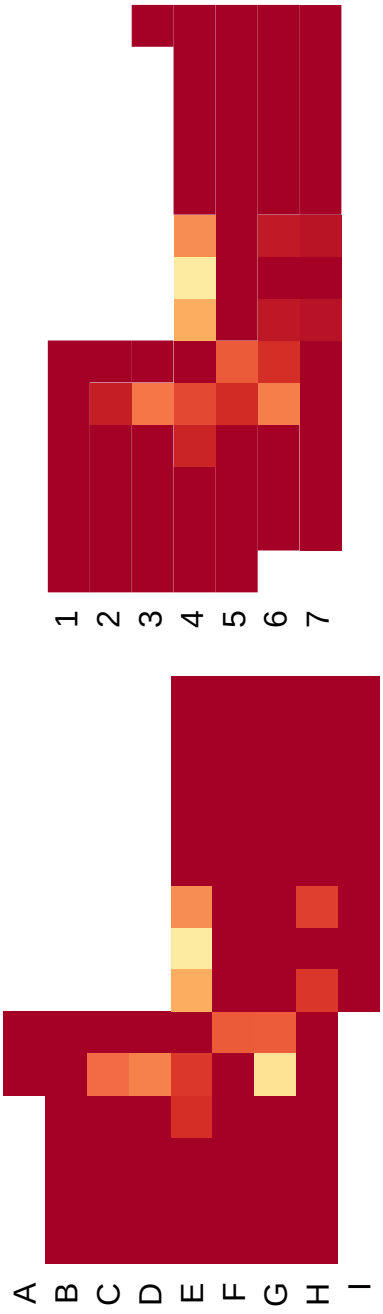
SAM_AP_1448_RMand_Int



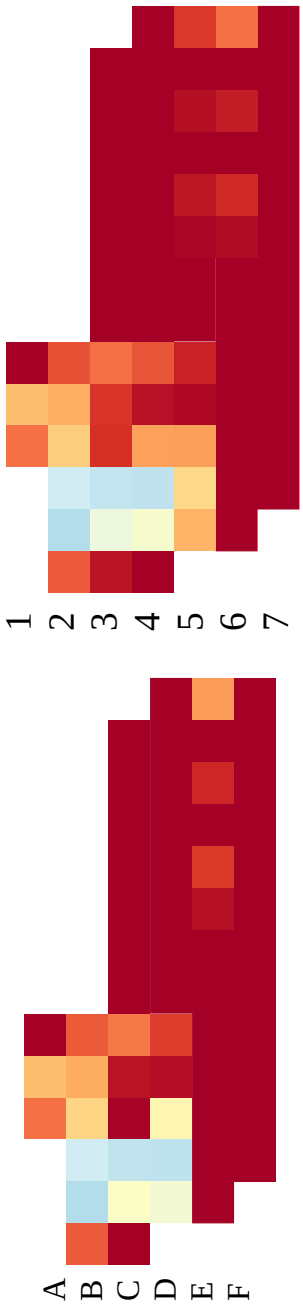
SAM_AP_1879_RMand_Ext



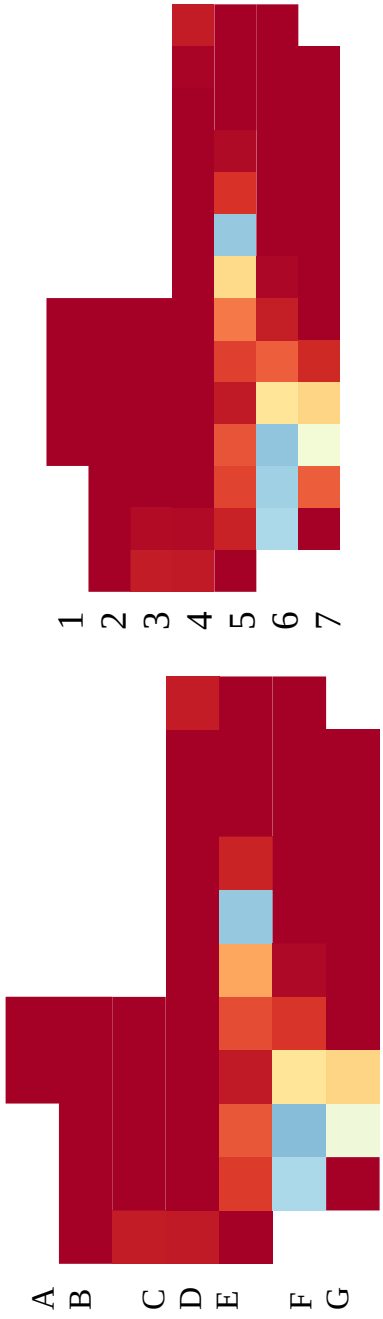
SAM_AP_1879_RMand_Int



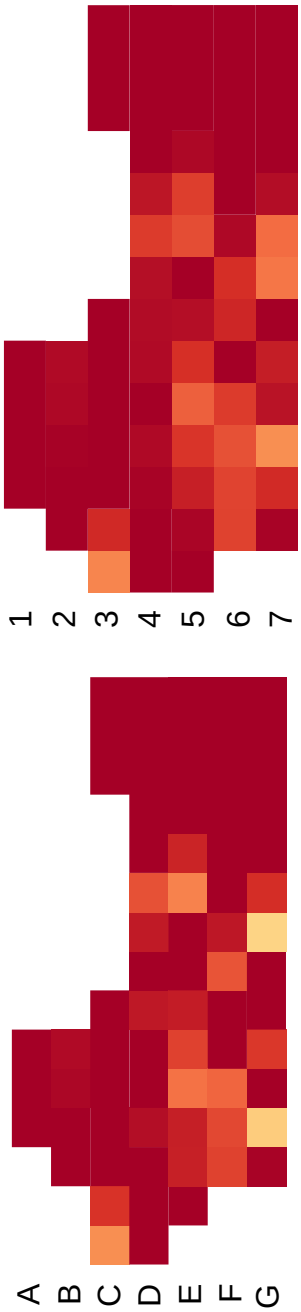
SAM_AP_3027_RMand_Ext



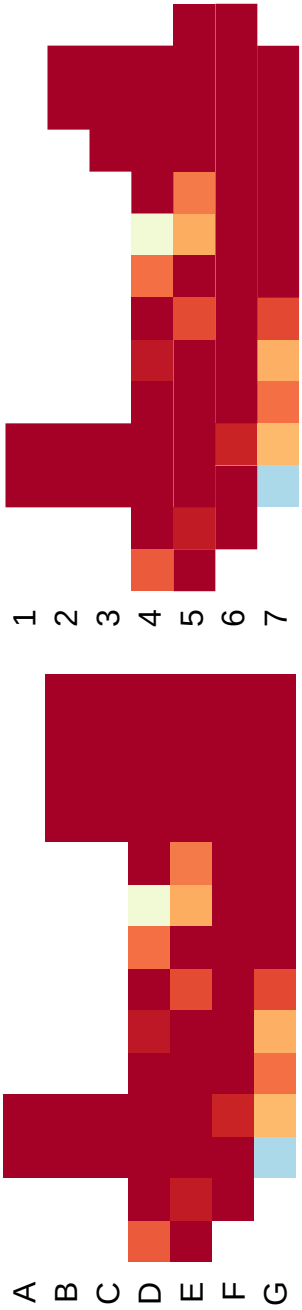
SAM_AP_3027_RMand_Int



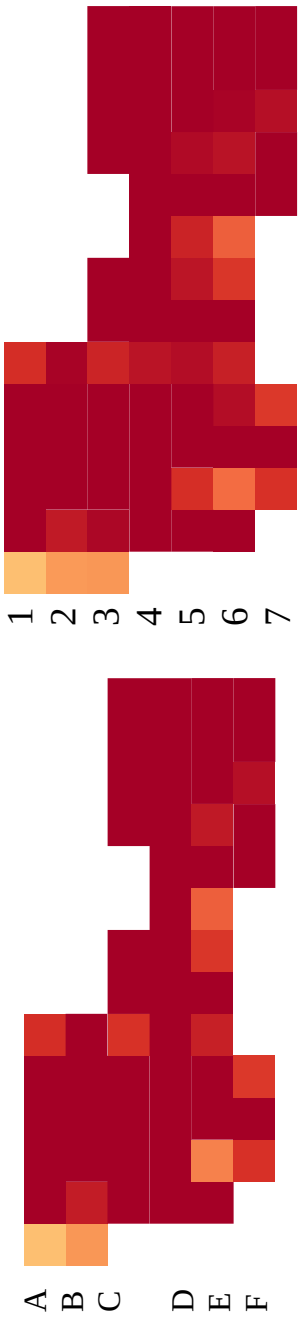
SAM_AP_3691_RMand_Ext



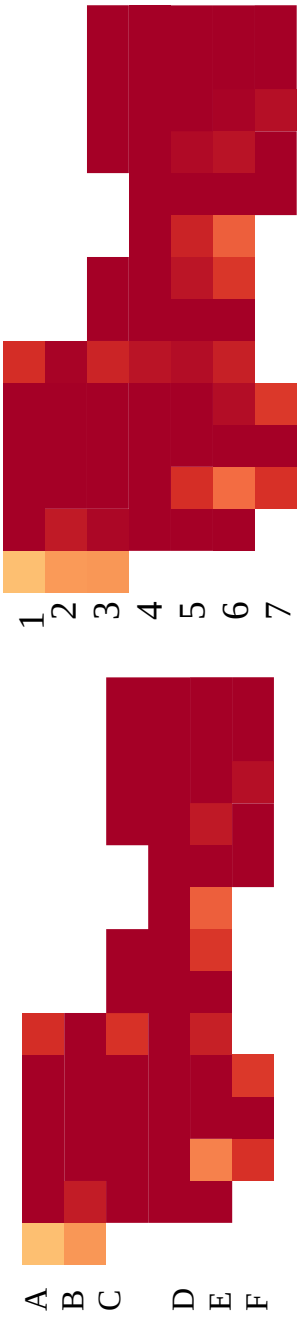
SAM_AP_3691_RMand_Int



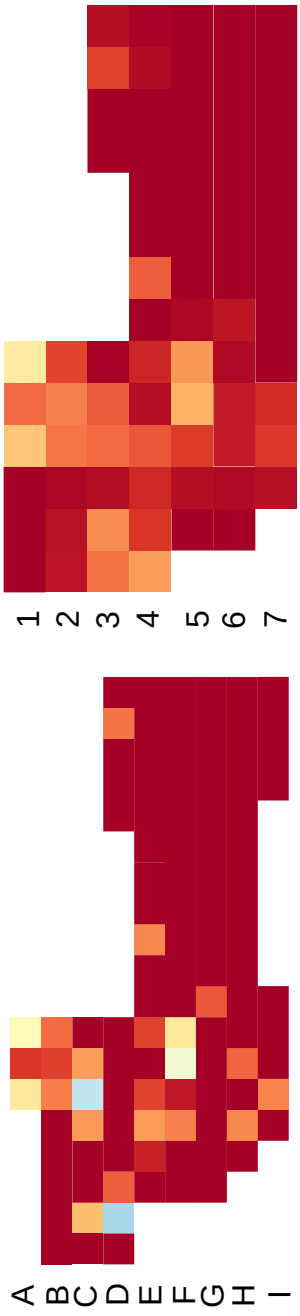
SAM_AP_3737a_RMand_Ext



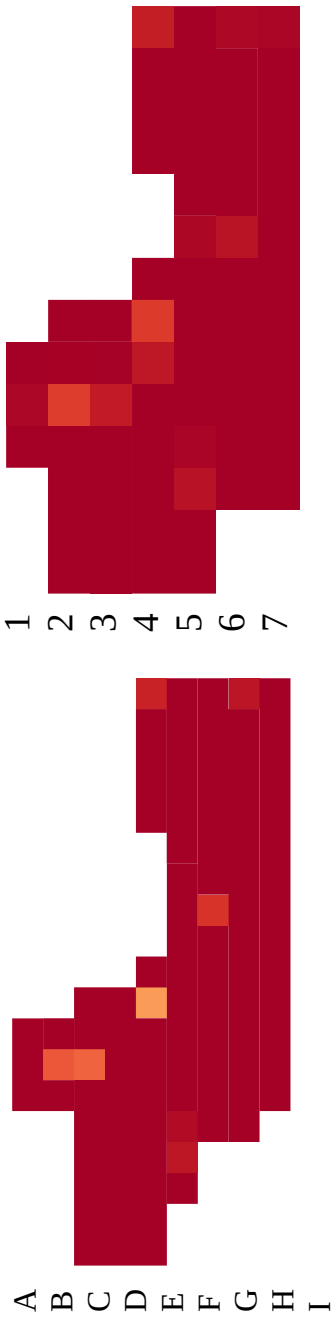
SAM_AP_3737a_RMand_Int



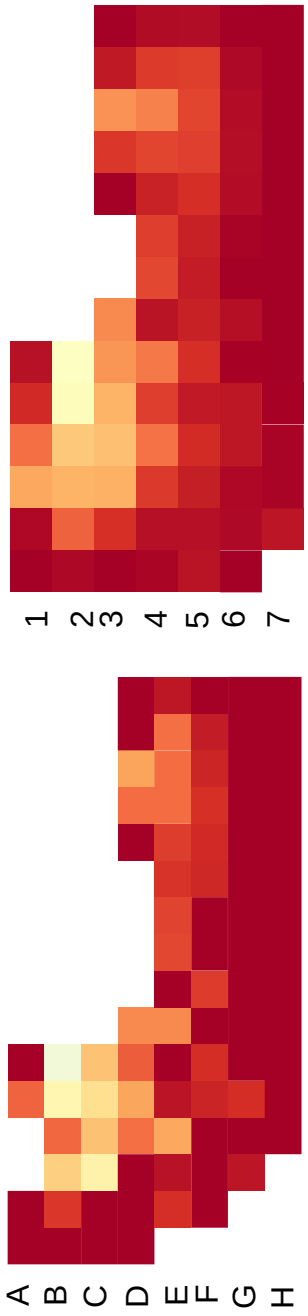
SAM_AP_4790_RMand_Ext



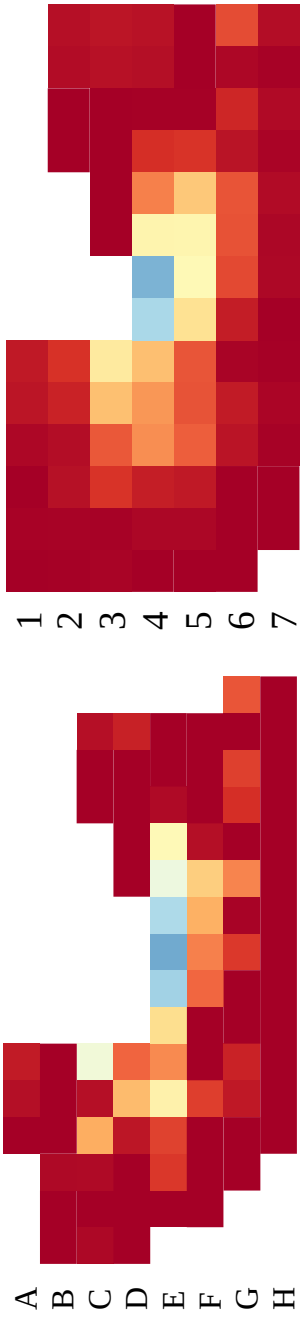
SAM_AP_4790_RMand_Int



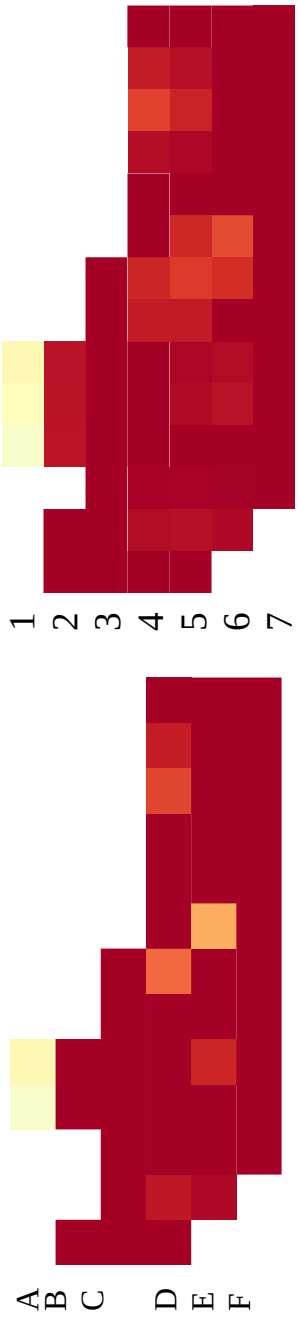
SAM_AP_4844_RMand_Ext



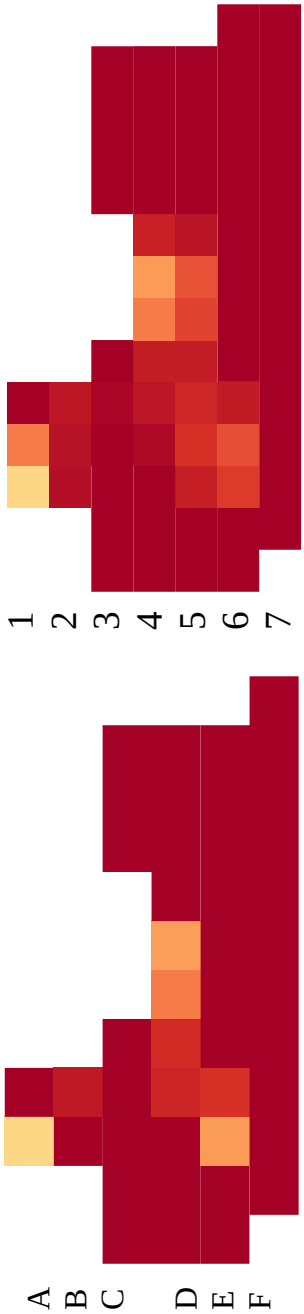
SAM_AP_4844_RMand_Int



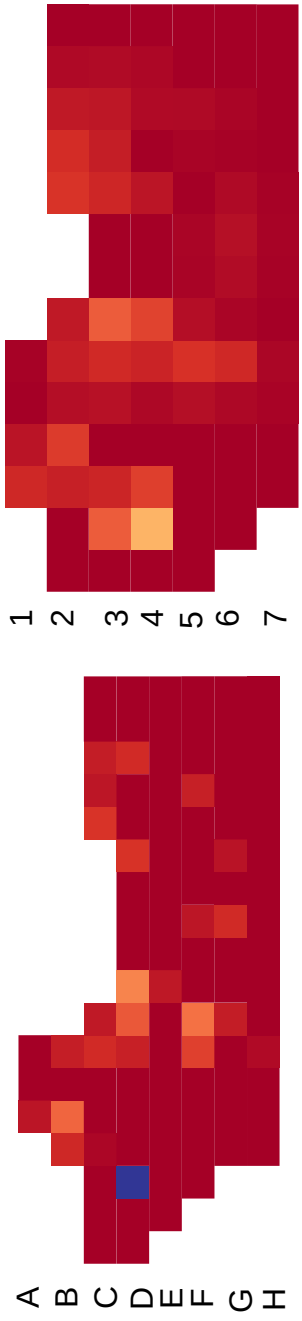
SAM_AP_6052_RMand_Ext



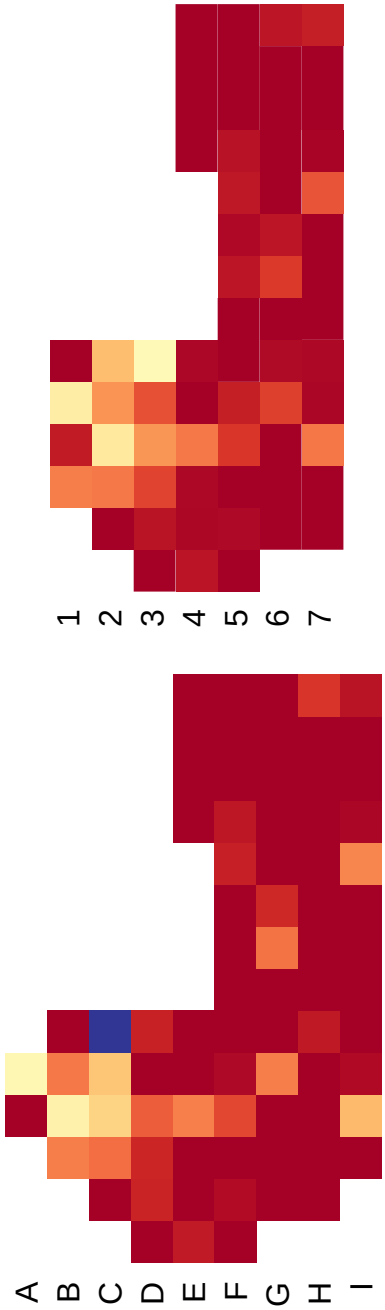
SAM_AP_6052_RMand_Int



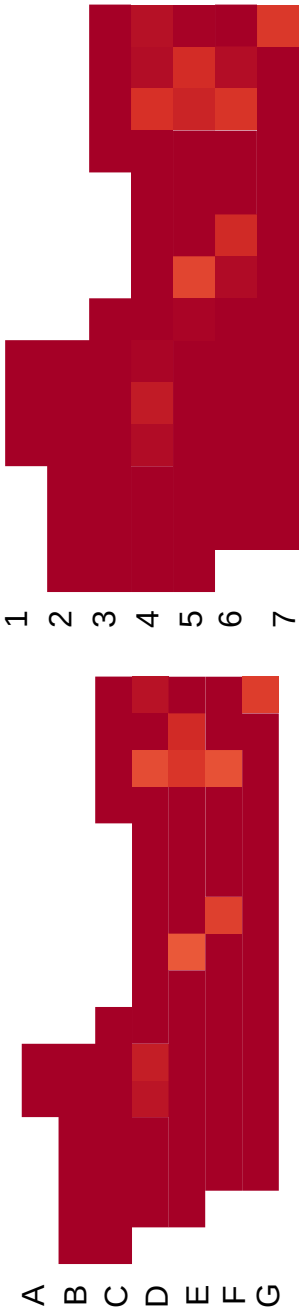
SAM_AP_6340_RMand_Ext



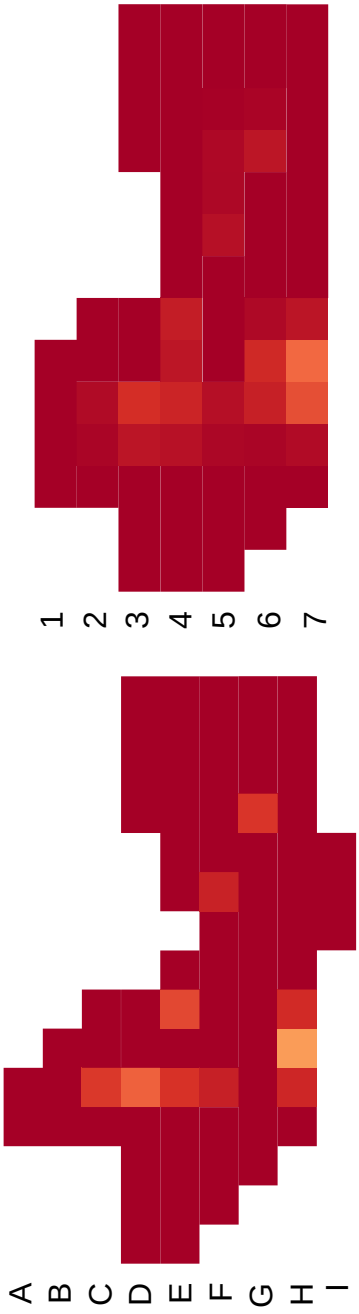
SAM_AP_6340_RMand_Int



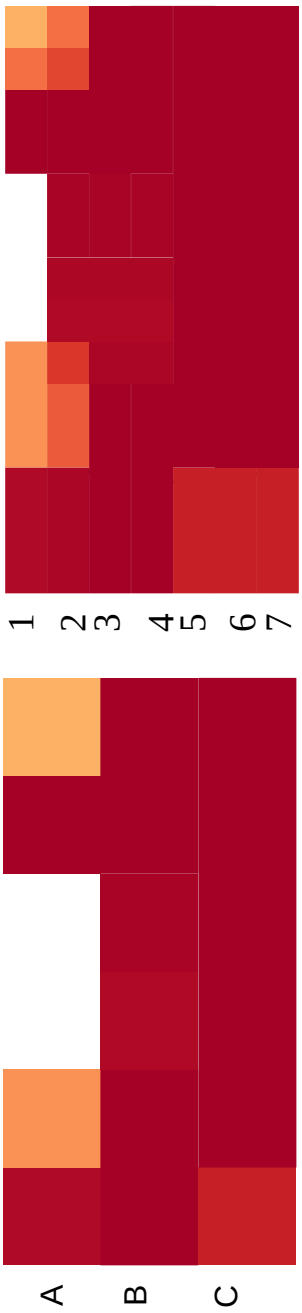
SAM_AP_6348a_RMand_Ext



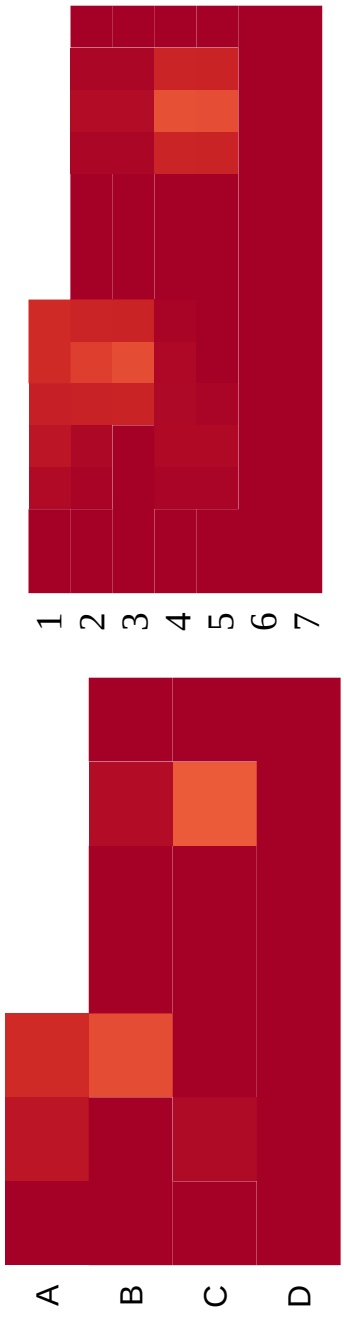
SAM_AP_6348a_RMand_Int



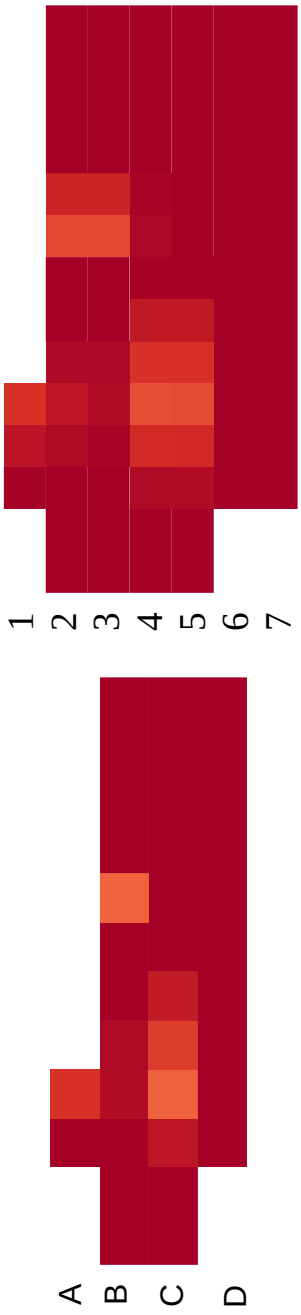
UCT_189_RMand_Ext



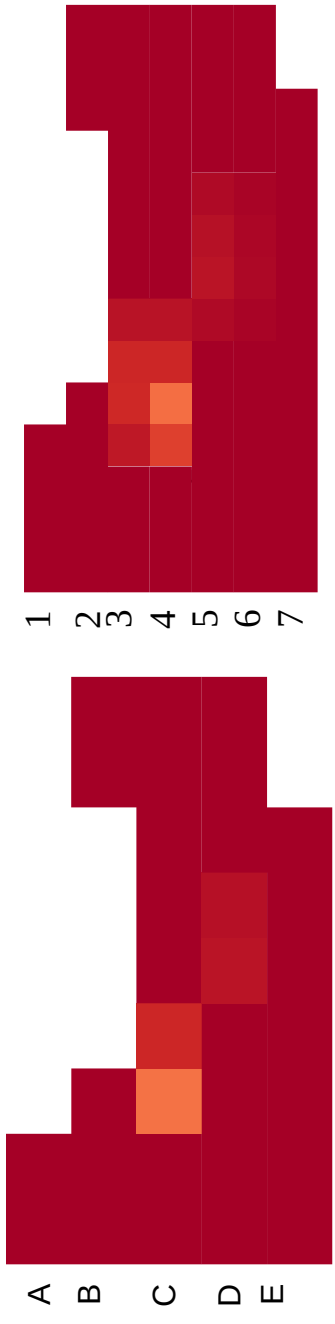
UCT_189_RMand_Int



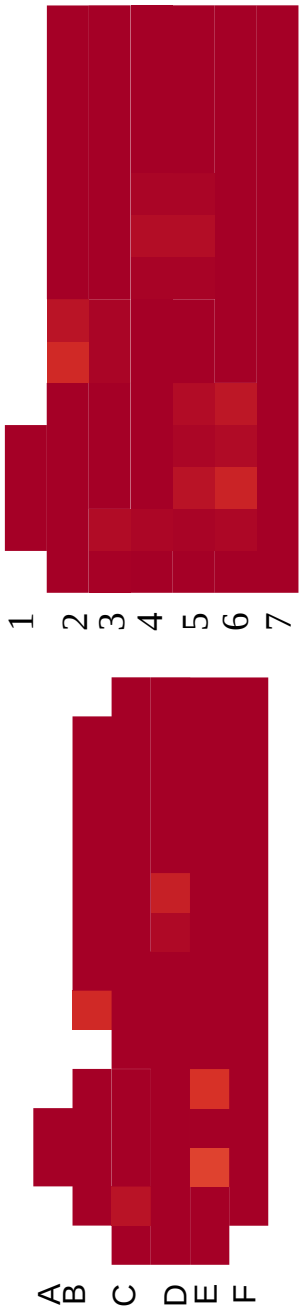
UCT_195_RMand_Ext



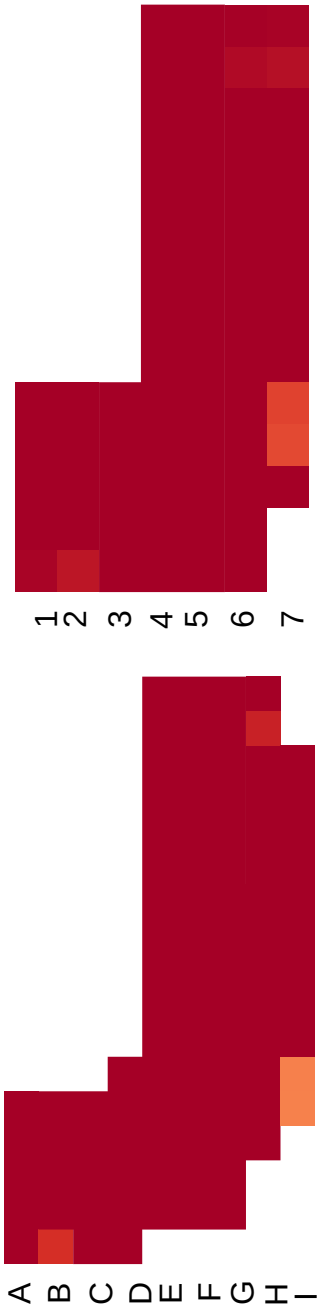
UCT_195_RMand_Int



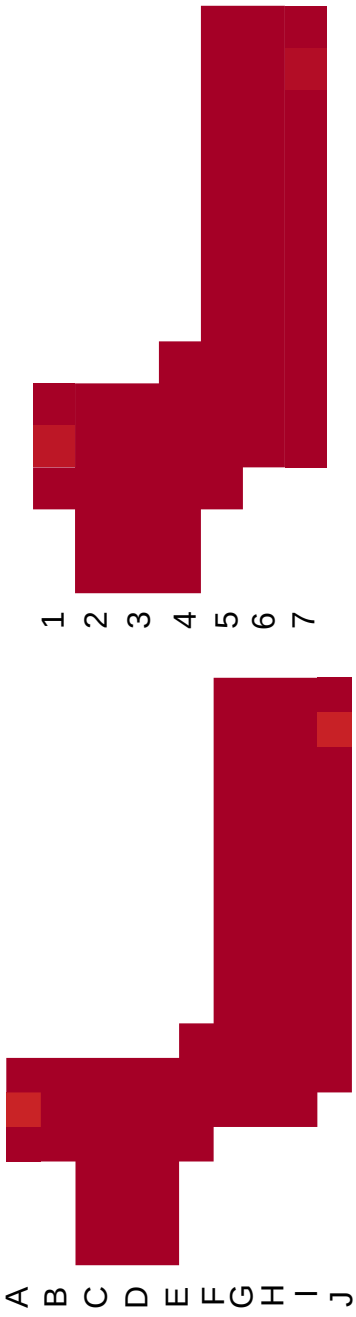
ULAC_131_199_RMand_Ext



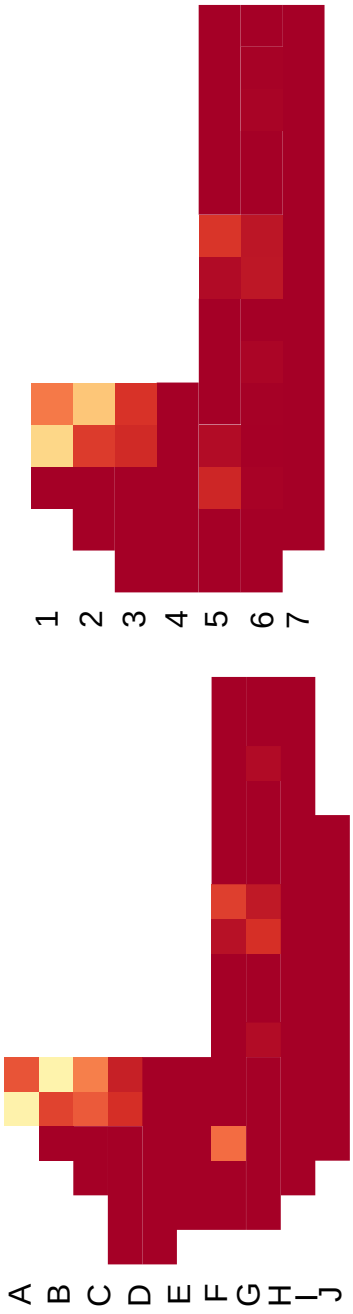
ULAC_131_199_RMand_Int



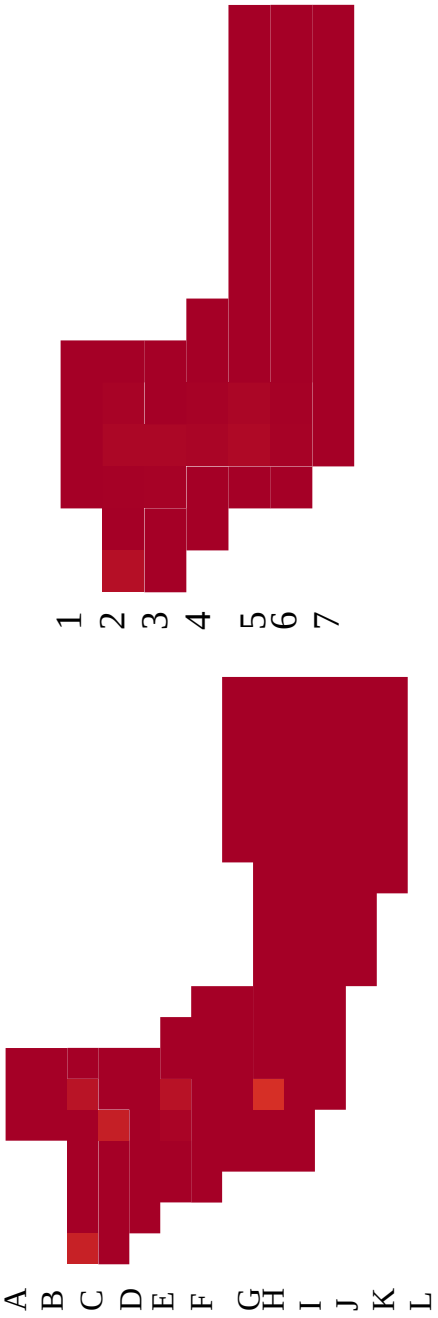
ULAC_186_201_RMand_Ext



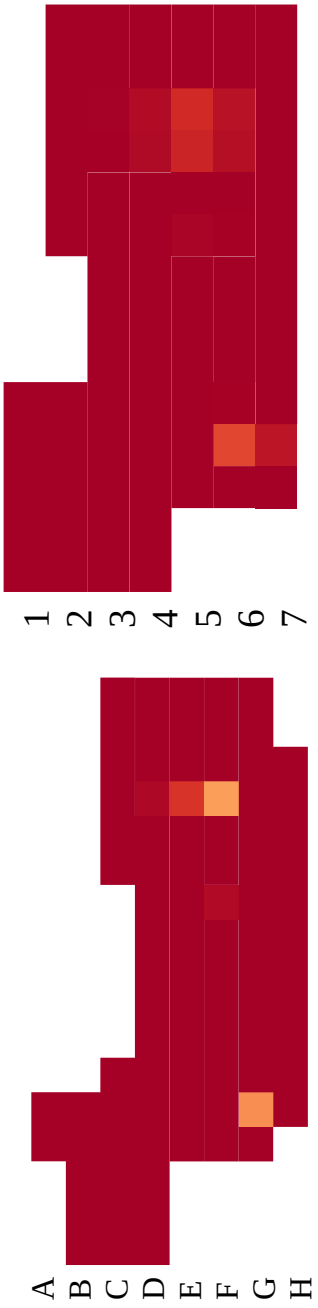
ULAC_186_201_RMand_Int



ULAC_201_12_RMand_Ext

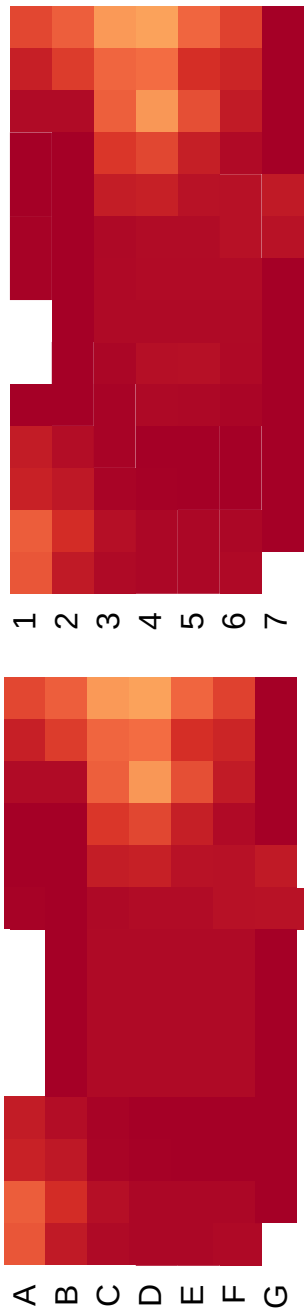


ULAC_201_12_RMand_Ext

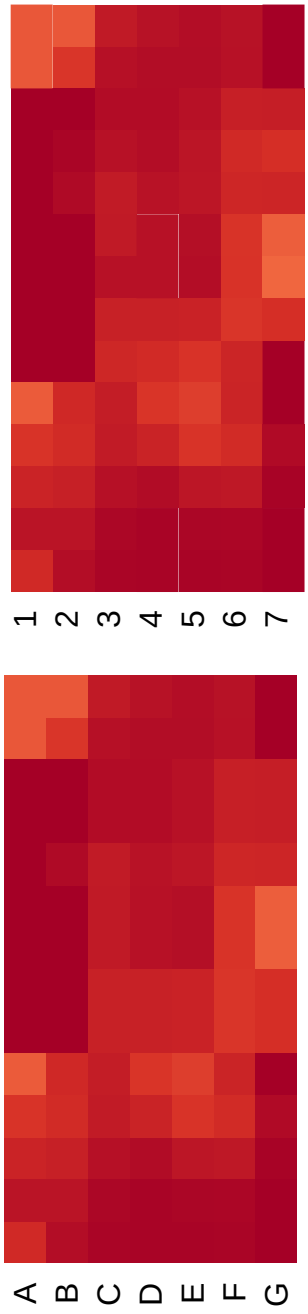


APPENDIX G MEAN MAPS

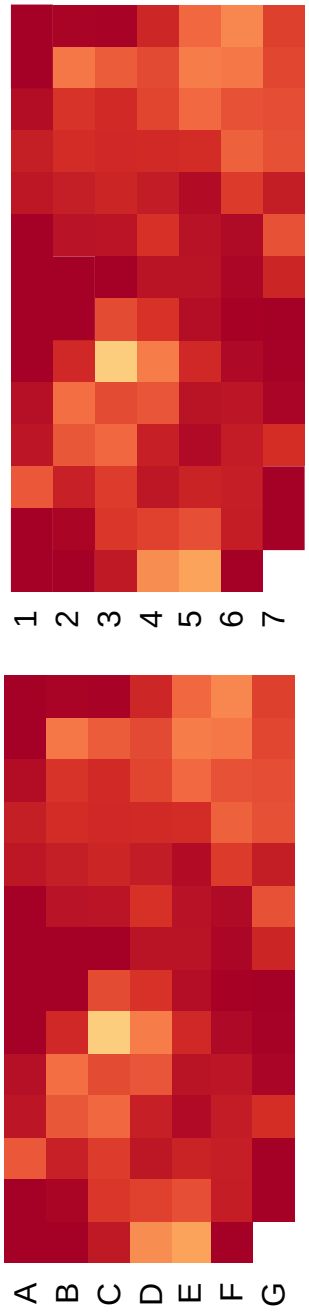
KAL_AG1_RMand_Ext_



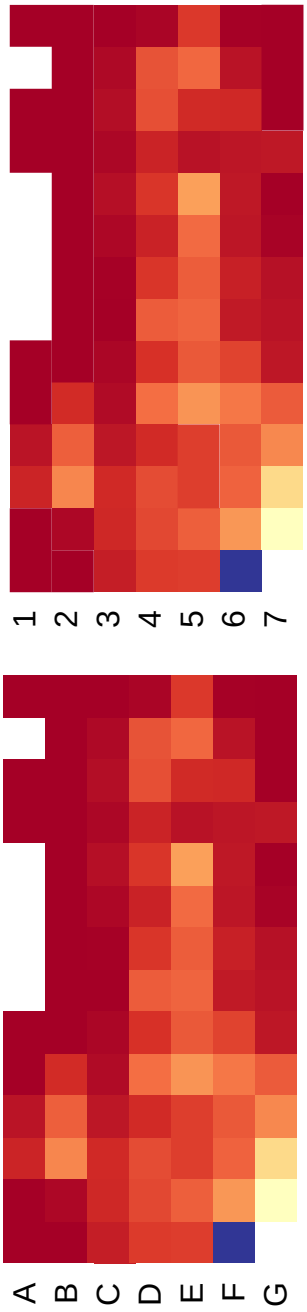
KAL_AG1_RMand_Int_



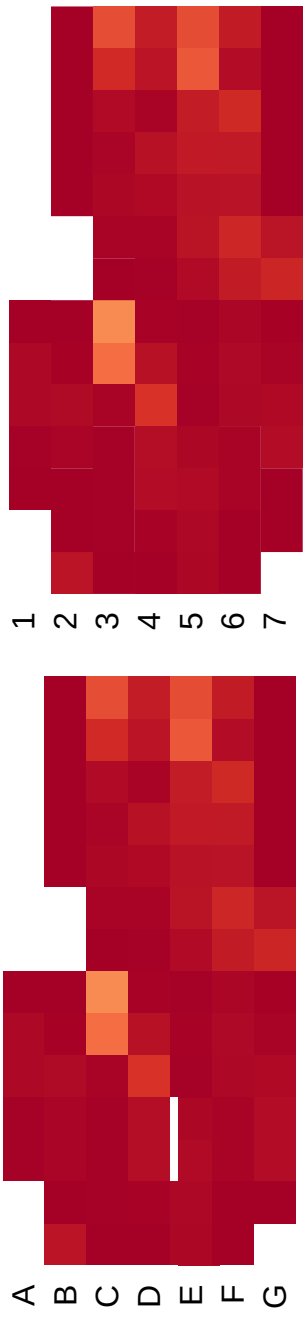
KAL_AG2_RMand_Ext_



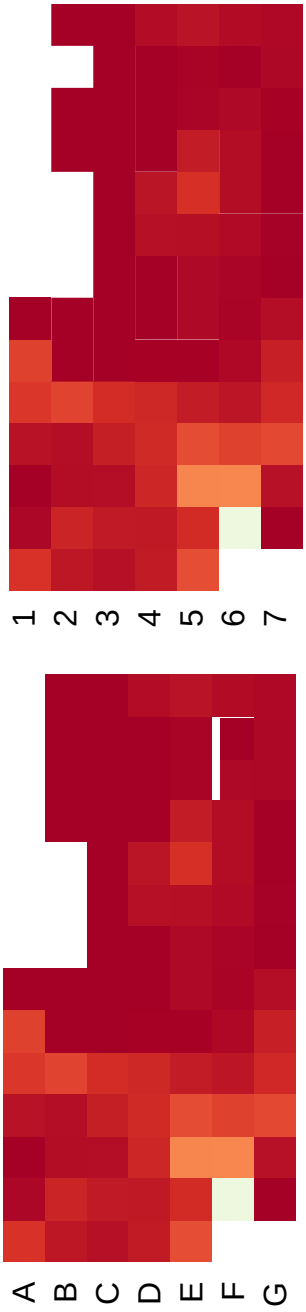
KAL_AG2_RMand_Int_



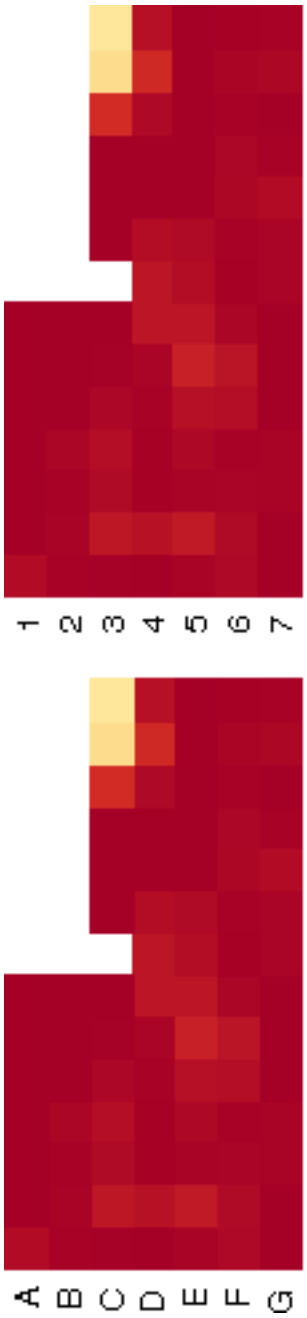
KAL_AG3_RMand_Ext_



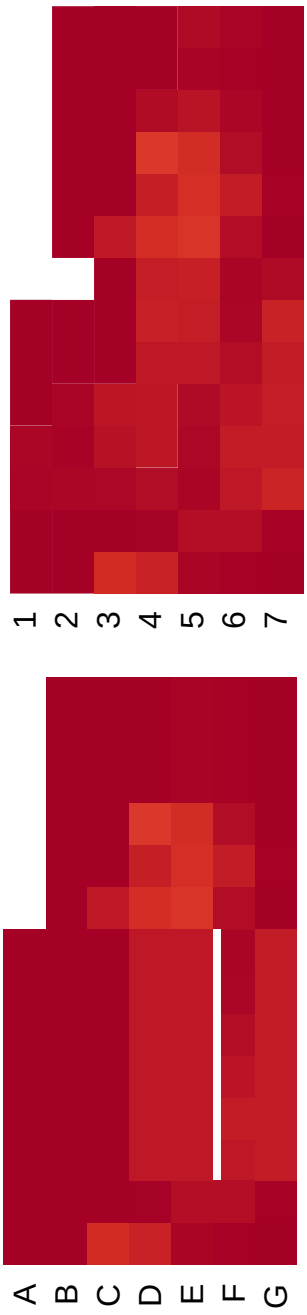
KAL_AG3_RMand_Int_



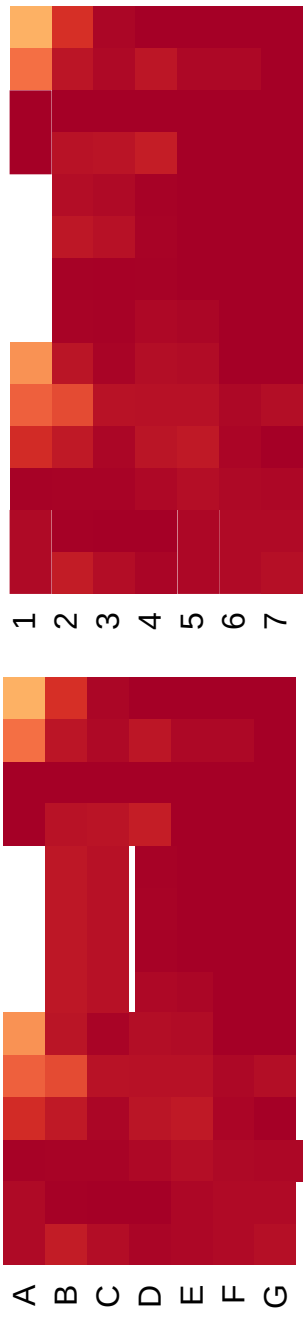
KAL_AG4_RMand_Ext_



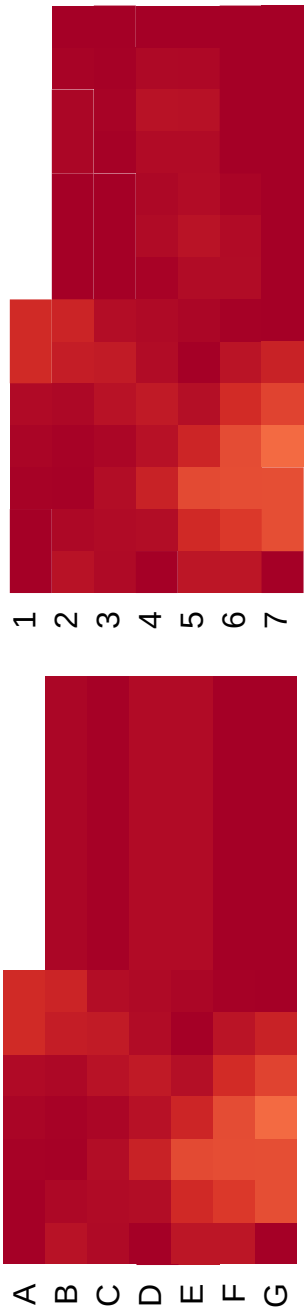
KAL_AG4_RMand_Int_



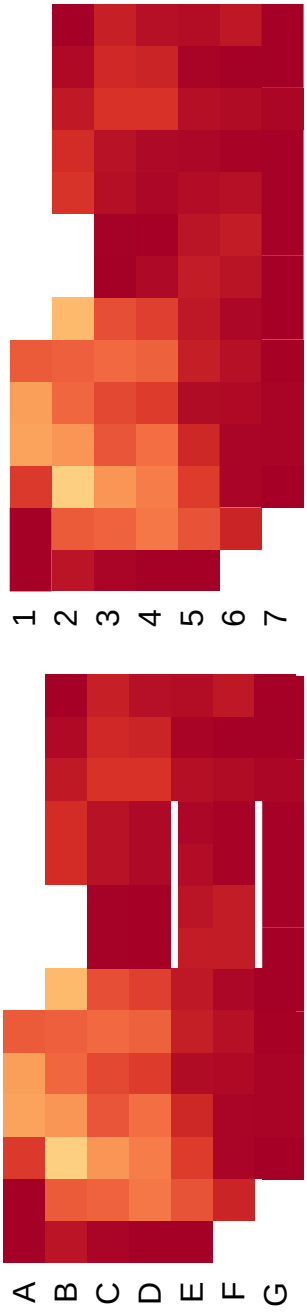
SA_AG1_RMand_Ext_



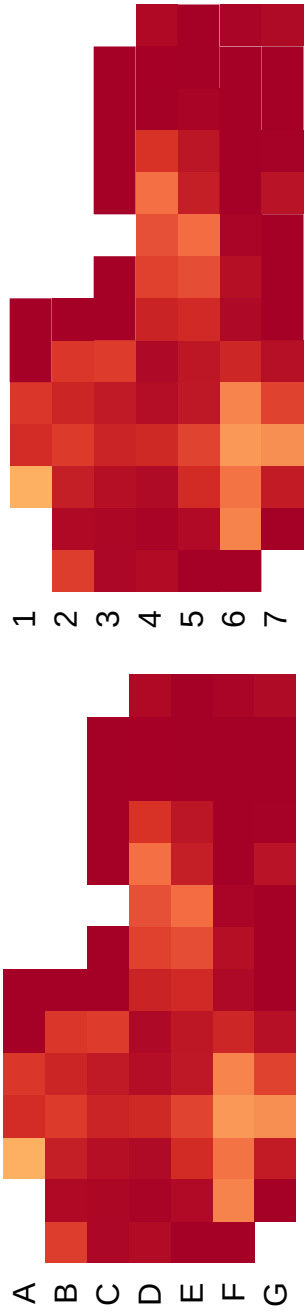
SA_AG1_RMand_Int_



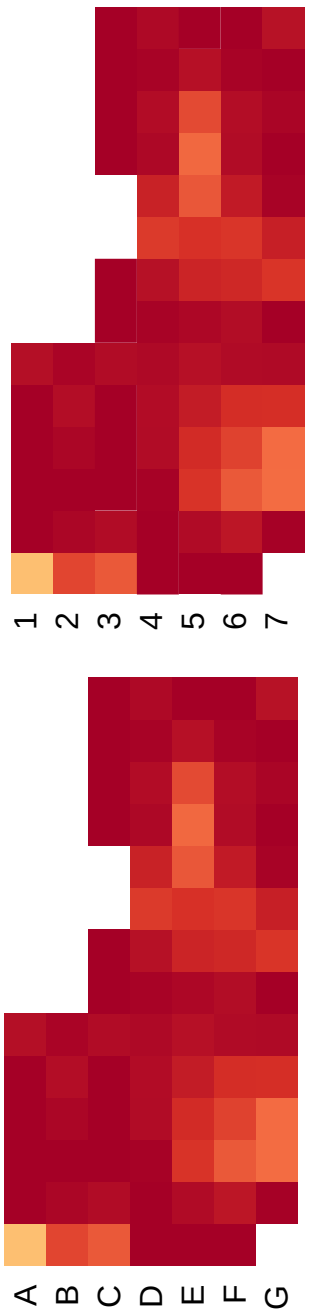
SA_AG2_RMand_Ext_



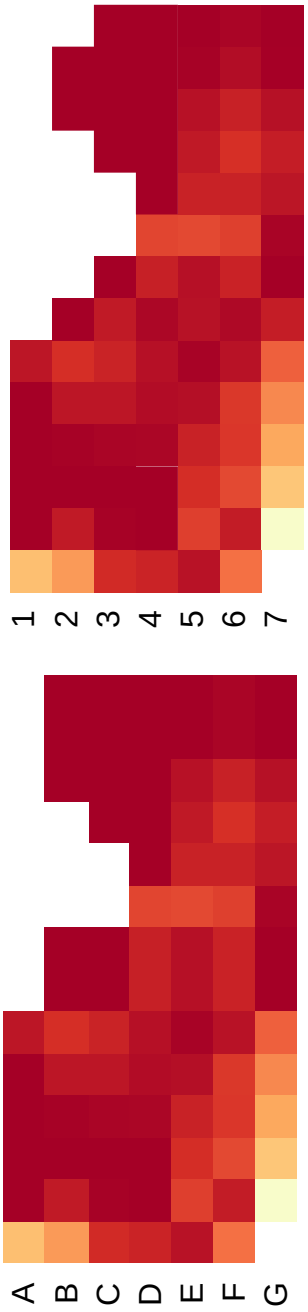
SA_AG2_RMand_Int_



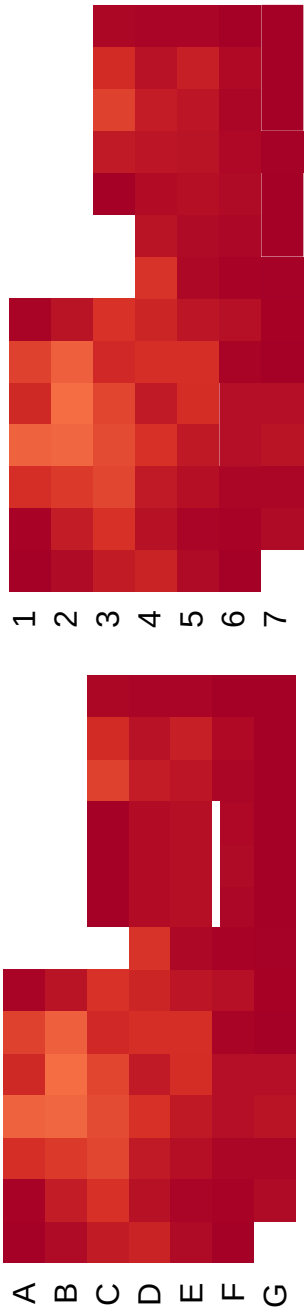
SA_AG3_RMand_Ext_



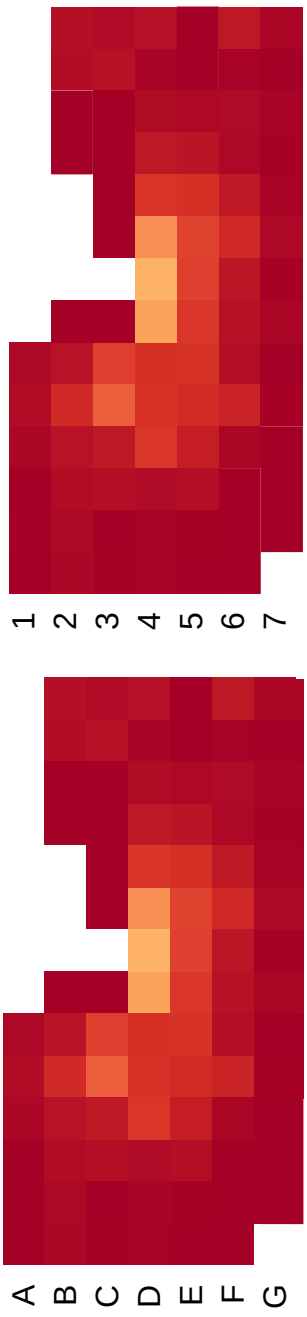
SA_AG3_RMand_Int_



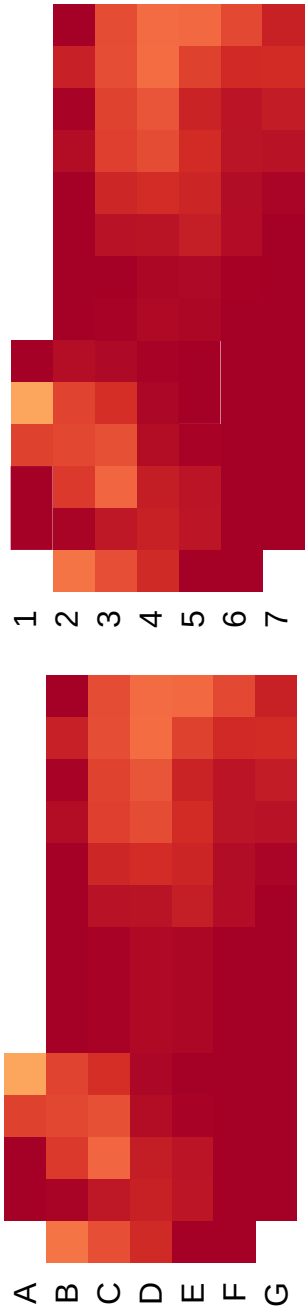
SA_AG4_RMand_Ext_



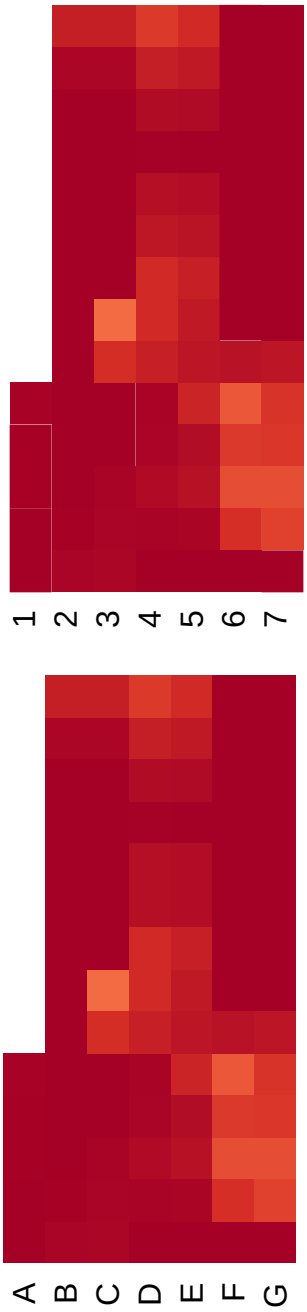
SA_AG4_RMand_Int_



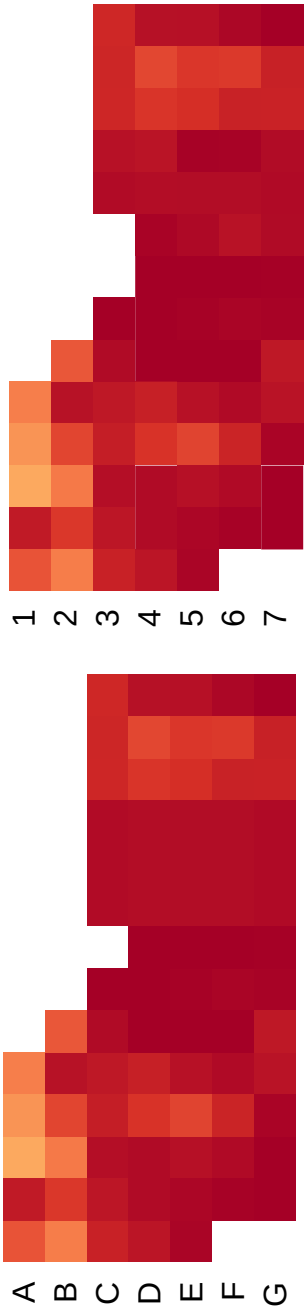
WE_AG1_RMand_Ext_



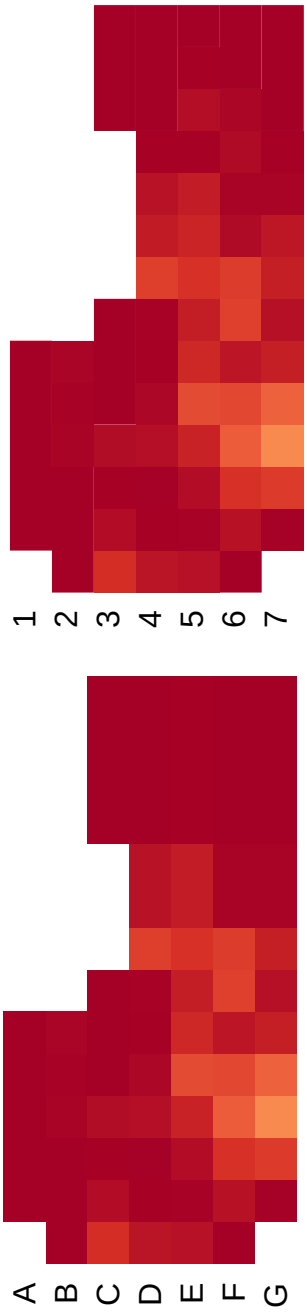
WE_AG1_RMand_Int_



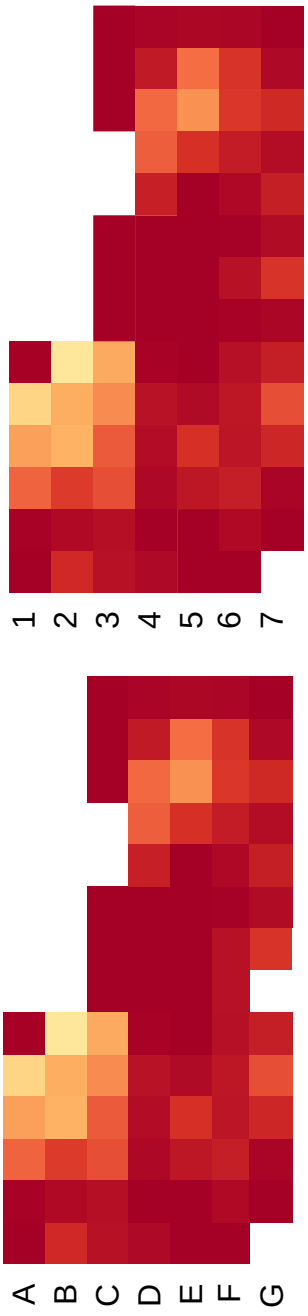
WE_AG2_RMand_Ext_



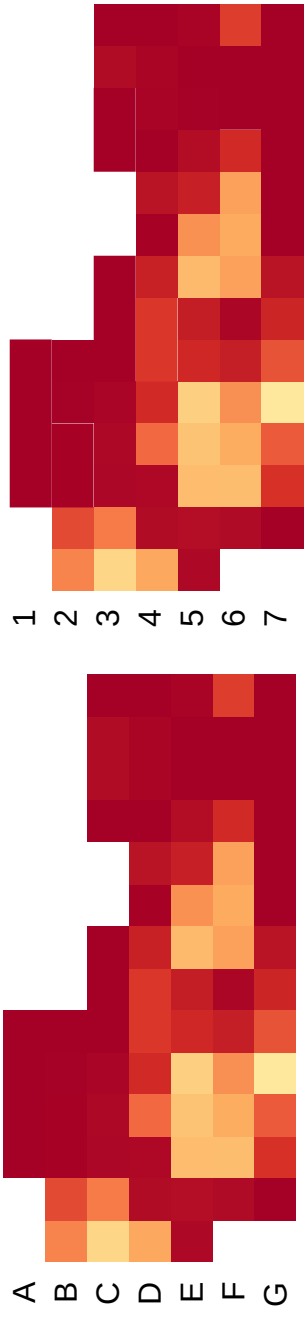
WE_AG2_RMand_Int_



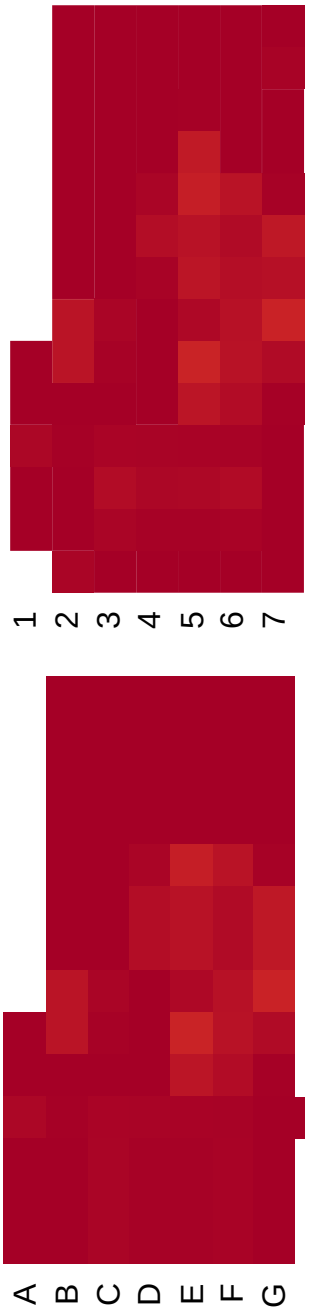
WE_AG3_RMand_Ext_



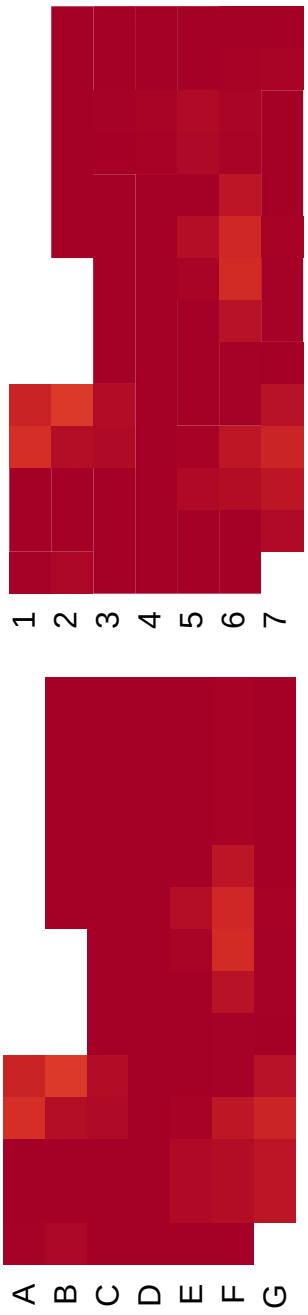
WE_AG3_RMand_Int_



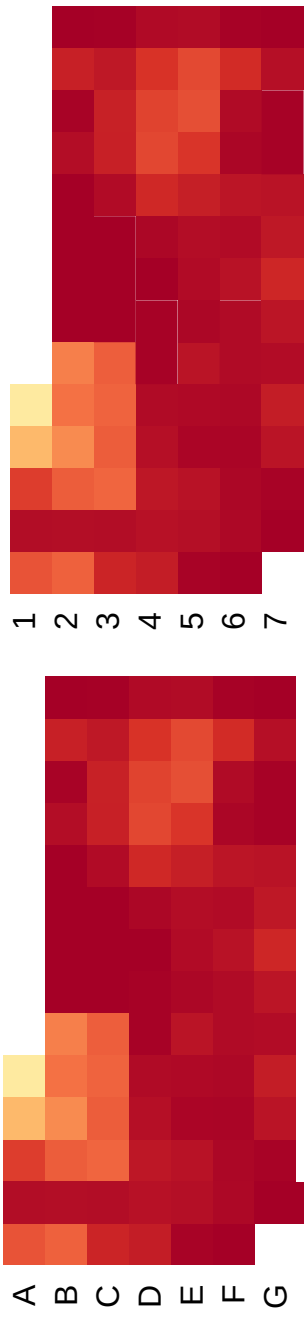
WE_AG4_RMand_Ext_



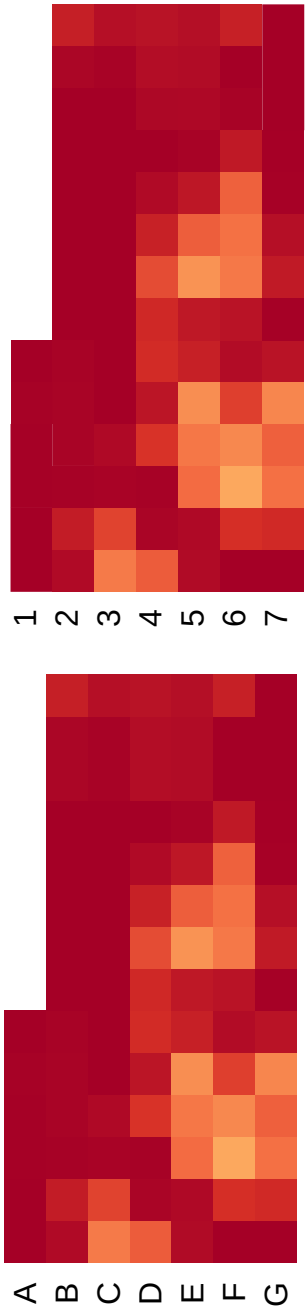
WE_AG4_RMand_Int_



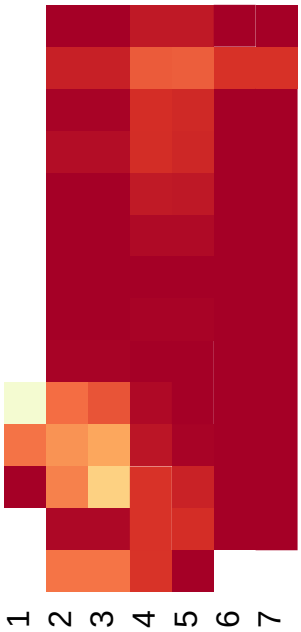
Male_RMand_Ext_

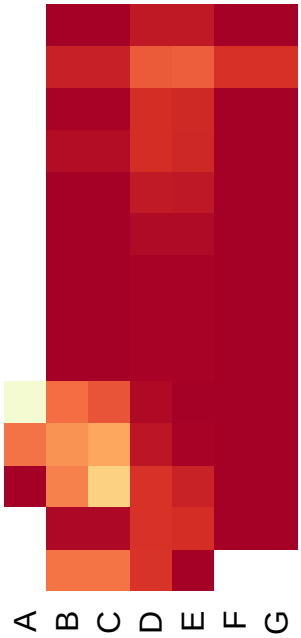


Male_RMand_Int_

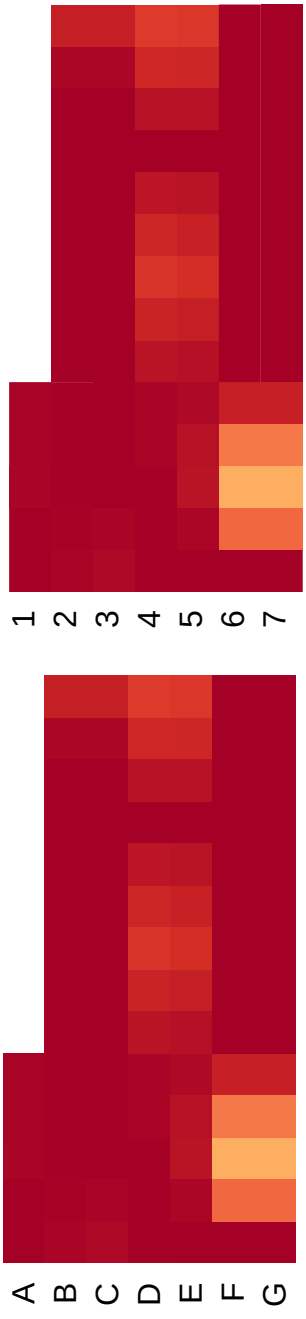


Male_AG1_RMand_Ext_



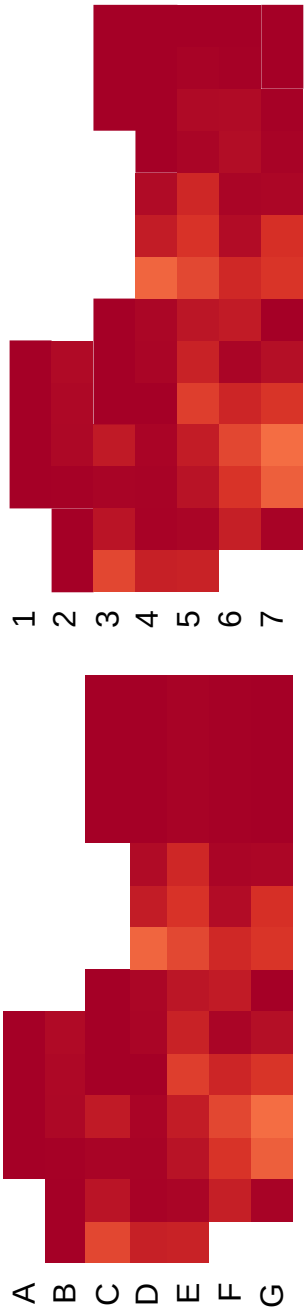


Male_AG1_RMand_Int_

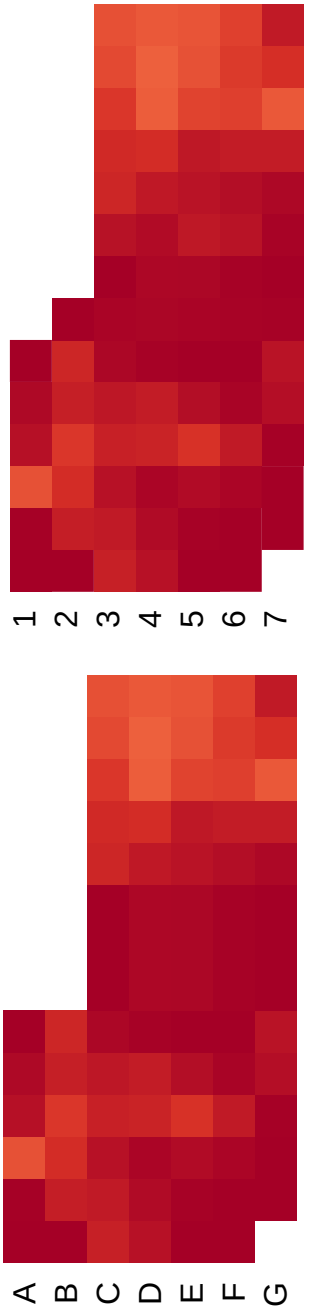


Male_AG2_RMand_Ext_

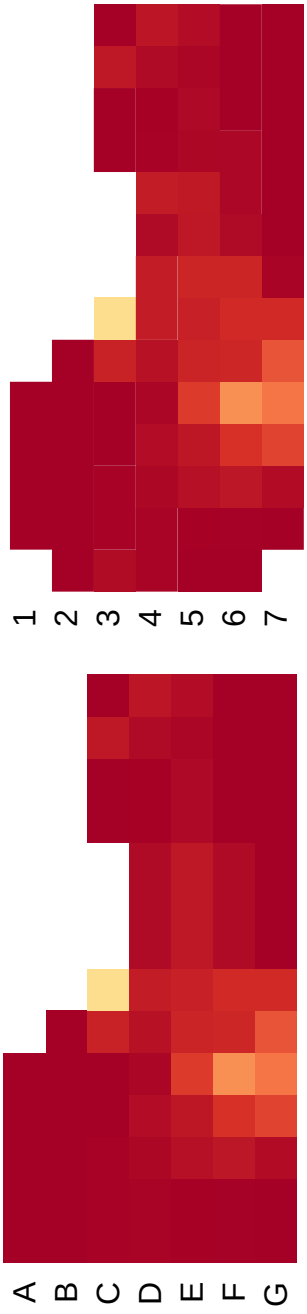
Male_AG2_RMand_Int_



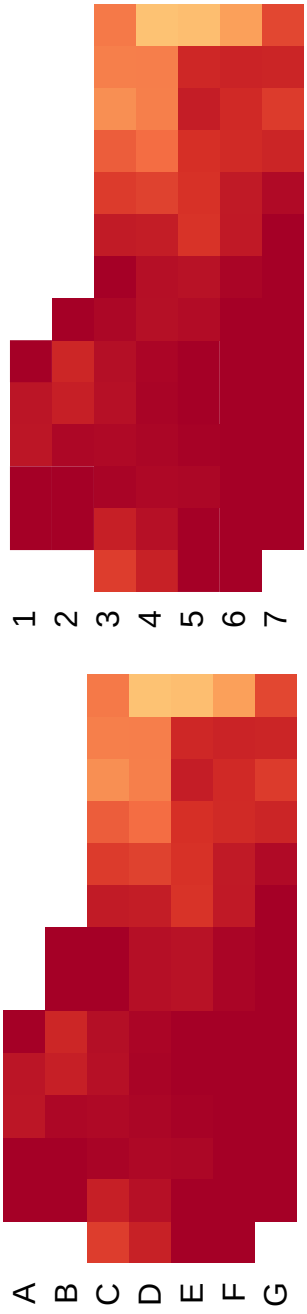
Female_RMand_Ext_



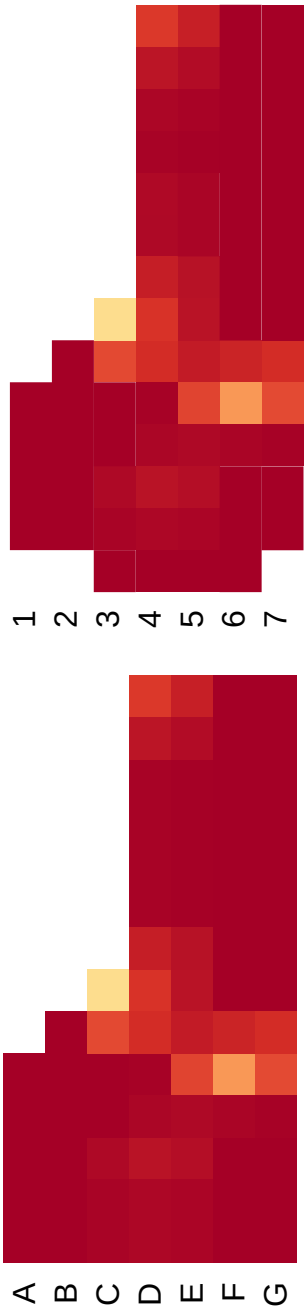
Female_RMand_Int_



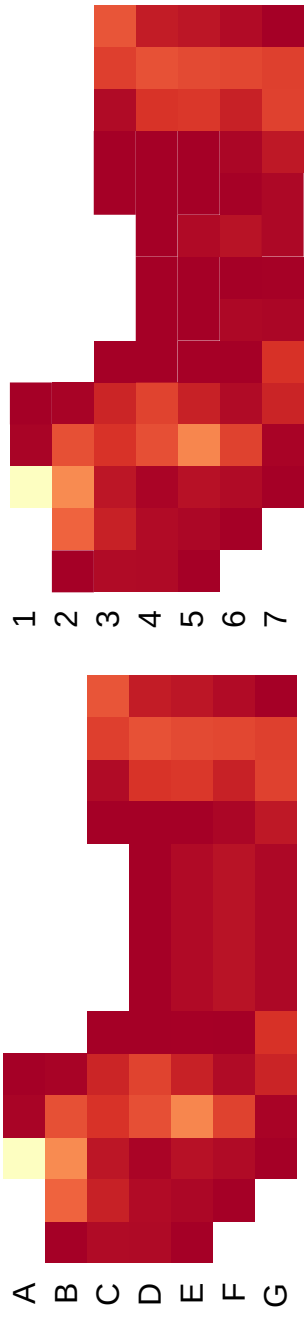
Female_AG1_RMand_Ext_



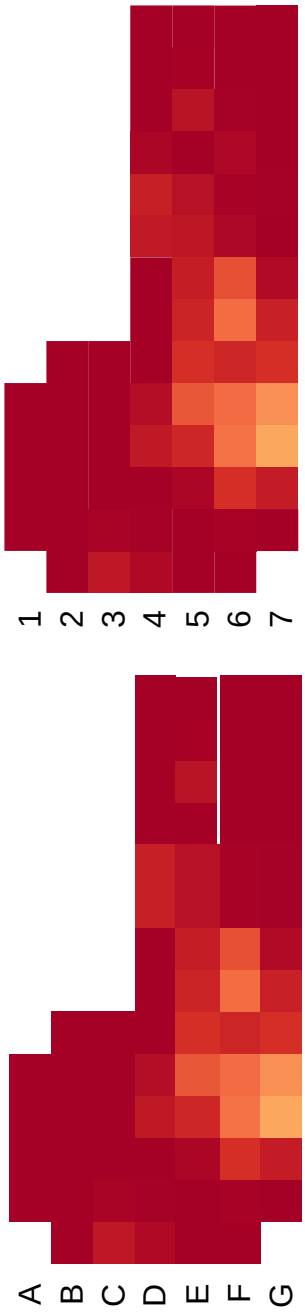
Female_AG1_RMand_Int_



Female_AG2_RMand_Ext_



Female_AG2_RMand_Int_



APPENDIX H R CODING

Standardized Grids
2023-02-28

```
##############################################################################
##############################################
##############################################################################
##############################################
####### MAPPING
```

```
library(ggplot2)
library(scales)
library(readr)

##
## Attaching package: 'readr'

## The following object is masked from 'package:scales':
##
##     col_factor

library(RColorBrewer)
library(fields)

## Loading required package: spam

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Attaching package: 'viridis'
```

```
## The following object is masked from 'package:scales':
##
##     viridis_pal

##
## Try help(fields) to get started.

library(raster)

## Loading required package: sp

library(colorRamps)
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4

library(proxy)

##
## Attaching package: 'proxy'

## The following object is masked from 'package:raster':
##
##     as.matrix

## The following object is masked from 'package:spam':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

#library(as.color)
```

###1. TRANSFORM DATA INTO GRIDS

```
library(polyclip)

## polyclip 1.10-4 built from Clipper C++ version 6.4.0
```

```
library(splancs)

##
## Spatial Point Pattern Analysis Code in S-Plus
##
##  Version 2 - Spatial and Space-Time analysis

##
## Attaching package: 'splancs'

## The following object is masked from 'package:raster':
##
##     zoom
```

###1.1. All maps

Homo sapiens

```
#All individual Maps

flist=list.files(path="~//Desktop//Mapping//Raw Data Abstract", full.names=T, recursive=T, patt
ern=".txt")      #indicate the path where the txt files for each individuals are
source("~//Desktop//Mapping//Scripts//grid_transform.r")      #indicate the path of the source co
de
res.mat=matrix(NA, ncol=2, nrow=length(flist))
for(i in 1:length(flist)){
  xdata=read.table(file=flist[i], header=T, sep="\t", fill=T)
  xdata$Mapping=as.character(xdata$Mapping)
  xdata$y=substr(xdata$Mapping, start=1, stop=1)
  xdata$y=match(xdata$y, rev(LETTERS))
  xdata$x=as.numeric(substr(xdata$Mapping, start=2, stop=nchar(xdata$Mapping)))
  res.mat[i, 1]=diff(range(xdata$x))
  res.mat[i, 2]=diff(range(xdata$y))
}
hbw1=0.5
corners1=cbind(x=c(-hbw1, hbw1, hbw1, -hbw1, -hbw1), y=c(-hbw1, -hbw1, hbw1, hbw1, -hbw
1))
new.grid.resol1=14
new.grid.resol2=7
ramp=colorRamp(colors = brewer.pal(11, "RdYlBu"))
i=grep(flist, pattern= "6988", fixed=T)


#8x8 grid map for all specimens

main_dir <- "~//Desktop//Mapping"
```

```
import_dir <- "~//Desktop//Mapping//Raw Data Abstract//"
setwd(main_dir)
dir.create("Transformed")

## Warning in dir.create("Transformed"): 'Transformed' already exists

dir.create("Maps")          #creates a new folder where all the maps will be saved as PDF

## Warning in dir.create("Maps"): 'Maps' already exists

export_dir_transf <- paste(main_dir, "Transformed", sep = "")
export_dir <- paste(main_dir, "Maps", sep = "")

for(i in 1:length(flist)){
  xdata=read.table(file=flist[i], header=T, sep="\t", fill=T)
  xdata=subset(xdata, !is.na(Percent))
  xdata$Mapping=as.character(xdata$Mapping)
  xdata$y=substr(xdata$Mapping, start=1, stop=1)
  xdata$y=match(xdata$y, rev(LETTERS))
  xdata$y=xdata$y-min(xdata$y)+1
  xdata$x=as.numeric(substr(xdata$Mapping, start=2, stop=nchar(xdata$Mapping)))
  xdata$x=xdata$x-min(xdata$x)+1
  grid.1=lapply(1:nrow(xdata), function(x){
    t(t(corners1)+unlist(xdata[x, c("x", "y")]))
  })
  hbw2=(diff(range(xdata$x))+1)/new.grid.resol1
  hbh2=(diff(range(xdata$y))+1)/new.grid.resol2
  grid.2=data.frame(
    expand.grid(
      x=seq(hbw2, hbw2*14, length.out=new.grid.resol1),
      y=seq(hbh2, hbh2*7, length.out=new.grid.resol2)
    )
  )
  hbw2=hbw2/2
  hbh2=hbh2/2
  corners2=cbind(x=c(-hbw2, hbw2, hbw2, -hbw2, -hbw2), y=c(-hbh2, -hbh2, hbh2, hbh2, -hbh2
))
  grid.2b=lapply(1:nrow(grid.2), function(x){
    t(t(corners2)+unlist(grid.2[x, c("x", "y")]))
  })
  res=grid.transform(grid.from.cells=grid.1, grid.to.cells=grid.2b, grid.from.values=xdata$Percen
t)
  res=data.frame(grid.2, value=res)
  #fname=paste(c(substr(flist[i], start=1, stop=nchar(flist[i])-4), "_transf.txt"), collapse="")
  fname=gsub(".txt","",list.files(import_dir))
  res$x=seq(1, new.grid.resol1, by=1)[match(res$x, sort(unique(res$x)))]
  res$y=seq(1, new.grid.resol2, by=1)[match(res$y, sort(unique(res$y)))]
  setwd("~/Desktop/Mapping/Transformed")
```

```
  write.table(x=res, file=paste(fname[i],"_transf.txt",sep=""), row.names=F, col.names=T, sep="\
t")

  setwd("~/Desktop/Mapping/Maps")
  pdf(file=paste(fname[i],".pdf",sep=""))

  xcol=ramp(xdata$Percent/100)
  par(mar=c(2, 2, 0.2, 0.2), mfrow=c(1, 2))
  plot(x=xdata$x, y=xdata$y, xlab="", ylab="", asp=1, xaxt="n", yaxt="n", type="n", bty="n",
      xlim=range(xdata$x)+hbw1*c(-1, 1), ylim=range(xdata$y)+hbw1*c(-1, 1), xaxs="i", yaxs="
i")
  rect(xleft=xdata$x-hbw1, xright=xdata$x+hbw1, ybottom=xdata$y-hbw1, ytop=xdata$y+hbw1
,
      col=rgb(red=xcol, maxColorValue=255), border=rgb(red=xcol, maxColorValue=255))
  mtext(text=sort(unique(xdata$x)), at=sort(unique(xdata$x)), side=1, line=0.5)
  mtext(text=LETTERS[sort(unique(xdata$y))], at=sort(unique(xdata$y)), side=2, line=0.5, las=
1)

  res$value[res$value>100]=100
  xcol=ramp(res$value/100)

  plot(x=res$x, y=res$y, xlab="", ylab="", asp=1, xaxt="n", yaxt="n", type="n", bty="n",
      xlim=range(res$x)+hbw1*c(-1, 1), ylim=range(res$y)+hbw1*c(-1, 1), xaxs="i", yaxs="i")
  rect(xleft=res$x[!is.na(xcol[, 1])]-hbw1, xright=res$x[!is.na(xcol[, 1])]+hbw1,
      ybottom=res$y[!is.na(xcol[, 1])]-hbw1, ytop=res$y[!is.na(xcol[, 1])]+hbw1,
      col=rgb(red=xcol[!is.na(xcol[, 1]), ], maxColorValue=255),
      border=rgb(red=xcol[!is.na(xcol[, 1]), ], maxColorValue=255))
  mtext(text=sort(unique(res$x)), at=sort(unique(res$x)), side=1, line=0.5)
  mtext(text=round(sort(unique(res$y)), 1), at=sort(unique(res$y)), side=2, line=0.5, las=1)

  dev.off()

}
```

Western European Mean Maps
2023-02-28

```
library(ggplot2)
library(scales)
library(readr)

##
## Attaching package: 'readr'

## The following object is masked from 'package:scales':
##
##     col_factor

library(RColorBrewer)
library(fields)

## Loading required package: spam

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':
##
##     viridis_pal

##
## Try help(fields) to get started.

library(raster)

## Loading required package: sp
```

```
library(colorRamps)
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4

library(proxy)

##
## Attaching package: 'proxy'

## The following object is masked from 'package:raster':
##
##     as.matrix

## The following object is masked from 'package:spam':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

library(fields)
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/WE AG1 Ext")


#List all txt files
txtfile_list_WE_AG1_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_WE_AG1_Ext<- do.call("cbind", lapply(txtfile_list_WE_AG1_Ext, FUN = function(fi
```

```
les){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG1_Ext)){
  All_data_WE_AG1_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG1_Ext <- gsub("_transf.txt", "", txtfile_list_WE_AG1_Ext)
colnames(All_data_WE_AG1_Ext) <- txtfile_list_WE_AG1_Ext


#Save data
write.table(All_data_WE_AG1_Ext, "~/Desktop/Mean Mapping/WE AG1 Ext/All_data_WE_A
G1_Ext.txt", sep = "\t")
```

2.2. Create mean maps per WE AG

WE AGE GROUP 1

```
#Select individuals
WE_AG1_Ext <- All_data_WE_AG1_Ext


#Calculate mean

Mean_WE_AG1_Ext <- c()
for(i in 1:nrow(WE_AG1_Ext)){
  mean1 <- mean(as.numeric(WE_AG1_Ext[i,]), na.rm = T)
  Mean_WE_AG1_Ext <- c(Mean_WE_AG1_Ext, mean1)
}
```

```
WE_AG1_Ext <- cbind(WE_AG1_Ext, Mean_WE_AG1_Ext)


#Save
write.table(WE_AG1_Ext[,5], "~/Desktop/Mean Mapping/WE AG1 Ext/Mean_WE_AG1_Ext.tx
t")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/WE AG1 Int")


#List all txt files
txtfile_list_WE_AG1_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.n
ames = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_WE_AG1_Int<- do.call("cbind", lapply(txtfile_list_WE_AG1_Int, FUN = function(fil
es){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG1_Int)){
  All_data_WE_AG1_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG1_Int <- gsub("_transf.txt", "", txtfile_list_WE_AG1_Int)
colnames(All_data_WE_AG1_Int) <- txtfile_list_WE_AG1_Int


#Save data
write.table(All_data_WE_AG1_Int, "~/Desktop/Mean Mapping/WE AG1 Int/All_data_WE_AG
1_Int.txt", sep = "\t")
```

```
###2.2. Create mean maps per WE AG



## WE AGE GROUP 1

#Select individuals
WE_AG1_Int <- All_data_WE_AG1_Int


#Calculate mean

Mean_WE_AG1_Int <- c()
for(i in 1:nrow(WE_AG1_Int)){
  mean1 <- mean(as.numeric(WE_AG1_Int[i,]), na.rm = T)
  Mean_WE_AG1_Int <- c(Mean_WE_AG1_Int, mean1)
}

WE_AG1_Int <- cbind(WE_AG1_Int, Mean_WE_AG1_Int)


#Save
write.table(WE_AG1_Int[,5], "~/Desktop/Mean Mapping/WE AG1 Int/  Mean_WE_AG1_Int.txt
")
```

```
###2.1. Create dataframe with all data



#Set wd
setwd("~/Desktop/Mean Mapping/WE AG2 Ext")


#List all txt files
txtfile_list_WE_AG2_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)
```

```
#Create dataframe with all data
All_data_WE_AG2_Ext<- do.call("cbind", lapply(txtfile_list_WE_AG2_Ext, FUN = function(fi
les){

 read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG2_Ext)){
 All_data_WE_AG2_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG2_Ext <- gsub("_transf.txt", "", txtfile_list_WE_AG2_Ext)
colnames(All_data_WE_AG2_Ext) <- txtfile_list_WE_AG2_Ext


#Save data
write.table(All_data_WE_AG2_Ext, "~/Desktop/Mean Mapping/WE AG2 Ext/All_data_WE_A
G2_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per WE AG

```
## WE AGE GROUP 2

#Select individuals
WE_AG2_Ext <- All_data_WE_AG2_Ext


#Calculate mean

Mean_WE_AG2_Ext <- c()
for(i in 1:nrow(WE_AG2_Ext)){
 mean1 <- mean(as.numeric(WE_AG2_Ext[i,]), na.rm = T)
```

```
  Mean_WE_AG2_Ext <- c(Mean_WE_AG2_Ext, mean1)
}

WE_AG2_Ext <- cbind(WE_AG2_Ext, Mean_WE_AG2_Ext)


#Save
write.table(WE_AG2_Ext[,5], "~/Desktop/Mean Mapping/WE AG2 Ext/  Mean_WE_AG2_Ext.t
xt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/WE AG2 Int")


#List all txt files
txtfile_list_WE_AG2_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.n
ames = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_WE_AG2_Int<- do.call("cbind", lapply(txtfile_list_WE_AG2_Int, FUN = function(fil
es){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG2_Int)){
  All_data_WE_AG2_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG2_Int <- gsub("_transf.txt", "", txtfile_list_WE_AG2_Int)
colnames(All_data_WE_AG2_Int) <- txtfile_list_WE_AG2_Int


#Save data
```

```
write.table(All_data_WE_AG2_Int, "~/Desktop/Mean Mapping/WE AG2 Int/All_data_WE_AG
2_Int.txt", sep = "\t")
```

###2.2. Create mean maps per WE AG

```
## WE AGE GROUP 2

#Select individuals
WE_AG2_Int <- All_data_WE_AG2_Int


#Calculate mean

Mean_WE_AG2_Int <- c()
for(i in 1:nrow(WE_AG2_Int)){
  mean1 <- mean(as.numeric(WE_AG2_Int[i,]), na.rm = T)
  Mean_WE_AG2_Int <- c(Mean_WE_AG2_Int, mean1)
}

WE_AG2_Int <- cbind(WE_AG2_Int, Mean_WE_AG2_Int)


#Save
write.table(WE_AG2_Int[,5], "~/Desktop/Mean Mapping/WE AG2 Int/  Mean_WE_AG2_Int.txt
")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/WE AG3 Ext")


#List all txt files
txtfile_list_WE_AG3_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)
```

```
#Create dataframe with all data
All_data_WE_AG3_Ext<- do.call("cbind", lapply(txtfile_list_WE_AG3_Ext, FUN = function(fi
les){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG3_Ext)){
  All_data_WE_AG3_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG3_Ext <- gsub("_transf.txt", "", txtfile_list_WE_AG3_Ext)
colnames(All_data_WE_AG3_Ext) <- txtfile_list_WE_AG3_Ext


#Save data
write.table(All_data_WE_AG3_Ext, "~/Desktop/Mean Mapping/WE AG3 Ext/All_data_WE_A
G3_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per WE AG

```
## WE AGE GROUP 3

#Select individuals
WE_AG3_Ext <- All_data_WE_AG3_Ext


#Calculate mean

Mean_WE_AG3_Ext <- c()
```

```
for(i in 1:nrow(WE_AG3_Ext)){
  mean1 <- mean(as.numeric(WE_AG3_Ext[i,]), na.rm = T)
  Mean_WE_AG3_Ext <- c(Mean_WE_AG3_Ext, mean1)
}

WE_AG3_Ext <- cbind(WE_AG3_Ext, Mean_WE_AG3_Ext)


#Save
write.table(WE_AG3_Ext[,5], "~/Desktop/Mean Mapping/WE AG3 Ext/  Mean_WE_AG3_Ext.t
xt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/WE AG3 Int")


#List all txt files
txtfile_list_WE_AG3_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.n
ames = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_WE_AG3_Int<- do.call("cbind", lapply(txtfile_list_WE_AG3_Int, FUN = function(fil
es){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG3_Int)){
  All_data_WE_AG3_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG3_Int <- gsub("_transf.txt", "", txtfile_list_WE_AG3_Int)
colnames(All_data_WE_AG3_Int) <- txtfile_list_WE_AG3_Int
```

```
#Save data
write.table(All_data_WE_AG3_Int, "~/Desktop/Mean Mapping/WE AG3 Int/All_data_WE_AG
3_Int.txt", sep = "\t")
```

###2.2. Create mean maps per WE AG

WE AGE GROUP 3

```
#Select individuals
WE_AG3_Int <- All_data_WE_AG3_Int


#Calculate mean

Mean_WE_AG3_Int <- c()
for(i in 1:nrow(WE_AG3_Int)){
  mean1 <- mean(as.numeric(WE_AG3_Int[i,]), na.rm = T)
  Mean_WE_AG3_Int <- c(Mean_WE_AG3_Int, mean1)
}

WE_AG3_Int <- cbind(WE_AG3_Int, Mean_WE_AG3_Int)


#Save
write.table(WE_AG3_Int[,5], "~/Desktop/Mean Mapping/WE AG3 Int/  Mean_WE_AG3_Int.txt
")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/WE AG4 Ext")


#List all txt files
txtfile_list_WE_AG4_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
```

```
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_WE_AG4_Ext<- do.call("cbind", lapply(txtfile_list_WE_AG4_Ext, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG4_Ext)){
  All_data_WE_AG4_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG4_Ext <- gsub("_transf.txt", "", txtfile_list_WE_AG4_Ext)
colnames(All_data_WE_AG4_Ext) <- txtfile_list_WE_AG4_Ext


#Save data
write.table(All_data_WE_AG4_Ext, "~/Desktop/Mean Mapping/WE AG4 Ext/All_data_WE_AG4_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per WE AG

WE AGE GROUP 4

```
#Select individuals
WE_AG4_Ext <- All_data_WE_AG4_Ext


#Calculate mean
```

```
Mean_WE_AG4_Ext <- c()
for(i in 1:nrow(WE_AG4_Ext)){
  mean1 <- mean(as.numeric(WE_AG4_Ext[i,]), na.rm = T)
  Mean_WE_AG4_Ext <- c(Mean_WE_AG4_Ext, mean1)
}

WE_AG4_Ext <- cbind(WE_AG4_Ext, Mean_WE_AG4_Ext)


#Save
write.table(WE_AG4_Ext[,5], "~/Desktop/Mean Mapping/WE AG4 Ext/  Mean_WE_AG4_Ext.t
xt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/WE AG4 Int")


#List all txt files
txtfile_list_WE_AG4_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.n
ames = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_WE_AG4_Int<- do.call("cbind", lapply(txtfile_list_WE_AG4_Int, FUN = function(fil
es){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_WE_AG4_Int)){
  All_data_WE_AG4_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_WE_AG4_Int <- gsub("_transf.txt", "", txtfile_list_WE_AG4_Int)
```

```
colnames(All_data_WE_AG4_Int) <- txtfile_list_WE_AG4_Int


#Save data
write.table(All_data_WE_AG4_Int, "~/Desktop/Mean Mapping/WE AG4 Int/All_data_WE_AG
4_Int.txt", sep = "\t")
```

###2.2. Create mean maps per WE AG

WE AGE GROUP 4

```
#Select individuals
WE_AG4_Int <- All_data_WE_AG4_Int


#Calculate mean

Mean_WE_AG4_Int <- c()
for(i in 1:nrow(WE_AG4_Int)){
  mean1 <- mean(as.numeric(WE_AG4_Int[i,]), na.rm = T)
  Mean_WE_AG4_Int <- c(Mean_WE_AG4_Int, mean1)
}

WE_AG4_Int <- cbind(WE_AG4_Int, Mean_WE_AG4_Int)


#Save
write.table(WE_AG4_Int[,5], "~/Desktop/Mean Mapping/WE AG4 Int/  Mean_WE_AG4_Int.txt
")
```

cho = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
library(ggplot2)
library(scales)
library(readr)

##
## Attaching package: 'readr'

## The following object is masked from 'package:scales':
##
##     col_factor

library(RColorBrewer)
library(fields)

## Loading required package: spam

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':
##
##     viridis_pal

##
## Try help(fields) to get started.

library(raster)

## Loading required package: sp
```

```
library(colorRamps)
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4

library(proxy)

##
## Attaching package: 'proxy'

## The following object is masked from 'package:raster':
##
##     as.matrix

## The following object is masked from 'package:spam':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

library(fields)
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/South African AG1 Ext")


#List all txt files
txtfile_list_South_African_AG1_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FA
LSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no
.. = FALSE)


#Create dataframe with all data
All_data_South_African_AG1_Ext<- do.call("cbind", lapply(txtfile_list_South_African_AG1_E
```

```
xt, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_South_African_AG1_Ext)){
  All_data_South_African_AG1_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_South_African_AG1_Ext <- gsub("_transf.txt", "", txtfile_list_South_African_AG1_
Ext)
colnames(All_data_South_African_AG1_Ext) <- txtfile_list_South_African_AG1_Ext


#Save data
write.table(All_data_South_African_AG1_Ext, "~/Desktop/Mean Mapping/South African AG1
Ext/All_data_South_African_AG1_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per South African AG

South African AGE GROUP 1

```
#Select individuals
South_African_AG1_Ext <- All_data_South_African_AG1_Ext


#Calculate mean

Mean_South_African_AG1_Ext <- c()
for(i in 1:nrow(South_African_AG1_Ext)){
  mean1 <- mean(as.numeric(South_African_AG1_Ext[i,]), na.rm = T)
  Mean_South_African_AG1_Ext <- c(Mean_South_African_AG1_Ext, mean1)
```

```
}

South_African_AG1_Ext <- cbind(South_African_AG1_Ext, Mean_South_African_AG1_Ext)


#Save
write.table(South_African_AG1_Ext[,5], "~/Desktop/Mean Mapping/South African AG1 Ext/Mean_South_African_AG1_Ext.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/South African AG1 Int")


#List all txt files
txtfile_list_South_African_AG1_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_South_African_AG1_Int<- do.call("cbind", lapply(txtfile_list_South_African_AG1_Int, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_South_African_AG1_Int)){
  All_data_South_African_AG1_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_South_African_AG1_Int <- gsub("_transf.txt", "", txtfile_list_South_African_AG1_Int)
colnames(All_data_South_African_AG1_Int) <- txtfile_list_South_African_AG1_Int


#Save data
```

```
write.table(All_data_South_African_AG1_Int, "~/Desktop/Mean Mapping/South African AG1 I
nt/All_data_South_African_AG1_Int.txt", sep = "\t")
```

###2.2. Create mean maps per South African AG

South African AGE GROUP 1

```
#Select individuals
South_African_AG1_Int <- All_data_South_African_AG1_Int


#Calculate mean

Mean_South_African_AG1_Int <- c()
for(i in 1:nrow(South_African_AG1_Int)){
  mean1 <- mean(as.numeric(South_African_AG1_Int[i,]), na.rm = T)
  Mean_South_African_AG1_Int <- c(Mean_South_African_AG1_Int, mean1)
}

South_African_AG1_Int <- cbind(South_African_AG1_Int, Mean_South_African_AG1_Int)


#Save
write.table(South_African_AG1_Int[,5], "~/Desktop/Mean Mapping/South African AG1 Int/  Me
an_South_African_AG1_Int.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/South African AG2 Ext")


#List all txt files
txtfile_list_South_African_AG2_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FA
LSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no
.. = FALSE)
```

```
#Create dataframe with all data
All_data_South_African_AG2_Ext<- do.call("cbind", lapply(txtfile_list_South_African_AG2_E
xt, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_South_African_AG2_Ext)){
  All_data_South_African_AG2_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_South_African_AG2_Ext <- gsub("_transf.txt", "", txtfile_list_South_African_AG2_
Ext)
colnames(All_data_South_African_AG2_Ext) <- txtfile_list_South_African_AG2_Ext


#Save data
write.table(All_data_South_African_AG2_Ext, "~/Desktop/Mean Mapping/South African AG2
Ext/All_data_South_African_AG2_Ext.txt", sep = "\t")
```

2.2. Create mean maps per South African AG

South African AGE GROUP 2

```
#Select individuals
South_African_AG2_Ext <- All_data_South_African_AG2_Ext


#Calculate mean
```

```
Mean_South_African_AG2_Ext <- c()
for(i in 1:nrow(South_African_AG2_Ext)){
  mean1 <- mean(as.numeric(South_African_AG2_Ext[i,]), na.rm = T)
  Mean_South_African_AG2_Ext <- c(Mean_South_African_AG2_Ext, mean1)
}

South_African_AG2_Ext <- cbind(South_African_AG2_Ext, Mean_South_African_AG2_Ext)


#Save
write.table(South_African_AG2_Ext[,5], "~/Desktop/Mean Mapping/South African AG2 Ext/
Mean_South_African_AG2_Ext.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/South African AG2 Int")


#List all txt files
txtfile_list_South_African_AG2_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FA
LSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no
.. = FALSE)


#Create dataframe with all data
All_data_South_African_AG2_Int<- do.call("cbind", lapply(txtfile_list_South_African_AG2_Int
, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_South_African_AG2_Int)){
  All_data_South_African_AG2_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_South_African_AG2_Int <- gsub("_transf.txt", "", txtfile_list_South_African_AG2_I
nt)
```

```
colnames(All_data_South_African_AG2_Int) <- txtfile_list_South_African_AG2_Int


#Save data
write.table(All_data_South_African_AG2_Int, "~/Desktop/Mean Mapping/South African AG2 I
nt/All_data_South_African_AG2_Int.txt", sep = "\t")
```

###2.2. Create mean maps per South African AG

South African AGE GROUP 2

```
#Select individuals
South_African_AG2_Int <- All_data_South_African_AG2_Int


#Calculate mean

Mean_South_African_AG2_Int <- c()
for(i in 1:nrow(South_African_AG2_Int)){
  mean1 <- mean(as.numeric(South_African_AG2_Int[i,]), na.rm = T)
  Mean_South_African_AG2_Int <- c(Mean_South_African_AG2_Int, mean1)
}

South_African_AG2_Int <- cbind(South_African_AG2_Int, Mean_South_African_AG2_Int)


#Save
write.table(South_African_AG2_Int[,5], "~/Desktop/Mean Mapping/South African AG2 Int/  Me
an_South_African_AG2_Int.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/South African AG3 Ext")
```

```
#List all txt files
txtfile_list_South_African_AG3_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FA
LSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no
.. = FALSE)


#Create dataframe with all data
All_data_South_African_AG3_Ext<- do.call("cbind", lapply(txtfile_list_South_African_AG3_E
xt, FUN = function(files){

 read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_South_African_AG3_Ext)){
  All_data_South_African_AG3_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_South_African_AG3_Ext <- gsub("_transf.txt", "", txtfile_list_South_African_AG3_
Ext)
colnames(All_data_South_African_AG3_Ext) <- txtfile_list_South_African_AG3_Ext


#Save data
write.table(All_data_South_African_AG3_Ext, "~/Desktop/Mean Mapping/South African AG3
Ext/All_data_South_African_AG3_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per South African AG

South African AGE GROUP 3

```
#Select individuals
South_African_AG3_Ext <- All_data_South_African_AG3_Ext
```

```
#Calculate mean

Mean_South_African_AG3_Ext <- c()
for(i in 1:nrow(South_African_AG3_Ext)){
  mean1 <- mean(as.numeric(South_African_AG3_Ext[i,]), na.rm = T)
  Mean_South_African_AG3_Ext <- c(Mean_South_African_AG3_Ext, mean1)
}

South_African_AG3_Ext <- cbind(South_African_AG3_Ext, Mean_South_African_AG3_Ext)


#Save
write.table(South_African_AG3_Ext[,5], "~/Desktop/Mean Mapping/South African AG3 Ext/
Mean_South_African_AG3_Ext.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/South African AG3 Int")


#List all txt files
txtfile_list_South_African_AG3_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FA
LSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no
.. = FALSE)


#Create dataframe with all data
All_data_South_African_AG3_Int<- do.call("cbind", lapply(txtfile_list_South_African_AG3_Int
, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_South_African_AG3_Int)){
  All_data_South_African_AG3_Int[,c("x", "y")] <- NULL
}
```

```
#Name
txtfile_list_South_African_AG3_Int <- gsub("_transf.txt", "", txtfile_list_South_African_AG3_I
nt)
colnames(All_data_South_African_AG3_Int) <- txtfile_list_South_African_AG3_Int


#Save data
write.table(All_data_South_African_AG3_Int, "~/Desktop/Mean Mapping/South African AG3 I
nt/All_data_South_African_AG3_Int.txt", sep = "\t")
```

###2.2. Create mean maps per South African AG

South African AGE GROUP 3

```
#Select individuals
South_African_AG3_Int <- All_data_South_African_AG3_Int


#Calculate mean

Mean_South_African_AG3_Int <- c()
for(i in 1:nrow(South_African_AG3_Int)){
  mean1 <- mean(as.numeric(South_African_AG3_Int[i,]), na.rm = T)
  Mean_South_African_AG3_Int <- c(Mean_South_African_AG3_Int, mean1)
}

South_African_AG3_Int <- cbind(South_African_AG3_Int, Mean_South_African_AG3_Int)


#Save
write.table(South_African_AG3_Int[,5], "~/Desktop/Mean Mapping/South African AG3 Int/  Me
an_South_African_AG3_Int.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/South African AG4 Ext")


#List all txt files
txtfile_list_South_African_AG4_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FA
LSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no
.. = FALSE)


#Create dataframe with all data
All_data_South_African_AG4_Ext<- do.call("cbind", lapply(txtfile_list_South_African_AG4_E
xt, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_South_African_AG4_Ext)){
  All_data_South_African_AG4_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_South_African_AG4_Ext <- gsub("_transf.txt", "", txtfile_list_South_African_AG4_
Ext)
colnames(All_data_South_African_AG4_Ext) <- txtfile_list_South_African_AG4_Ext


#Save data
write.table(All_data_South_African_AG4_Ext, "~/Desktop/Mean Mapping/South African AG4
Ext/All_data_South_African_AG4_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per South African AG

```
## South African AGE GROUP 4

#Select individuals
South_African_AG4_Ext <- All_data_South_African_AG4_Ext


#Calculate mean

Mean_South_African_AG4_Ext <- c()
for(i in 1:nrow(South_African_AG4_Ext)){
  mean1 <- mean(as.numeric(South_African_AG4_Ext[i,]), na.rm = T)
  Mean_South_African_AG4_Ext <- c(Mean_South_African_AG4_Ext, mean1)
}

South_African_AG4_Ext <- cbind(South_African_AG4_Ext, Mean_South_African_AG4_Ext)


#Save
write.table(South_African_AG4_Ext[,5], "~/Desktop/Mean Mapping/South African AG4 Ext/
Mean_South_African_AG4_Ext.txt")

###2.1. Create dataframe with all data




#Set wd
setwd("~/Desktop/Mean Mapping/South African AG4 Int")


#List all txt files
txtfile_list_South_African_AG4_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FA
LSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no
.. = FALSE)


#Create dataframe with all data
All_data_South_African_AG4_Int<- do.call("cbind", lapply(txtfile_list_South_African_AG4_Int
, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns
```

```
for(i in 1:length(All_data_South_African_AG4_Int)){
  All_data_South_African_AG4_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_South_African_AG4_Int <- gsub("_transf.txt", "", txtfile_list_South_African_AG4_I
nt)
colnames(All_data_South_African_AG4_Int) <- txtfile_list_South_African_AG4_Int


#Save data
write.table(All_data_South_African_AG4_Int, "~/Desktop/Mean Mapping/South African AG4 I
nt/All_data_South_African_AG4_Int.txt", sep = "\t")
```

###2.2. Create mean maps per South African AG

South African AGE GROUP 4

```
#Select individuals
South_African_AG4_Int <- All_data_South_African_AG4_Int


#Calculate mean

Mean_South_African_AG4_Int <- c()
for(i in 1:nrow(South_African_AG4_Int)){
  mean1 <- mean(as.numeric(South_African_AG4_Int[i,]), na.rm = T)
  Mean_South_African_AG4_Int <- c(Mean_South_African_AG4_Int, mean1)
}

South_African_AG4_Int <- cbind(South_African_AG4_Int, Mean_South_African_AG4_Int)


#Save
write.table(South_African_AG4_Int[,5], "~/Desktop/Mean Mapping/South African AG4 Int/  Me
an_South_African_AG4_Int.txt")
```

Inuit Mean Maps
2023-02-28

title: “Inuit Mean Maps” output:word_document date: “2023-02-28”

```
library(ggplot2)
library(scales)
library(readr)

##
## Attaching package: 'readr'

## The following object is masked from 'package:scales':
##
##     col_factor

library(RColorBrewer)
library(fields)

## Loading required package: spam

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':
##
##     viridis_pal

##
## Try help(fields) to get started.

library(raster)

## Loading required package: sp
```

```
library(colorRamps)
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4

library(proxy)

##
## Attaching package: 'proxy'

## The following object is masked from 'package:raster':
##
##     as.matrix

## The following object is masked from 'package:spam':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

library(fields)
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG1 Ext")


#List all txt files
txtfile_list_Inuit_AG1_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_Inuit_AG1_Ext<- do.call("cbind", lapply(txtfile_list_Inuit_AG1_Ext, FUN = function(
```

```
files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG1_Ext)){
  All_data_Inuit_AG1_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG1_Ext <- gsub("_transf.txt", "", txtfile_list_Inuit_AG1_Ext)
colnames(All_data_Inuit_AG1_Ext) <- txtfile_list_Inuit_AG1_Ext


#Save data
write.table(All_data_Inuit_AG1_Ext, "~/Desktop/Mean Mapping/Inuit AG1 Ext/All_data_Inuit_
AG1_Ext.txt", sep = "\t")
```

2.2. Create mean maps per Inuit AG

Inuit AGE GROUP 1

```
#Select individuals
Inuit_AG1_Ext <- All_data_Inuit_AG1_Ext


#Calculate mean

Mean_Inuit_AG1_Ext <- c()
for(i in 1:nrow(Inuit_AG1_Ext)){
  mean1 <- mean(as.numeric(Inuit_AG1_Ext[i,]), na.rm = T)
  Mean_Inuit_AG1_Ext <- c(Mean_Inuit_AG1_Ext, mean1)
}
```

```
Inuit_AG1_Ext <- cbind(Inuit_AG1_Ext, Mean_Inuit_AG1_Ext)


#Save
write.table(Inuit_AG1_Ext[,5], "~/Desktop/Mean Mapping/Inuit AG1 Ext/Mean_Inuit_AG1_Ext
.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG1 Int")


#List all txt files
txtfile_list_Inuit_AG1_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_Inuit_AG1_Int<- do.call("cbind", lapply(txtfile_list_Inuit_AG1_Int, FUN = function(f
iles){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG1_Int)){
  All_data_Inuit_AG1_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG1_Int <- gsub("_transf.txt", "", txtfile_list_Inuit_AG1_Int)
colnames(All_data_Inuit_AG1_Int) <- txtfile_list_Inuit_AG1_Int


#Save data
write.table(All_data_Inuit_AG1_Int, "~/Desktop/Mean Mapping/Inuit AG1 Int/All_data_Inuit_A
G1_Int.txt", sep = "\t")
```

```
###2.2. Create mean maps per Inuit AG



## Inuit AGE GROUP 1

#Select individuals
Inuit_AG1_Int <- All_data_Inuit_AG1_Int


#Calculate mean

Mean_Inuit_AG1_Int <- c()
for(i in 1:nrow(Inuit_AG1_Int)){
  mean1 <- mean(as.numeric(Inuit_AG1_Int[i,]), na.rm = T)
  Mean_Inuit_AG1_Int <- c(Mean_Inuit_AG1_Int, mean1)
}

Inuit_AG1_Int <- cbind(Inuit_AG1_Int, Mean_Inuit_AG1_Int)


#Save
write.table(Inuit_AG1_Int[,5], "~/Desktop/Mean Mapping/Inuit AG1 Int/  Mean_Inuit_AG1_Int.
txt")
```

```
###2.1. Create dataframe with all data



#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG2 Ext")


#List all txt files
txtfile_list_Inuit_AG2_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)
```

```
#Create dataframe with all data
All_data_Inuit_AG2_Ext<- do.call("cbind", lapply(txtfile_list_Inuit_AG2_Ext, FUN = function(
files){

 read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG2_Ext)){
 All_data_Inuit_AG2_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG2_Ext <- gsub("_transf.txt", "", txtfile_list_Inuit_AG2_Ext)
colnames(All_data_Inuit_AG2_Ext) <- txtfile_list_Inuit_AG2_Ext


#Save data
write.table(All_data_Inuit_AG2_Ext, "~/Desktop/Mean Mapping/Inuit AG2 Ext/All_data_Inuit_
AG2_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per Inuit AG

```
## Inuit AGE GROUP 2

#Select individuals
Inuit_AG2_Ext <- All_data_Inuit_AG2_Ext


#Calculate mean

Mean_Inuit_AG2_Ext <- c()
for(i in 1:nrow(Inuit_AG2_Ext)){
 mean1 <- mean(as.numeric(Inuit_AG2_Ext[i,]), na.rm = T)
```

```
  Mean_Inuit_AG2_Ext <- c(Mean_Inuit_AG2_Ext, mean1)
}

Inuit_AG2_Ext <- cbind(Inuit_AG2_Ext, Mean_Inuit_AG2_Ext)


#Save
write.table(Inuit_AG2_Ext[,5], "~/Desktop/Mean Mapping/Inuit AG2 Ext/  Mean_Inuit_AG2_E
xt.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG2 Int")


#List all txt files
txtfile_list_Inuit_AG2_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_Inuit_AG2_Int<- do.call("cbind", lapply(txtfile_list_Inuit_AG2_Int, FUN = function(f
iles){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG2_Int)){
  All_data_Inuit_AG2_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG2_Int <- gsub("_transf.txt", "", txtfile_list_Inuit_AG2_Int)
colnames(All_data_Inuit_AG2_Int) <- txtfile_list_Inuit_AG2_Int


#Save data
```

```
write.table(All_data_Inuit_AG2_Int, "~/Desktop/Mean Mapping/Inuit AG2 Int/All_data_Inuit_AG2_Int.txt", sep = "\t")
```

###2.2. Create mean maps per Inuit AG

Inuit AGE GROUP 2

```
#Select individuals
Inuit_AG2_Int <- All_data_Inuit_AG2_Int


#Calculate mean

Mean_Inuit_AG2_Int <- c()
for(i in 1:nrow(Inuit_AG2_Int)){
  mean1 <- mean(as.numeric(Inuit_AG2_Int[i,]), na.rm = T)
  Mean_Inuit_AG2_Int <- c(Mean_Inuit_AG2_Int, mean1)
}

Inuit_AG2_Int <- cbind(Inuit_AG2_Int, Mean_Inuit_AG2_Int)


#Save
write.table(Inuit_AG2_Int[,5], "~/Desktop/Mean Mapping/Inuit AG2 Int/  Mean_Inuit_AG2_Int.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG3 Ext")


#List all txt files
txtfile_list_Inuit_AG3_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)
```

```
#Create dataframe with all data
All_data_Inuit_AG3_Ext<- do.call("cbind", lapply(txtfile_list_Inuit_AG3_Ext, FUN = function(
files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG3_Ext)){
  All_data_Inuit_AG3_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG3_Ext <- gsub("_transf.txt", "", txtfile_list_Inuit_AG3_Ext)
colnames(All_data_Inuit_AG3_Ext) <- txtfile_list_Inuit_AG3_Ext


#Save data
write.table(All_data_Inuit_AG3_Ext, "~/Desktop/Mean Mapping/Inuit AG3 Ext/All_data_Inuit_
AG3_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per Inuit AG

```
## Inuit AGE GROUP 3

#Select individuals
Inuit_AG3_Ext <- All_data_Inuit_AG3_Ext


#Calculate mean

Mean_Inuit_AG3_Ext <- c()
```

```
for(i in 1:nrow(Inuit_AG3_Ext)){
  mean1 <- mean(as.numeric(Inuit_AG3_Ext[i,]), na.rm = T)
  Mean_Inuit_AG3_Ext <- c(Mean_Inuit_AG3_Ext, mean1)
}

Inuit_AG3_Ext <- cbind(Inuit_AG3_Ext, Mean_Inuit_AG3_Ext)


#Save
write.table(Inuit_AG3_Ext[,5], "~/Desktop/Mean Mapping/Inuit AG3 Ext/  Mean_Inuit_AG3_Ext.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG3 Int")


#List all txt files
txtfile_list_Inuit_AG3_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_Inuit_AG3_Int<- do.call("cbind", lapply(txtfile_list_Inuit_AG3_Int, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG3_Int)){
  All_data_Inuit_AG3_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG3_Int <- gsub("_transf.txt", "", txtfile_list_Inuit_AG3_Int)
colnames(All_data_Inuit_AG3_Int) <- txtfile_list_Inuit_AG3_Int
```

```
#Save data
write.table(All_data_Inuit_AG3_Int, "~/Desktop/Mean Mapping/Inuit AG3 Int/All_data_Inuit_AG3_Int.txt", sep = "\t")
```

###2.2. Create mean maps per Inuit AG

Inuit AGE GROUP 3

```
#Select individuals
Inuit_AG3_Int <- All_data_Inuit_AG3_Int

#Calculate mean

Mean_Inuit_AG3_Int <- c()
for(i in 1:nrow(Inuit_AG3_Int)){
  mean1 <- mean(as.numeric(Inuit_AG3_Int[i,]), na.rm = T)
  Mean_Inuit_AG3_Int <- c(Mean_Inuit_AG3_Int, mean1)
}

Inuit_AG3_Int <- cbind(Inuit_AG3_Int, Mean_Inuit_AG3_Int)

#Save
write.table(Inuit_AG3_Int[,5], "~/Desktop/Mean Mapping/Inuit AG3 Int/  Mean_Inuit_AG3_Int.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG4 Ext")

#List all txt files
txtfile_list_Inuit_AG4_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
```

```
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_Inuit_AG4_Ext<- do.call("cbind", lapply(txtfile_list_Inuit_AG4_Ext, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG4_Ext)){
  All_data_Inuit_AG4_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG4_Ext <- gsub("_transf.txt", "", txtfile_list_Inuit_AG4_Ext)
colnames(All_data_Inuit_AG4_Ext) <- txtfile_list_Inuit_AG4_Ext


#Save data
write.table(All_data_Inuit_AG4_Ext, "~/Desktop/Mean Mapping/Inuit AG4 Ext/All_data_Inuit_AG4_Ext.txt", sep = "\t")
```

###2.2. Create mean maps per Inuit AG

Inuit AGE GROUP 4

```
#Select individuals
Inuit_AG4_Ext <- All_data_Inuit_AG4_Ext


#Calculate mean
```

```
Mean_Inuit_AG4_Ext <- c()
for(i in 1:nrow(Inuit_AG4_Ext)){
  mean1 <- mean(as.numeric(Inuit_AG4_Ext[i,]), na.rm = T)
  Mean_Inuit_AG4_Ext <- c(Mean_Inuit_AG4_Ext, mean1)
}

Inuit_AG4_Ext <- cbind(Inuit_AG4_Ext, Mean_Inuit_AG4_Ext)


#Save
write.table(Inuit_AG4_Ext[,5], "~/Desktop/Mean Mapping/Inuit AG4 Ext/  Mean_Inuit_AG4_E
xt.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Inuit AG4 Int")


#List all txt files
txtfile_list_Inuit_AG4_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.
names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALS
E)


#Create dataframe with all data
All_data_Inuit_AG4_Int<- do.call("cbind", lapply(txtfile_list_Inuit_AG4_Int, FUN = function(f
iles){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Inuit_AG4_Int)){
  All_data_Inuit_AG4_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Inuit_AG4_Int <- gsub("_transf.txt", "", txtfile_list_Inuit_AG4_Int)
```

```
colnames(All_data_Inuit_AG4_Int) <- txtfile_list_Inuit_AG4_Int


#Save data
write.table(All_data_Inuit_AG4_Int, "~/Desktop/Mean Mapping/Inuit AG4 Int/All_data_Inuit_AG4_Int.txt", sep = "\t")
```

###2.2. Create mean maps per Inuit AG

Inuit AGE GROUP 4

```
#Select individuals
Inuit_AG4_Int <- All_data_Inuit_AG4_Int


#Calculate mean

Mean_Inuit_AG4_Int <- c()
for(i in 1:nrow(Inuit_AG4_Int)){
  mean1 <- mean(as.numeric(Inuit_AG4_Int[i,]), na.rm = T)
  Mean_Inuit_AG4_Int <- c(Mean_Inuit_AG4_Int, mean1)
}

Inuit_AG4_Int <- cbind(Inuit_AG4_Int, Mean_Inuit_AG4_Int)


#Save
write.table(Inuit_AG4_Int[,5], "~/Desktop/Mean Mapping/Inuit AG4 Int/Mean_Inuit_AG4_Int.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Female Ext")


#List all txt files
txtfile_list_Female_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.na
mes = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_Female_Ext<- do.call("cbind", lapply(txtfile_list_Female_Ext, FUN = function(files){

 read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Female_Ext)){
  All_data_Female_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Female_Ext <- gsub("_transf.txt", "", txtfile_list_Female_Ext)
colnames(All_data_Female_Ext) <- txtfile_list_Female_Ext


#Save data
write.table(All_data_Female_Ext, "~/Desktop/Mean Mapping/Female Ext/All_data_Female_Ext.
txt", sep = "\t")
```

###2.2. Create mean maps

```
##

#Select individuals
Female_Ext <- All_data_Female_Ext


#Calculate mean

Mean_Female_Ext <- c()
for(i in 1:nrow(Female_Ext)){
  mean1 <- mean(as.numeric(Female_Ext[i,]), na.rm = T)
  Mean_Female_Ext <- c(Mean_Female_Ext, mean1)
}

Female_Ext <- cbind(Female_Ext, Mean_Female_Ext)


#Save
write.table(Female_Ext[,6], "~/Desktop/Mean Mapping/Female Ext/  Mean_Female_Ext.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
setwd("~/Desktop/Mean Mapping/Female Int")


#List all txt files
txtfile_list_Female_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.nam
es = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_Female_Int<- do.call("cbind", lapply(txtfile_list_Female_Int, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns
```

```
for(i in 1:length(All_data_Female_Int)){
  All_data_Female_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Female_Int <- gsub("_transf.txt", "", txtfile_list_Female_Int)
colnames(All_data_Female_Int) <- txtfile_list_Female_Int


#Save data
write.table(All_data_Female_Int, "~/Desktop/Mean Mapping/Female Int/All_data_Female_Int.tx
t", sep = "\t")









#Select individuals
Female_Int <- All_data_Female_Int


#Calculate mean

Mean_Female_Int <- c()
for(i in 1:nrow(Female_Int)){
  mean1 <- mean(as.numeric(Female_Int[i,]), na.rm = T)
  Mean_Female_Int <- c(Mean_Female_Int, mean1)
}

Female_Int <- cbind(Female_Int, Mean_Female_Int)


#Save
write.table(Female_Int[,6], "~/Desktop/Mean Mapping/Female Int/  Mean_Female_Int.txt")
```

###2.1. Create dataframe with all data

```
#Set wd
```

```
setwd("~/Desktop/Mean Mapping/Male Int")


#List all txt files
txtfile_list_Male_Int <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.names
= FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_Male_Int<- do.call("cbind", lapply(txtfile_list_Male_Int, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Male_Int)){
  All_data_Male_Int[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Male_Int <- gsub("_transf.txt", "", txtfile_list_Male_Int)
colnames(All_data_Male_Int) <- txtfile_list_Male_Int


#Save data
write.table(All_data_Male_Int, "~/Desktop/Mean Mapping/Male Int/All_data_Male_Int.txt", sep
= "\t")









#Select individuals
Male_Int <- All_data_Male_Int


#Calculate mean

Mean_Male_Int <- c()
```

```
for(i in 1:nrow(Male_Int)){
  mean1 <- mean(as.numeric(Male_Int[i,]), na.rm = T)
  Mean_Male_Int <- c(Mean_Male_Int, mean1)
}

Male_Int <- cbind(Male_Int, Mean_Male_Int)


#Save
write.table(Male_Int[,8], "~/Desktop/Mean Mapping/Male Int/  Mean_Male_Int.txt")
```

###2.1. Create dataframe with all data

```


#Set wd
setwd("~/Desktop/Mean Mapping/Male Ext")


#List all txt files
txtfile_list_Male_Ext <- list.files(path = "./", pattern = "_transf.txt", all.files = FALSE, full.name
s = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)


#Create dataframe with all data
All_data_Male_Ext<- do.call("cbind", lapply(txtfile_list_Male_Ext, FUN = function(files){

  read.table(files, header = TRUE, sep = "\t", na = "NA")

}))


#Get rid of x and y columns

for(i in 1:length(All_data_Male_Ext)){
  All_data_Male_Ext[,c("x", "y")] <- NULL
}


#Name
txtfile_list_Male_Ext <- gsub("_transf.txt", "", txtfile_list_Male_Ext)
colnames(All_data_Male_Ext) <- txtfile_list_Male_Ext


#Save data
write.table(All_data_Male_Ext, "~/Desktop/Mean Mapping/Male Ext/All_data_Male_Ext.txt", s
```

```
ep = "\t")




#Select individuals
Male_Ext <- All_data_Male_Ext


#Calculate mean

Mean_Male_Ext <- c()
for(i in 1:nrow(Male_Ext)){
  mean1 <- mean(as.numeric(Male_Ext[i,]), na.rm = T)
  Mean_Male_Ext <- c(Mean_Male_Ext, mean1)
}

Male_Ext <- cbind(Male_Ext, Mean_Male_Ext)


#Save
write.table(Male_Ext[,8], "~/Desktop/Mean Mapping/Male Ext/  Mean_Male_Ext.txt")
```

Means and Standard Deviation

```
setwd("~/Desktop")
library(readxl)
```

```
Exterior_Total_BR <- read_excel("Exterior Total BR.xlsx")
View(Exterior_Total_BR)

Exterior_Total_BR1<-data.frame(t(Exterior_Total_BR[-1]))
colnames(Exterior_Total_BR1)<-Exterior_Total_BR[,1]

European_Ext_AG1_mean<-mean(Exterior_Total_BR1[1:4,])
print(European_Ext_AG1_mean)

## [1] 10.41356

European_Ext_AG1_sd<-sd(Exterior_Total_BR1[1:4,])
print(European_Ext_AG1_sd)

## [1] 8.725082

European_Ext_AG2_mean<-mean(Exterior_Total_BR1[5:8,])
print(European_Ext_AG2_mean)

## [1] 4.687986

European_Ext_AG2_sd<-sd(Exterior_Total_BR1[5:8,])
print(European_Ext_AG2_sd)

## [1] 1.768021

European_Ext_AG3_mean<-mean(Exterior_Total_BR1[9:12,])
print(European_Ext_AG3_mean)

## [1] 4.599532

European_Ext_AG3_sd<-sd(Exterior_Total_BR1[9:12,])
print(European_Ext_AG3_sd)

## [1] 1.76247

European_Ext_AG4_mean<-mean(Exterior_Total_BR1[13:16,])
print(European_Ext_AG4_mean)

## [1] 1.696388

European_Ext_AG4_sd<-sd(Exterior_Total_BR1[13:16,])
print(European_Ext_AG4_sd)

## [1] 2.643581

Inuit_Ext_AG1_mean<-mean(Exterior_Total_BR1[17:20,])
print(Inuit_Ext_AG1_mean)

## [1] 6.880468
```

```
Inuit_Ext_AG1_sd<-sd(Exterior_Total_BR1[17:20,])
print(Inuit_Ext_AG1_sd)
```

```
## [1] 5.530399
```

```
Inuit_Ext_AG2_mean<-mean(Exterior_Total_BR1[21:24,])
print(Inuit_Ext_AG2_mean)
```

```
## [1] 7.06303
```

```
Inuit_Ext_AG2_sd<-sd(Exterior_Total_BR1[21:24,])
print(Inuit_Ext_AG2_sd)
```

```
## [1] 9.466398
```

```
Inuit_Ext_AG3_mean<-mean(Exterior_Total_BR1[25:28,])
print(Inuit_Ext_AG3_mean)
```

```
## [1] 1.985705
```

```
Inuit_Ext_AG3_sd<-sd(Exterior_Total_BR1[25:28,])
print(Inuit_Ext_AG3_sd)
```

```
## [1] 1.601037
```

```
Inuit_Ext_AG4_mean<-mean(Exterior_Total_BR1[29:32,])
print(Inuit_Ext_AG4_mean)
```

```
## [1] 1.078883
```

```
Inuit_Ext_AG4_sd<-sd(Exterior_Total_BR1[29:32,])
print(Inuit_Ext_AG4_sd)
```

```
## [1] 0.7269953
```

```
SouthAfrican_Ext_AG1_mean<-mean(Exterior_Total_BR1[33:36,])
print(SouthAfrican_Ext_AG1_mean)
```

```
## [1] 4.456158
```

```
SouthAfrican_Ext_AG1_sd<-sd(Exterior_Total_BR1[33:36,])
print(SouthAfrican_Ext_AG1_sd)
```

```
## [1] 4.665758
```

```
SouthAfrican_Ext_AG2_mean<-mean(Exterior_Total_BR1[37:40,])
print(SouthAfrican_Ext_AG2_mean)
```

```
## [1] 6.138426
```

```
SouthAfrican_Ext_AG2_sd<-sd(Exterior_Total_BR1[37:40,])
print(SouthAfrican_Ext_AG2_sd)
```

```
## [1] 5.662077

SouthAfrican_Ext_AG3_mean<-mean(Exterior_Total_BR1[41:44,])
print(SouthAfrican_Ext_AG3_mean)

## [1] 3.748117

SouthAfrican_Ext_AG3_sd<-sd(Exterior_Total_BR1[41:44,])
print(SouthAfrican_Ext_AG3_sd)

## [1] 4.782113

SouthAfrican_Ext_AG4_mean<-mean(Exterior_Total_BR1[45:48,])
print(SouthAfrican_Ext_AG4_mean)

## [1] 3.532443

SouthAfrican_Ext_AG4_sd<-sd(Exterior_Total_BR1[45:48,])
print(SouthAfrican_Ext_AG4_sd)

## [1] 3.065049

setwd("~/Desktop")
library(readxl)
Interior_Total_BR <- read_excel("Interior Total BR.xlsx")
View(Interior_Total_BR)
Interior_Total_BR1<-data.frame(t(Interior_Total_BR[-1]))
colnames(Interior_Total_BR1)<-Interior_Total_BR[,1]

European_Int_AG1_mean<-mean(Interior_Total_BR1[1:4,])
print(European_Int_AG1_mean)

## [1] 6.553798

European_Int_AG1_sd<-sd(Interior_Total_BR1[1:4,])
print(European_Int_AG1_sd)

## [1] 7.134685

European_Int_AG2_mean<-mean(Interior_Total_BR1[5:8,])
print(European_Int_AG2_mean)

## [1] 4.362783

European_Int_AG2_sd<-sd(Interior_Total_BR1[5:8,])
print(European_Int_AG2_sd)

## [1] 2.042617

European_Int_AG3_mean<-mean(Interior_Total_BR1[9:12,])
print(European_Int_AG3_mean)
```

```
## [1] 7.033437

European_Int_AG3_sd<-sd(Interior_Total_BR1[9:12,])
print(European_Int_AG3_sd)

## [1] 4.650554

European_Int_AG4_mean<-mean(Interior_Total_BR1[13:16,])
print(European_Int_AG4_mean)

## [1] 2.023294

European_Int_AG4_sd<-sd(Interior_Total_BR1[13:16,])
print(European_Int_AG4_sd)

## [1] 1.887167

Inuit_Int_AG1_mean<-mean(Interior_Total_BR1[17:20,])
print(Inuit_Int_AG1_mean)

## [1] 2.749402

Inuit_Int_AG1_sd<-sd(Interior_Total_BR1[17:20,])
print(Inuit_Int_AG1_sd)

## [1] 1.907089

Inuit_Int_AG2_mean<-mean(Interior_Total_BR1[21:24,])
print(Inuit_Int_AG2_mean)

## [1] 7.334208

Inuit_Int_AG2_sd<-sd(Interior_Total_BR1[21:24,])
print(Inuit_Int_AG2_sd)

## [1] 6.712832

Inuit_Int_AG3_mean<-mean(Interior_Total_BR1[25:28,])
print(Inuit_Int_AG3_mean)

## [1] 2.700047

Inuit_Int_AG3_sd<-sd(Interior_Total_BR1[25:28,])
print(Inuit_Int_AG3_sd)

## [1] 3.043529

Inuit_Int_AG4_mean<-mean(Interior_Total_BR1[29:32,])
print(Inuit_Int_AG4_mean)

## [1] 2.180603
```

```
Inuit_Int_AG4_sd<-sd(Interior_Total_BR1[29:32,])
print(Inuit_Int_AG4_sd)
```

```
## [1] 1.864205
```

```
SouthAfrican_Int_AG1_mean<-mean(Interior_Total_BR1[33:36,])
print(SouthAfrican_Int_AG1_mean)
```

```
## [1] 2.51206
```

```
SouthAfrican_Int_AG1_sd<-sd(Interior_Total_BR1[33:36,])
print(SouthAfrican_Int_AG1_sd)
```

```
## [1] 2.886581
```

```
SouthAfrican_Int_AG2_mean<-mean(Interior_Total_BR1[37:40,])
print(SouthAfrican_Int_AG2_mean)
```

```
## [1] 4.779655
```

```
SouthAfrican_Int_AG2_sd<-sd(Interior_Total_BR1[37:40,])
print(SouthAfrican_Int_AG2_sd)
```

```
## [1] 3.744011
```

```
SouthAfrican_Int_AG3_mean<-mean(Interior_Total_BR1[41:44,])
print(SouthAfrican_Int_AG3_mean)
```

```
## [1] 4.241579
```

```
SouthAfrican_Int_AG3_sd<-sd(Interior_Total_BR1[41:44,])
print(SouthAfrican_Int_AG3_sd)
```

```
## [1] 5.169931
```

```
SouthAfrican_Int_AG4_mean<-mean(Interior_Total_BR1[45:48,])
print(SouthAfrican_Int_AG4_mean)
```

```
## [1] 3.048672
```

```
SouthAfrican_Int_AG4_sd<-sd(Interior_Total_BR1[45:48,])
print(SouthAfrican_Int_AG4_sd)
```

```
## [1] 2.878917
```

Intraobserver Error Testing
2023-02-22

```
setwd("~/Desktop/xls files")
library(readxl)
Error_Test_Madison <- read_excel("~/Desktop/xls files/Error Test_Madison.xlsx")
View(Error_Test_Madison)

library(irr)

## Loading required package: lpSolve

UCT_195_Ext_icc_Intra<-icc(Error_Test_Madison[1:3], model= "two", type= "agreement", unit
="single", conf.level= 0.95)
print(UCT_195_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.507
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,5.03) = 8.62 , p = 0.0179
##
##  95%-Confidence Interval for ICC Population Values:
##   0.023 < ICC < 0.915

UCT_195_Int_icc_Intra<-icc(Error_Test_Madison[4:6], model= "two", type= "agreement", unit
="single", conf.level= 0.95)
print(UCT_195_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.318
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,6.7) = 4.08 , p = 0.054
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   -0.045 < ICC < 0.846
```

```
SAM_AP_3027_Ext_icc_Intra<-icc(Error_Test_Madison[7:9], model= "two", type= "agreement
", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.5
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,9.98) = 4.35 , p = 0.0272
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.008 < ICC < 0.918
```

```
SAM_AP_3027_Int_icc_Intra<-icc(Error_Test_Madison[10:12], model= "two", type= "agreeme
nt", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.926
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,5.11) = 76.4 , p = 9.76e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.615 < ICC < 0.992
```

```
SAM_AP_6340_Ext_icc_Intra<-icc(Error_Test_Madison[13:15], model= "two", type= "agreem
ent", unit="single", conf.level= 0.95)
print(SAM_AP_6340_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.211
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,8.15) = 2.7 , p = 0.107
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.078 < ICC < 0.786
```

```
SAM_AP_6340_Int_icc_Intra<-icc(Error_Test_Madison[16:18], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_6340_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.494
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,8.21) = 3.39 , p = 0.0648
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.149 < ICC < 0.923
```

```
SAM_AP_4844_Ext_icc_Intra<-icc(Error_Test_Madison[19:21], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_4844_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.571
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,5.89) = 9 , p = 0.0109
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.066 < ICC < 0.932

SAM_AP_4844_Int_icc_Intra<-icc(Error_Test_Madison[22:24], model= "two", type= "agreeme
nt", unit="single", conf.level= 0.95)
print(SAM_AP_4844_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.58
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.38) = 12.4 , p = 0.0122
##
##  95%-Confidence Interval for ICC Population Values:
##   0.052 < ICC < 0.934

KAL_0869_Ext_icc_Intra<-icc(Error_Test_Madison[25:27], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0869_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.336
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,6.01) = 4.67 , p = 0.0469
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.036 < ICC < 0.852

KAL_0869_Int_icc_Intra<-icc(Error_Test_Madison[28:30], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0869_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.342
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,6.69) = 4.33 , p = 0.0475
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.039 < ICC < 0.857
```

```
KAL_0707_Ext_icc_Intra<-icc(Error_Test_Madison[31:33], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0707_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.516
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,9.93) = 4.64 , p = 0.0226
##
##  95%-Confidence Interval for ICC Population Values:
##   0.01 < ICC < 0.921
```

```
KAL_0707_Int_icc_Intra<-icc(Error_Test_Madison[34:36], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0707_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.97
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##     F(4,10) = 98.2 , p = 5.55e-08
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.874 < ICC < 0.997

KAL_0674_Ext_icc_Intra<-icc(Error_Test_Madison[37:39], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0674_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.63
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.78) = 17.7 , p = 0.00996
##
##  95%-Confidence Interval for ICC Population Values:
##   0.067 < ICC < 0.945

KAL_0674_Int_icc_Intra<-icc(Error_Test_Madison[40:42], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0674_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.0805
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,6.5) = 2.39 , p = 0.155
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.039 < ICC < 0.568

KAL_0401_Ext_icc_Intra<-icc(Error_Test_Madison[43:45], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0401_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.693
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.18) = 19.6 , p = 0.00587
##
##  95%-Confidence Interval for ICC Population Values:
##   0.123 < ICC < 0.958
```

```
KAL_0401_Int_icc_Intra<-icc(Error_Test_Madison[46:48], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0401_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.868
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.63) = 45.3 , p = 0.000632
##
##  95%-Confidence Interval for ICC Population Values:
##   0.407 < ICC < 0.984
```

```
E1906_07_37_Ext_icc_Intra<-icc(Error_Test_Madison[49:51], model= "two", type= "agreement
", unit="single", conf.level= 0.95)
print(E1906_07_37_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.92
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,7.14) = 52.2 , p = 2.31e-05
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.658 < ICC < 0.991

E1906_07_37_Int_icc_Intra<-icc(Error_Test_Madison[52:54], model= "two", type= "agreement
", unit="single", conf.level= 0.95)
print(E1906_07_37_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.755
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,7.23) = 15.2 , p = 0.00128
##
##  95%-Confidence Interval for ICC Population Values:
##   0.283 < ICC < 0.967

E1892_92_286_180_Ext_icc_Intra<-icc(Error_Test_Madison[55:57], model= "two", type= "agr
eement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.607
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,2.68) = 33.8 , p = 0.0118
##
##  95%-Confidence Interval for ICC Population Values:
##   0.036 < ICC < 0.941

E1892_92_286_180_Int_icc_Intra<-icc(Error_Test_Madison[58:60], model= "two", type= "agre
ement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.762
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,5.64) = 19.4 , p = 0.00184
##
##  95%-Confidence Interval for ICC Population Values:
##   0.254 < ICC < 0.969
```

```
E1902_144_594_Ext_icc_Intra<-icc(Error_Test_Madison[61:63], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.805
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.68) = 39.6 , p = 0.00264
##
##  95%-Confidence Interval for ICC Population Values:
##   0.223 < ICC < 0.976
```

```
E1902_144_594_Int_icc_Intra<-icc(Error_Test_Madison[64:66], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.789
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,9.28) = 11 , p = 0.00147
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.339 < ICC < 0.973

E1879_73_121_Ext_icc_Intra<-icc(Error_Test_Madison[67:69], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.519
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.38) = 10.3 , p = 0.0174
##
##  95%-Confidence Interval for ICC Population Values:
##   0.024 < ICC < 0.918

E1879_73_121_Int_icc_Intra<-icc(Error_Test_Madison[70:72], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 3
##    ICC(A,1) = 0.722
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##     F(4,10) = 9.13 , p = 0.00226
##
##  95%-Confidence Interval for ICC Population Values:
##   0.25 < ICC < 0.962
```

Intraobserver Error Test 2-3
2023-02-24

```
setwd("~/Desktop/xls files")
library(readxl)
Error_Test_Madison <- read_excel("~/Desktop/xls files/Error Test_Madison.xlsx")
View(Error_Test_Madison)

library(irr)

## Loading required package: lpSolve

UCT_195_Ext_icc_Intra<-icc(Error_Test_Madison[2:3], model= "two", type= "agreement", unit
="single", conf.level= 0.95)
print(UCT_195_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.999
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.05) = 2401 , p = 1.17e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.98 < ICC < 1

UCT_195_Int_icc_Intra<-icc(Error_Test_Madison[5:6], model= "two", type= "agreement", unit
="single", conf.level= 0.95)
print(UCT_195_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.999
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.98) = 2329 , p = 2.52e-08
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.993 < ICC < 1
```

```
SAM_AP_3027_Ext_icc_Intra<-icc(Error_Test_Madison[8:9], model= "two", type= "agreement
", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.989
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.39) = 233 , p = 2.46e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.914 < ICC < 0.999
```

```
SAM_AP_3027_Int_icc_Intra<-icc(Error_Test_Madison[11:12], model= "two", type= "agreeme
nt", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.999
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,3.6) = 4638 , p = 5.81e-07
##
##  95%-Confidence Interval for ICC Population Values:
##   0.993 < ICC < 1
```

```
SAM_AP_6340_Ext_icc_Intra<-icc(Error_Test_Madison[14:15], model= "two", type= "agreem
ent", unit="single", conf.level= 0.95)
print(SAM_AP_6340_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.997
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##      F(4,4) = 566 , p = 9.3e-06
##
##  95%-Confidence Interval for ICC Population Values:
##   0.973 < ICC < 1
```

```
SAM_AP_6340_Int_icc_Intra<-icc(Error_Test_Madison[17:18], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_6340_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.998
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,2.64) = 1783 , p = 6.74e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.96 < ICC < 1
```

```
SAM_AP_4844_Ext_icc_Intra<-icc(Error_Test_Madison[20:21], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_4844_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.985
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.84) = 149 , p = 2.89e-05
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.894 < ICC < 0.998

SAM_AP_4844_Int_icc_Intra<-icc(Error_Test_Madison[23:24], model= "two", type= "agreeme
nt", unit="single", conf.level= 0.95)
print(SAM_AP_4844_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.948
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##      F(4,5) = 37.7 , p = 0.000631
##
##  95%-Confidence Interval for ICC Population Values:
##   0.669 < ICC < 0.994

KAL_0869_Ext_icc_Intra<-icc(Error_Test_Madison[26:27], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0869_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.998
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.32) = 1000 , p = 1.25e-06
##
##  95%-Confidence Interval for ICC Population Values:
##   0.978 < ICC < 1

KAL_0869_Int_icc_Intra<-icc(Error_Test_Madison[29:30], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0869_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.992
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,1.76) = 816 , p = 0.00247
##
##  95%-Confidence Interval for ICC Population Values:
##   0.675 < ICC < 0.999
```

```
KAL_0707_Ext_icc_Intra<-icc(Error_Test_Madison[32:33], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0707_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.99
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,2.68) = 383 , p = 0.000475
##
##  95%-Confidence Interval for ICC Population Values:
##   0.837 < ICC < 0.999
```

```
KAL_0707_Int_icc_Intra<-icc(Error_Test_Madison[35:36], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0707_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.996
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.23) = 615 , p = 4.45e-06
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.964 < ICC < 1

KAL_0674_Ext_icc_Intra<-icc(Error_Test_Madison[38:39], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0674_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 1
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.24) = 8278 , p = 1.76e-08
##
##  95%-Confidence Interval for ICC Population Values:
##   0.997 < ICC < 1

KAL_0674_Int_icc_Intra<-icc(Error_Test_Madison[41:42], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0674_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.874
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.19) = 12.4 , p = 0.0139
##
##  95%-Confidence Interval for ICC Population Values:
##   0.185 < ICC < 0.986

KAL_0401_Ext_icc_Intra<-icc(Error_Test_Madison[44:45], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0401_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.992
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##      F(4,5) = 238 , p = 6.91e-06
##
##  95%-Confidence Interval for ICC Population Values:
##   0.939 < ICC < 0.999
```

```
KAL_0401_Int_icc_Intra<-icc(Error_Test_Madison[47:48], model= "two", type= "agreement",
unit="single", conf.level= 0.95)
print(KAL_0401_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.99
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.79) = 183 , p = 1.95e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.926 < ICC < 0.999
```

```
E1906_07_37_Ext_icc_Intra<-icc(Error_Test_Madison[50:51], model= "two", type= "agreement
", unit="single", conf.level= 0.95)
print(E1906_07_37_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.999
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.78) = 2631 , p = 8.99e-07
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.989 < ICC < 1

E1906_07_37_Int_icc_Intra<-icc(Error_Test_Madison[53:54], model= "two", type= "agreement
", unit="single", conf.level= 0.95)
print(E1906_07_37_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.996
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,4.5) = 414 , p = 5.39e-06
##
##  95%-Confidence Interval for ICC Population Values:
##   0.966 < ICC < 1

E1892_92_286_180_Ext_icc_Intra<-icc(Error_Test_Madison[56:57], model= "two", type= "agr
eement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Ext_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.98
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##     F(4,4) = 81.3 , p = 0.000439
##
##  95%-Confidence Interval for ICC Population Values:
##   0.824 < ICC < 0.998

E1892_92_286_180_Int_icc_Intra<-icc(Error_Test_Madison[59:60], model= "two", type= "agre
ement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Int_icc_Intra)

##  Single Score Intraclass Correlation
##
##    Model: twoway
```

```
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.978
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,4.7) = 103 , p = 8.7e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.84 < ICC < 0.998
```

```
E1902_144_594_Ext_icc_Intra<-icc(Error_Test_Madison[62:63], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.974
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,1.34) = 446 , p = 0.0135
##
##  95%-Confidence Interval for ICC Population Values:
##   0.115 < ICC < 0.998
```

```
E1902_144_594_Int_icc_Intra<-icc(Error_Test_Madison[65:66], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.976
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.09) = 139 , p = 0.000832
##
```

```
##  95%-Confidence Interval for ICC Population Values:
##   0.723 < ICC < 0.998
```

```
E1879_73_121_Ext_icc_Intra<-icc(Error_Test_Madison[68:69], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Ext_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.998
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.95) = 971 , p = 2.35e-07
##
##  95%-Confidence Interval for ICC Population Values:
##   0.985 < ICC < 1
```

```
E1879_73_121_Int_icc_Intra<-icc(Error_Test_Madison[71:72], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Int_icc_Intra)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.999
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.54) = 3311 , p = 1.31e-06
##
##  95%-Confidence Interval for ICC Population Values:
##   0.99 < ICC < 1
```

Interobserver Error Testing
2023-02-21

```
library(nortest)
library(fBasics)

library(nortest)
library(fBasics)
library(DescTools)
library(epitools)
library(Hmisc)
library(psych)

library(readxl)
Error_Test_Madison_Alexandra <-
  read_excel("~/Desktop/xls files/Error Test Madison-Alexandra.xlsx")
View(Error_Test_Madison_Alexandra)

library(irr)

## Loading required package: lpSolve

UCT_195_Ext_icc<-icc(Error_Test_Madison_Alexandra[1:2], model= "two",
                     type= "agreement", unit="single", conf.level= 0.95)
print(UCT_195_Ext_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.287
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.92) = 2.2 , p = 0.207
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.313 < ICC < 0.868

UCT_195_Int_icc<-icc(Error_Test_Madison_Alexandra[3:4], model= "two",
                     type= "agreement", unit="single", conf.level= 0.95)
print(UCT_195_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.944
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.79) = 38.8 , p = 0.000745
##
##  95%-Confidence Interval for ICC Population Values:
##   0.64 < ICC < 0.994
```

```
SAM_AP_3027_Ext_icc<-icc(Error_Test_Madison_Alexandra[5:6], model= "two",
                         type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Ext_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = -0.204
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.99) = 0.728 , p = 0.617
##
##  95%-Confidence Interval for ICC Population Values:
##   -1.353 < ICC < 0.788
```

```
SAM_AP_3027_Int_icc<-icc(Error_Test_Madison_Alexandra[7:8], model= "two",
                         type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.952
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
```

```
##   F(4,4.45) = 34.7 , p = 0.0014
##
##  95%-Confidence Interval for ICC Population Values:
##   0.647 < ICC < 0.995

SAM_AP_6340_Ext_icc<-icc(Error_Test_Madison_Alexandra[9:10], model= "two",
                    type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_6340_Ext_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.98
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.08) = 80.9 , p = 0.000395
##
##  95%-Confidence Interval for ICC Population Values:
##   0.825 < ICC < 0.998

SAM_AP_6340_Int_icc<-icc(Error_Test_Madison_Alexandra[11:12], model= "two",
                    type= "agreement", unit="single", conf.level= 0.95)

print(SAM_AP_6340_Int_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.0904
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,2.34) = 3.11 , p = 0.23
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.043 < ICC < 0.595

SAM_AP_4844_Ext_icc<-icc(Error_Test_Madison_Alexandra[13:14], model= "two",
                    type= "agreement", unit="single", conf.level= 0.95)

print(SAM_AP_4844_Ext_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.947
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.83) = 41 , p = 0.000625
##
##  95%-Confidence Interval for ICC Population Values:
##   0.662 < ICC < 0.994
```

```
SAM_AP_4844_Int_icc<-icc(Error_Test_Madison_Alexandra[15:16], model= "two",
                 type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_4844_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.97
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,1.37) = 350 , p = 0.0146
##
##  95%-Confidence Interval for ICC Population Values:
##   0.099 < ICC < 0.997
```

```
KAL_0869_Ext_icc<-icc(Error_Test_Madison_Alexandra[17:18], model= "two",
                 type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0869_Ext_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.938
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
```

```
##   F(4,1.87) = 88.6 , p = 0.0142
##
##  95%-Confidence Interval for ICC Population Values:
##   0.127 < ICC < 0.994

KAL_0869_Int_icc<-icc(Error_Test_Madison_Alexandra[19:20], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0869_Int_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = -0.0418
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,3.7) = 0.903 , p = 0.543
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.605 < ICC < 0.761

KAL_0707_Ext_icc<-icc(Error_Test_Madison_Alexandra[21:22], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0707_Ext_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.381
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,2.21) = 5.67 , p = 0.138
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.12 < ICC < 0.885

KAL_0707_Int_icc<-icc(Error_Test_Madison_Alexandra[23:24], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0707_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.771
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.85) = 8.75 , p = 0.019
##
##  95%-Confidence Interval for ICC Population Values:
##   0.06 < ICC < 0.972
```

```
KAL_0674_Ext_icc<-icc(Error_Test_Madison_Alexandra[25:26], model= "two",
                type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0674_Ext_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.971
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.03) = 53.9 , p = 0.000942
##
##  95%-Confidence Interval for ICC Population Values:
##   0.744 < ICC < 0.997
```

```
KAL_0674_Int_icc<-icc(Error_Test_Madison_Alexandra[27:28], model= "two",
                type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0674_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.881
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
```

```
##   F(4,4.21) = 13.2 , p = 0.0123
##
##  95%-Confidence Interval for ICC Population Values:
##   0.22 < ICC < 0.987

KAL_0401_Ext_icc<-icc(Error_Test_Madison_Alexandra[29:30], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0401_Ext_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.986
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.24) = 118 , p = 0.000138
##
##  95%-Confidence Interval for ICC Population Values:
##   0.882 < ICC < 0.999

KAL_0401_Int_icc<-icc(Error_Test_Madison_Alexandra[31:32], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0401_Int_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.954
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(4,4.8) = 38.4 , p = 0.000757
##
##  95%-Confidence Interval for ICC Population Values:
##   0.686 < ICC < 0.995

E1906_07_37_Ext_icc<-icc(Error_Test_Madison_Alexandra[33:34], model= "two",
                         type= "agreement", unit="single", conf.level= 0.95)

print(E1906_07_37_Ext_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.821
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,4.97) = 10.8 , p = 0.0114
##
##  95%-Confidence Interval for ICC Population Values:
##   0.176 < ICC < 0.979
```

```
E1906_07_37_Int_icc<-icc(Error_Test_Madison_Alexandra[35:36], model= "two",
                         type= "agreement", unit="single", conf.level= 0.95)
```

```
print(E1906_07_37_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.99
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.94) = 259 , p = 5.04e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.904 < ICC < 0.999
```

```
E1892_92_286_180_Ext_icc<-icc(Error_Test_Madison_Alexandra[37:38], model= "two",
                              type= "agreement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Ext_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.888
##
```

```
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,1.54) = 68 , p = 0.0325
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.036 < ICC < 0.989

E1892_92_286_180_Int_icc<-icc(Error_Test_Madison_Alexandra[39:40], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Int_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.766
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,3.16) = 12.6 , p = 0.0284
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.024 < ICC < 0.972

E1902_144_594_Ext_icc<-icc(Error_Test_Madison_Alexandra[41:42], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Ext_icc)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.659
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,1.88) = 14.4 , p = 0.0746
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.112 < ICC < 0.957

E1902_144_594_Int_icc<-icc(Error_Test_Madison_Alexandra[43:44], model= "two",
                      type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.694
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,1.66) = 19.8 , p = 0.0717
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.096 < ICC < 0.963
```

```
E1879_73_121_Ext_icc<-icc(Error_Test_Madison_Alexandra[45:46], model= "two",
                    type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Ext_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.918
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(4,2.75) = 43 , p = 0.00777
##
##  95%-Confidence Interval for ICC Population Values:
##   0.273 < ICC < 0.991
```

```
E1879_73_121_Int_icc<-icc(Error_Test_Madison_Alexandra[47:48], model= "two",
                    type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Int_icc)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 5
##      Raters = 2
##    ICC(A,1) = 0.902
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
```

```
##    F(4,4.7) = 22.6 , p = 0.00276
##
##  95%-Confidence Interval for ICC Population Values:
##   0.443 < ICC < 0.989
```

3-Way Error Test
2023-02-23

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
setwd("~/Desktop")
library(readxl)
Error_Test_Mel_Mad_Alex <- read_excel("Error Test Mel_Mad_Alex.xlsx")
View(Error_Test_Mel_Mad_Alex)
library(irr)

## Loading required package: lpSolve

UCT_195_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[1:3], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(UCT_195_Ext_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.161
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.8) = 1.88 , p = 0.173
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.128 < ICC < 0.711

UCT_195_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[4:6], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(UCT_195_Int_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
```

```
##    ICC(A,1) = 0.0902
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.1) = 1.71 , p = 0.212
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.087 < ICC < 0.588
```

```
SAM_AP_3027_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[7:9], model= "two", type= "ag
reement", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Ext_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = -0.0733
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,8.95) = 0.789 , p = 0.583
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.365 < ICC < 0.581
```

```
SAM_AP_3027_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[10:12], model= "two", type= "
agreement", unit="single", conf.level= 0.95)
print(SAM_AP_3027_Int_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.103
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,10.9) = 1.8 , p = 0.194
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.086 < ICC < 0.608
```

```
SAM_AP_6340_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[13:15], model= "two", type= "
agreement", unit="single", conf.level= 0.95)
print(SAM_AP_6340_Ext_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.914
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,10.1) = 27.7 , p = 1.36e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.689 < ICC < 0.986

SAM_AP_6340_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[16:18], model= "two", type= "
agreement", unit="single", conf.level= 0.95)
print(SAM_AP_6340_Int_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.144
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,3.03) = 7.39 , p = 0.0641
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.013 < ICC < 0.596

SAM_AP_4844_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[19:21], model= "two", type= "
agreement", unit="single", conf.level= 0.95)
print(SAM_AP_4844_Ext_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
```

```
##      Raters = 3
##    ICC(A,1) = 0.954
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.9) = 65.5 , p = 3.03e-08
##
##  95%-Confidence Interval for ICC Population Values:
##   0.834 < ICC < 0.993
```

```
SAM_AP_4844_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[22:24], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(SAM_AP_4844_Int_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.962
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,7.17) = 123 , p = 8.86e-07
##
##  95%-Confidence Interval for ICC Population Values:
##   0.826 < ICC < 0.994
```

```
KAL_0869_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[25:27], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0869_Ext_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.257
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.8) = 2.36 , p = 0.105
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.113 < ICC < 0.786
```

```
KAL_0869_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[28:30], model= "two", type= "agree
ment", unit="single", conf.level= 0.95)
print(KAL_0869_Int_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.331
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.5) = 2.95 , p = 0.0599
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.069 < ICC < 0.823

KAL_0707_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[31:33], model= "two", type= "agre
ement", unit="single", conf.level= 0.95)
print(KAL_0707_Ext_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = -0.0842
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,6.91) = 0.699 , p = 0.642
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.289 < ICC < 0.495

KAL_0707_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[34:36], model= "two", type= "agree
ment", unit="single", conf.level= 0.95)
print(KAL_0707_Int_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
```

```
##      Raters = 3
##    ICC(A,1) = 0.44
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.6) = 3.88 , p = 0.0264
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.005 < ICC < 0.869
```

```
KAL_0674_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[37:39], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0674_Ext_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.643
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,10.6) = 7.86 , p = 0.00253
##
##  95%-Confidence Interval for ICC Population Values:
##   0.193 < ICC < 0.93
```

```
KAL_0674_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[40:42], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(KAL_0674_Int_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = -0.094
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##  F(5,0.573) = 0.314 , p = 0.868
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.143 < ICC < 0.25
```

```
KAL_0401_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[43:45], model= "two", type= "agre
ement", unit="single", conf.level= 0.95)
print(KAL_0401_Ext_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.905
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,10.7) = 25.8 , p = 1.24e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.671 < ICC < 0.985

KAL_0401_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[46:48], model= "two", type= "agree
ment", unit="single", conf.level= 0.95)
print(KAL_0401_Int_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.534
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,10.9) = 5.42 , p = 0.0096
##
##  95%-Confidence Interval for ICC Population Values:
##   0.079 < ICC < 0.9

E1906_07_37_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[49:51], model= "two", type= "a
greement", unit="single", conf.level= 0.95)
print(E1906_07_37_Ext_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
```

```
##      Raters = 3
##    ICC(A,1) = 0.88
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.5) = 25.7 , p = 7.28e-06
##
##  95%-Confidence Interval for ICC Population Values:
##   0.62 < ICC < 0.98
```

```
E1906_07_37_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[52:54], model= "two", type= "ag
reement", unit="single", conf.level= 0.95)
print(E1906_07_37_Int_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.755
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.8) = 11.1 , p = 0.000393
##
##  95%-Confidence Interval for ICC Population Values:
##   0.358 < ICC < 0.956
```

```
E1892_92_286_180_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[55:57], model= "two", typ
e= "agreement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Ext_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.744
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.5) = 11 , p = 0.000464
##
##  95%-Confidence Interval for ICC Population Values:
##   0.34 < ICC < 0.954
```

```
E1892_92_286_180_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[58:60], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1892_92_286_180_Int_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.638
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,9.99) = 8.11 , p = 0.00272
##
##  95%-Confidence Interval for ICC Population Values:
##   0.185 < ICC < 0.929

E1902_144_594_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[61:63], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Ext_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.251
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##   F(5,11.8) = 2.35 , p = 0.106
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.11 < ICC < 0.78

E1902_144_594_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[64:66], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1902_144_594_Int_icc_Three)

##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
```

```
##      Raters = 3
##    ICC(A,1) = -0.263
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(5,8.3) = 0.443 , p = 0.808
##
##  95%-Confidence Interval for ICC Population Values:
##   -0.512 < ICC < 0.426
```

```
E1879_73_121_Ext_icc_Three<-icc(Error_Test_Mel_Mad_Alex[67:69], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Ext_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.925
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##    F(5,7.9) = 56.2 , p = 5.75e-06
##
##  95%-Confidence Interval for ICC Population Values:
##   0.702 < ICC < 0.988
```

```
E1879_73_121_Int_icc_Three<-icc(Error_Test_Mel_Mad_Alex[70:72], model= "two", type= "agreement", unit="single", conf.level= 0.95)
print(E1879_73_121_Int_icc_Three)
```

```
##  Single Score Intraclass Correlation
##
##    Model: twoway
##    Type : agreement
##
##    Subjects = 6
##      Raters = 3
##    ICC(A,1) = 0.842
##
##  F-Test, H0: r0 = 0 ; H1: r0 > 0
##     F(5,12) = 17.5 , p = 3.87e-05
##
##  95%-Confidence Interval for ICC Population Values:
##   0.531 < ICC < 0.973
```

Manova Test
2023-03-20

```
library(missMDA)
library(Morpho)
library(geomorph)

## Loading required package: RRPP

##
## Attaching package: 'RRPP'

## The following object is masked from 'package:Morpho':
##
##     classify

## Loading required package: rgl

## Loading required package: Matrix

library(ggplot2)
library(scales)
library(readr)

##
## Attaching package: 'readr'

## The following object is masked from 'package:scales':
##
##     col_factor

library(RColorBrewer)
library(fields)

## Loading required package: spam

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following object is masked from 'package:Matrix':
##
##     det
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':
##
##     viridis_pal

##
## Try help(fields) to get started.

library(raster)

## Loading required package: sp

library(colorRamps)
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4

library(proxy)

##
## Attaching package: 'proxy'

## The following object is masked from 'package:raster':
##
##     as.matrix

## The following object is masked from 'package:spam':
##
##     as.matrix

## The following object is masked from 'package:Matrix':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix

library(base)

read.table("~/Desktop/AG1_NA_Test.txt")

##         MMK_146   MMK_238    UCT_195     UCT_189 SAM_AP_1448 SAM_AP_3027
## 1   0.000000000  3.668905  2.4208326  6.61873156  16.5967815   12.555374
## 2   0.000000000  0.000000  2.1575182  6.61873156  14.8769430   11.492452
## 3   0.000000000  0.000000  0.0000000  6.61873156   0.0000000    0.000000
## 4   0.000000000  0.000000  0.0000000  0.00000000   3.8000000    0.000000
## 5   0.000000000 11.980000  0.0000000  0.00000000   2.4381538    0.000000
## 6   0.465438915  0.000000  0.0000000  0.00000000   0.4129231    0.000000
## 7   0.572112948  0.000000  0.0000000  0.00000000   0.0000000    0.000000
## 8   0.197931045  0.000000  0.0000000  0.00000000   0.0000000    0.000000
## 9   0.306076452  0.000000  0.0000000  0.00000000   0.2963077    0.000000
## 10  0.548171586  0.000000  0.0000000  0.00000000   0.6255385    0.000000
## 11  0.607922700  0.000000  0.0000000  0.00000000   1.1584615    0.000000
## 12  0.498320191  0.000000  0.0000000  0.00000000   5.3289231    0.000000
## 13  0.530585704  0.000000  0.0000000  0.00000000   0.0000000    0.000000
## 14  0.879418506  0.000000  0.0000000  0.00000000   0.0000000    0.000000
## 15  0.000000000  0.000000  1.9130977  6.61873156  12.9896468   10.293224
## 16  0.000000000  0.000000  1.6712261  6.61873156  28.3680000    0.000000
## 17  0.000000000  1.200000  0.0000000  6.61873156   4.0576695    0.000000
## 18  0.000000000  4.982400  0.0000000  0.00000000   4.2664769    0.000000
## 19  0.000000000  7.188000  0.0000000  0.00000000   2.7133538    0.000000
## 20  0.000000000  0.000000  0.0000000  0.00000000   2.9105692    0.000000
## 21  0.000000000  0.000000  0.0000000  0.00000000   4.7456000    0.000000
## 22  0.000000000  0.000000  0.0000000  0.00000000   4.3753846    0.000000
## 23  0.000000000  0.000000  0.0000000  0.00000000   2.9043692    2.559000
## 24  0.000000000  0.000000  0.0000000  0.00000000   2.6462769    8.757000
## 25  0.000000000  0.000000  0.0000000  0.00000000   2.1657231    0.000000
## 26  0.000000000  0.000000  0.0000000  0.00000000   3.1973539    5.958000
## 27  7.056000000  0.000000  0.0000000  0.00000000   0.0000000    0.000000
## 28  0.000000000  0.000000  0.0000000  0.00000000   0.0000000   20.368475
## 29  0.000000000  0.000000  0.0000000  6.61873156  13.2869055   10.557305
## 30  0.000000000  0.000000  0.0000000  6.61873156  28.3680000   31.271033
## 31  0.000000000  3.000000  2.4876923  6.61873156   8.9736923   38.707667
## 32  0.000000000 12.456000  8.9570833  0.00000000   4.9661923   27.764333
## 33  0.000000000  0.000000 14.5407500  0.00000000   3.1261538    1.842750
## 34  0.000000000  0.000000 10.1992917  0.00000000   6.6570385    7.116667
## 35  0.000000000  0.000000  5.2360000  0.00000000  11.8640000    0.000000
## 36  0.000000000  0.000000  0.4363333  0.00000000  10.9384615    0.000000
## 37  0.000000000  0.000000  0.0000000  0.00000000   6.8164615    1.421667
## 38  0.000000000  0.000000  0.0000000  0.00000000   5.6773846    4.865000
```

```
## 39  0.000000000  0.000000  0.0000000  0.00000000   3.6766154    0.000000
## 40  0.000000000  0.000000  0.0000000  0.00000000   0.0000000    3.310000
## 41  7.056000000  0.000000  0.0000000  0.00000000   0.0000000    0.000000
## 42  0.000000000  0.000000  0.0000000  0.00000000   0.0000000   11.315819
## 43  0.000000000  4.652000  0.0000000  0.00000000   0.0000000    0.000000
## 44  0.000000000  0.000000  0.0000000  0.00000000   0.0000000   52.667699
## 45  0.000000000  5.028000  2.3581250  0.00000000  11.0330769   66.217667
## 46  0.000000000  8.425000  9.1037500  0.00000000  53.1215385   28.059333
## 47  0.000000000  0.000000 14.8723907  0.00000000  43.0989231    3.587750
## 48  0.000000000  0.000000 10.4473076  1.37330313  49.0121538   16.160000
## 49  0.000000000  0.000000  5.2360000  2.05995469  31.4420000    0.000000
## 50  0.000000000  0.000000  0.4363333  1.65128568   0.0000000    0.000000
## 51  0.000000000  0.000000  1.7339704  0.83394766   0.8266154    0.000000
## 52  0.000000000  0.000000  0.9633169  0.83394766   1.7450769    0.000000
## 53  0.000000000 24.904000  0.0000000  0.00000000   5.1876923    0.000000
## 54  0.000000000  0.000000  0.0000000  0.00000000  26.5763077    0.000000
## 55  0.000000000 19.152000  0.0000000  0.00000000  23.1247692    0.000000
## 56  0.000000000  0.000000  0.0000000  0.00000000  13.4440000    0.000000
## 57  0.000000000 11.599344  0.0000000  0.00000000   0.0000000    4.255833
## 58  0.000000000  0.000000  0.0000000  0.00000000   0.0000000   55.625519
## 59  0.000000000  3.048000  0.0000000  0.00000000  32.9435078   65.172000
## 60  0.000000000  4.001742  1.1733333  0.00000000  54.9056616    9.985000
## 61  0.000000000 12.797010  2.6531255  0.00000000  41.6268923   10.617000
## 62  0.000000000  0.000000  1.9841270  1.37330313  48.6826998   20.541000
## 63  0.000000000  0.000000  0.0000000  2.05995469  43.5940473    0.000000
## 64  0.000000000  0.000000  0.0000000  1.65128568   0.0000000    0.000000
## 65  0.000000000  0.000000 13.8717633  0.83394766   0.8266154    0.000000
## 66  0.000000000  0.000000  7.7065351  0.83394766   1.7450769    0.000000
## 67  0.000000000 16.998400  0.0000000  0.00000000   5.1876923    0.000000
## 68  0.000000000  0.000000  0.0000000  0.00000000  26.5763077    0.000000
## 69  0.000000000  7.660800  0.0000000  0.00000000  23.1247692    0.000000
## 70  0.000000000  5.520000  0.0000000  0.00000000  13.4440000   10.070816
## 71 -0.369244672 16.230907  0.0000000  1.43732370   0.0000000   17.023331
## 72  0.000000000  0.000000  0.0000000  1.43732371   0.0000000   68.446077
## 73  0.000000000  1.382400  0.0000000  1.43732371  40.9900616   62.616000
## 74  0.000000000  2.058323  2.3124766 17.20748339  52.2052923   36.389761
## 75 -1.063131998 21.328350  4.9662471 17.20748339  38.1383384   30.191216
## 76 -0.256454214  0.000000  1.9841270 11.14847189  45.7408924   15.631000
## 77  0.090045166  0.000000  0.0000000  2.05995469  59.7967772   23.549441
## 78  0.366070177  0.000000  0.0000000  1.65128568  10.6920491    8.970444
## 79  0.002355274  0.000000 13.8717633  0.83394766  25.0831004   18.247809
## 80  0.257311850  0.000000  7.7065351  0.83394766  24.9448683   18.381012
## 81  0.080741167 11.728000  0.0000000  0.00705298  18.0391419   13.618744
## 82  0.463130858  0.000000  0.0000000  0.00705298  12.2194453   10.075037
## 83 -0.137706295  0.000000  0.0000000 13.62349232  16.2898638   12.258838
## 84 -0.781361627  9.200000  0.0000000 20.43171199  19.3255242   13.714593
```

```
## 85 -0.198292608  4.268675  0.6147699  2.15598555   0.0000000   -3.013588
## 86  0.397683840  1.974707  1.0359420  2.15598555   0.0000000    6.261522
## 87  0.000000000  0.000000  0.0000000  2.15598555  14.7484616   16.119231
## 88  0.000000000  6.080000  4.2110488 25.81122503  36.6461539   20.701131
## 89 -1.482107954  9.333686 10.1065172 25.81122503  28.1095384   33.350597
## 90 -1.889665995 10.903995  5.7866536 25.81122503  20.0040000    0.000000
## 91  0.131167878  3.013634  1.5136137  2.39498370   8.4109404    7.241256
## 92  0.135592146  3.006002  1.9951524  3.15557384  14.0302503   10.993282
## 93  0.135592146  3.006002  1.9951524  3.15557384  14.0302503   10.993282
## 94  0.135592146  3.006002  1.9951524  3.15557384  14.0302503   10.993282
## 95  0.298294787  2.369625  1.5949253  0.01057947  11.6641929    9.558951
## 96  0.298294787  2.369625  1.5949253  0.01057947  11.6641929    9.558951
## 97 -0.758379067  6.502585  4.1942638 20.43523845  27.0312604   18.874686
## 98 -1.286715991  8.569065  5.4939330 30.64756789  34.7147941   23.532553
##    SAM_AP_6052 SAM_AP_6340 SAM_AP_1273 SAM_AP_3691 SAM_AP_3737A SAM_
AP_6348A
## 1    6.0006841   3.4396610 10.35974006   4.1564936   3.36807189    0.9500262
## 2    5.0694440   3.2613554 10.87411279   0.6350000   3.41619391    0.0000000
## 3    0.0000000   0.0000000 60.64043837   8.9144427  10.11403551    0.0000000
## 4    0.0000000   0.0000000 53.58670689  25.2828283   0.00000000    0.0000000
## 5    0.0000000   0.8272315 23.51440001   4.1321313  11.38400000    0.0000000
## 6    0.0000000   1.3831858  0.00000000   6.3860210   4.28832785    0.0000000
## 7    0.0000000   0.1271270  0.00000000   0.0000000   3.78281281    0.0000000
## 8    0.0000000   0.5475977 23.80456198  21.3860208   7.49417731    0.0000000
## 9    0.0000000   0.7301303  0.00000000  19.8600862   6.11345965    0.0000000
## 10   0.0000000   0.5862258  0.00000000   2.8525608   0.00000000    0.0000000
## 11   0.0000000   0.0000000  0.00000000   0.0000000   0.00000000    0.0000000
## 12   0.0000000   0.0000000 14.18078889   0.0000000   3.28800000    0.0000000
## 13   0.0000000   0.0000000 14.09746482   0.0000000   0.00000000    0.0000000
## 14   0.0000000   0.0000000 12.13197663   0.0000000   0.00000000   11.3236966
## 15   4.3046569   2.9939179  0.00000000   4.6109544   3.62018922    1.2043810
## 16   1.9271949   0.0000000  5.46333325  13.0129959   0.00000000    0.0000000
## 17   0.2919992   0.0000000 33.80478105  13.2433202  19.84750888    0.0000000
## 18   0.0000000   0.0000000 36.92714318  15.4479975   0.00000000    0.0000000
## 19   3.7995000   1.7725065 23.02546668  11.9234025   2.84600000    0.0000000
## 20   2.4585000   8.4257275  2.57400000   0.0000000   6.78000000    0.0000000
## 21   0.0000000   1.1827246  2.97000000   8.0325650   0.00000000    0.0000000
## 22   9.4875000   2.5041719 10.24986667   9.7474285  10.98400000    2.3910000
## 23  14.6625000   3.3388958 15.05813333   1.8026232  17.82800000    8.7670000
## 24   0.0000000   2.1494948 22.42920002   0.0000000   0.00000000    0.0000000
## 25   0.0000000   0.4659762  6.14599997   0.0000000   4.04700000    0.0000000
## 26   0.0000000   1.1649405  0.00000000   0.0000000   0.82200000   10.6755000
## 27   0.0000000   0.0000000  0.00000000   0.0000000   0.00000000    2.9115000
## 28   0.0000000   0.0000000  0.00000000   0.0000000   0.00000000    0.0000000
## 29   0.0000000   0.0000000  0.51320000   0.0000000   3.24155505    0.0000000
## 30   3.4674997   0.0000000  8.30399995   1.1131279   0.00000000    0.0000000
```

```
## 31  0.7175637  0.0000000 26.31813332  6.6349785  9.62166665  0.0000000
## 32  0.0000000  0.0000000 25.66350003 10.7749088  0.00000000  0.0000000
## 33  2.1108333  3.1905117  5.68023333 17.8690429  0.00000000  0.0000000
## 34  1.3658333 10.1320796  1.31920000  9.8860568  2.82500000  0.0000000
## 35  5.6662516  3.0331448  2.52600000  3.0001621  0.00000000  1.0397500
## 36 11.8088159  0.8933644 12.64306667  0.0000000  4.57666666 13.5167500
## 37  8.1458333  1.1911526 17.95653333 14.8090433  7.42833332  0.0000000
## 38  0.0000000  0.0000000 53.66880007 12.2973250  0.00000000  0.0000000
## 39  1.5350641  0.8387572 72.76986667  1.7608383  2.24833333  0.0000000
## 40  7.4799487  2.0968929 50.03133326  0.0000000  0.00000000  7.5212500
## 41  3.4887821  0.0000000  4.52599996  0.0000000  0.00000000  9.2110000
## 42  0.0000000  0.0000000  0.00000000  0.0000000  0.00000000  0.5874667
## 43  0.0000000  0.0000000  0.00000000  0.0000000  5.23906649  0.0000000
## 44  2.5904487 31.2500001  0.00000000  0.0000000  0.00000000  0.0000000
## 45  0.5631410 12.4999999  1.14053332  0.6874019  0.00000000  0.0000000
## 46  0.0000000  0.0000000  5.68600000  2.0622057  0.00000000  2.5492500
## 47  0.0000000  1.7030765  4.78040000  0.0000000  0.00000000  5.7997500
## 48  0.0000000  7.4905269  0.00000000  2.2216311  4.27499999  1.1955000
## 49  5.6662516 13.1215136  0.00000000  2.5634206  0.00000000  0.0000000
## 50  7.9371617  0.0000000 11.19169205  3.1466080  0.00000000  0.0000000
## 51  0.0000000  0.0000000 34.79928310 11.8385681  0.00000000  0.0000000
## 52  0.0000000  4.5251981 24.07439996  4.6433025  0.00000000  0.0000000
## 53  2.6315384  0.0000000  6.78933332  0.0000000  0.00000000  0.0000000
## 54 12.8227693  2.2163905  0.67733332  0.0000000  0.00000000 10.1200000
## 55  5.9807692  1.6622929  0.00000000  0.0000000  0.00000000  2.7600000
## 56  0.0000000  0.0000000  4.08909677  0.0000000  0.00000000  3.5865356
## 57  0.0000000  0.0000000 30.54419193 23.8126698 26.49075421  0.0000000
## 58  0.0000000 17.3611110  0.00000000  8.6525597  1.47886333  0.0000000
## 59  0.0000000  7.7112356  0.25013333  0.0000000  0.00000000  0.0000000
## 60  0.0000000  0.0000000  0.75040000  0.0000000  0.00000000  0.0000000
## 61  0.0000000  3.6674908  0.00000000  0.0000000  0.00000000  0.0000000
## 62  0.0000000  8.6324509  0.09892009  0.0000000  7.69500000  0.0000000
## 63  0.0000000 17.2960686 -1.77263299  0.0000000  0.00000000  0.0000000
## 64  0.0000000  0.0000000 14.75209900  5.6270363  0.00000000  1.6310541
## 65  5.0012307  0.0000000 13.92548325  5.5204403  4.50082577  0.9926562
## 66  3.0553164  8.0447966 12.58980161  4.8685937  3.73665432  1.3992930
## 67  4.3081257  6.3553888  0.00000000  4.2772485  0.00000000  0.0000000
## 68  3.2508569  4.8956159  0.00000000  0.0000000  0.00000000  0.0000000
## 69  4.9860008  2.3273293  0.00000000  0.0000000  0.00000000  0.0000000
## 70  2.7594137  0.0000000  0.00000000  0.0000000  0.00000000  0.0000000
## 71  0.0000000  0.0000000 24.46235716  9.7925756 27.07340564  0.0000000
## 72  0.0000000  0.0000000  0.00000000  0.0000000  5.42249887  0.0000000
## 73  6.5106536  6.6897363  0.70746666  0.0000000  0.00000000  0.0000000
## 74  4.3412234 11.8439028  5.02917220  0.4797259  0.00000000  0.0000000
## 75  4.1145878  3.0313894  9.68924069  1.8475841  0.00000000  0.0000000
## 76  3.9946042  6.4064321 -3.63827648  2.2768076  0.79455166  0.0000000
```

```
## 77   7.7157560   5.3019146 -3.28506692  -1.1909610  -1.27554364    1.1711037
## 78   2.2986929   2.9683751 11.50449315   4.3890157   3.23649639    1.5795307
## 79   7.0881621   5.2985520  3.50455069   1.4836332   1.06818266    1.0199664
## 80   5.2999131  10.5445117  1.43707706   0.5489839   0.06458768    1.4264967
## 81   5.4342257   9.2442019  8.26994060   3.2871330   2.53542493    1.1348994
## 82   1.8670876   5.3298732  9.49856497   3.5613556   2.45368879    1.7365636
## 83   6.6718600   2.0419392 11.48214131   4.6457017   3.86902059    0.7839012
## 84  11.6101120   0.0000000 14.62907180   6.2029597   5.72123526   -0.2385543
## 85   3.4933980  -1.7527090 29.59990443  11.6913910  33.48257115    0.6544845
## 86   1.4325750   2.1645733 14.40019531   5.4933538   0.00000000    1.6240714
## 87   3.8764147   8.3436801  0.00000000   0.0000000   0.00000000    1.6200319
## 88  52.0946795   4.2198792  0.00000000   0.0000000   0.00000000    0.0000000
## 89  49.3750521   0.1878323  0.00000000   0.0000000   0.00000000    0.0000000
## 90  47.9352494   0.4226227 16.04418767   0.0000000   9.53461975    0.0000000
## 91   3.5664628   2.0332478 15.26727236   5.9693920   4.77201597    1.2015186
## 92   4.4219849   3.2263291 10.89472521   4.2743540   3.32875421    1.2166458
## 93   4.4219849   3.2263291 10.89472521   4.2743540   3.32875421    1.2166458
## 94   4.4219849   3.2263291 10.89472521   4.2743540   3.32875421    1.2166458
## 95   2.9215790   3.0422271 11.33251241   4.3581355   3.26603625    1.4728055
## 96   2.9215790   3.0422271 11.33251241   4.3581355   3.26603625    1.4728055
## 97  12.6660874   4.2380238  8.48877084   3.8138113   3.67318489   -0.1908253
## 98  17.5383415   4.8359221  7.06690006   3.5416493   3.87675921   -1.0226406
##     SAM_AP_34 SAM_AP_1879  SAM_AP_4790 SAM_AP_4844
## 1   3.5722386  0.59869836 8.865218e+00  11.4973010
## 2   3.3894750  0.00000000 7.760778e+00   4.6715821
## 3   1.0561426  0.00000000 3.448410e+00   0.9074296
## 4   0.8743143  0.00000000 1.145378e+01   0.9952031
## 5   1.9291556  0.00000000 8.991403e+00   0.5528906
## 6   2.2719223  0.00000000 1.650877e-01   0.0000000
## 7   1.1350000  0.63525000 0.000000e+00   0.0000000
## 8   1.1350000  0.00000000 0.000000e+00   0.0000000
## 9   1.6008854  0.00000000 0.000000e+00   0.0000000
## 10  2.5333478  0.00000000 0.000000e+00   0.0000000
## 11  2.6828038  0.71170312 0.000000e+00   0.0000000
## 12  2.3915633  0.39539062 0.000000e+00   0.0000000
## 13  2.3730643  0.00000000 0.000000e+00   0.0000000
## 14  2.3524879  0.00000000 0.000000e+00   0.0000000
## 15  3.1730119  0.00000000 6.856317e+00   0.0000000
## 16  2.8052057  0.00000000 0.000000e+00   1.9032372
## 17  1.1016916  0.00000000 2.048146e+00   1.8065884
## 18  1.8780000  0.00000000 5.928845e+00   4.9822031
## 19  1.8780000  0.00000000 5.943158e+00   4.8140625
## 20  2.8644957  0.00000000 2.369591e+00   0.5580469
## 21  2.6743741  2.32925001 4.537778e+00   3.3020313
## 22  2.8430000  0.00000000 0.000000e+00   0.0000000
## 23  5.4377667  0.00000000 0.000000e+00   0.8144531
```

```
## 24  6.9400000  0.00000000 0.000000e+00   2.6543750
## 25  2.6886115  2.60957813 0.000000e+00   2.9363281
## 26  2.9083279  1.44976563 0.000000e+00   2.6657812
## 27  2.7796870  4.19859374 0.000000e+00   1.9846094
## 28  2.9118731  0.34449999 0.000000e+00   0.1249167
## 29  3.2435563  0.00000000 6.293983e+00   4.1965000
## 30  3.6647158  0.00000000 0.000000e+00   3.4090000
## 31  2.6968120  0.00000000 3.321111e+00   6.5619531
## 32  0.7100000  0.00000000 1.176434e+01   9.0895469
## 33  1.2556000  0.00000000 3.082841e+01   5.6112344
## 34  2.7448667  0.00000000 2.646644e+01   9.7518594
## 35  2.2130000  6.87684377 1.815111e+00   6.9974844
## 36  0.9660000  0.00000000 3.160000e-08   6.0397969
## 37  1.9228658  0.00000000 0.000000e+00   6.8925469
## 38  1.6072979  0.00000000 0.000000e+00   9.7146250
## 39  1.6836119  0.00000000 0.000000e+00  12.5314844
## 40  2.3218738  0.00000000 0.000000e+00  13.5099063
## 41  3.2074560  7.55746876 0.000000e+00  12.4714844
## 42  1.1600000  0.62009999 0.000000e+00   2.7842833
## 43  0.4220000  0.00000000 2.784052e+01   1.2208000
## 44  0.4220000  0.00000000 1.088511e+01   3.4090000
## 45  2.3100000  0.00000000 8.130585e+00  11.5625781
## 46  3.7488000  0.00000000 1.623566e+01  20.8182344
## 47  6.6821333  0.00000000 2.930503e+00  12.4103125
## 48  8.7558333  0.00000000 7.852099e+00  21.7823023
## 49 12.4250000 14.54935542 0.000000e+00   4.0516219
## 50  3.0695777  0.00000000 1.778737e+01  13.8052500
## 51  1.9479814  0.00000000 0.000000e+00  12.4035000
## 52  2.4935085  0.00000000 0.000000e+00   7.2217600
## 53  3.1062238  0.00000000 0.000000e+00  13.5383843
## 54  0.9360000  0.00000000 0.000000e+00  23.2244092
## 55  0.9360000  0.00000000 2.858331e+00  11.9038763
## 56  1.3680000  0.00000000 7.155242e-01   2.5594333
## 57  1.6880000  0.00000000 2.126906e+01   0.0000000
## 58  1.6880000  4.07601563 2.500044e+01   9.8209375
## 59  5.7839144  7.33682815 2.999251e+00  30.6643282
## 60  5.1560000  0.00000000 1.962910e+01  33.5340625
## 61  5.8552333  0.00000000 1.713099e+01  31.2475469
## 62  6.8628000  0.00000000 4.232950e-01  26.3443569
## 63  6.6010000  0.00000000 8.627253e+00  24.4962512
## 64  2.3919014  0.74212407 3.322556e+00   5.2550664
## 65  2.9398493  0.37192727 7.696759e+00   8.4440177
## 66  2.9068741  0.74113592 5.375424e+00   0.0000000
## 67  3.2943927  0.75125715 0.000000e+00  11.1862943
## 68  4.2134456  1.58579263 0.000000e+00  25.8007275
## 69  3.6589047  0.86148974 1.286249e+01   5.3416690
```

```
## 70  3.1832484  0.98766085 3.219859e+00   0.0000000
## 71  3.1922000  0.00000000 4.351579e+00   1.6849219
## 72  0.7370000  1.11164061 3.771368e+00  18.2424738
## 73  5.6328803  2.00095311 1.658386e+00  31.2866982
## 74  6.2948588  0.00000000 2.150274e+01  35.3579797
## 75  7.9350000  0.00000000 2.316263e+01  49.1319688
## 76  8.0836333  0.00000000 1.328206e+01  50.6265000
## 77  8.2780000  0.00000000 1.082815e+01  21.6644079
## 78  2.9759763  0.94012209 4.469657e+00   8.0860564
## 79  4.6910196  1.13684158 1.011992e+01  16.7676483
## 80  4.7823099  1.55840340 7.981965e+00  16.8355903
## 81  3.8088599  0.87276325 8.179947e+00  12.4583226
## 82  3.2161411  1.18945188 3.945207e+00   9.0891216
## 83  3.4878292  0.40654776 9.669589e+00  11.2449778
## 84  3.6066437 -0.50664375 1.555920e+01  12.7350955
## 85  0.4955774  0.00000000 0.000000e+00   0.1382500
## 86  2.4560844  0.00000000 0.000000e+00   1.8028929
## 87  4.3733747  0.00000000 0.000000e+00  29.1463017
## 88  6.1652654  0.00000000 3.475027e+01  20.3182490
## 89  5.9440000  0.00000000 1.952628e+01   8.8652235
## 90  4.1140333  0.00000000 4.399837e+01   3.6303999
## 91  1.7210000  0.00000000 5.998217e+00   6.5338860
## 92  3.3111891  0.73971276 6.991011e+00  10.0119406
## 93  3.3111891  0.73971276 6.991011e+00  10.0119406
## 94  3.3111891  0.73971276 6.991011e+00  10.0119406
## 95  3.0733151  0.88065452 5.209357e+00   8.6465134
## 96  3.0733151  0.88065452 5.209357e+00   8.6465134
## 97  4.6182792 -0.03465461 1.678045e+01  17.5147172
## 98  5.3907613 -0.49230918 2.256600e+01  21.9488191
```

```
#AG 2
library(readxl)
GI_Ext <- read.table("~/Desktop/GI_NA.txt")
as.matrix((GI_Ext))
```

```
##       V1           V2           V3           V4           V5
##  [1,] "KAL_0169"    "KAL_0707"    "KAL_1418"    "KAL_1800"    "KAL_0028"
##  [2,] NA           NA           NA           NA           NA
##  [3,] NA           NA           NA           "0"          "0"
##  [4,] "0"          "0"          NA           "0"          "0"
##  [5,] "0"          "0"          NA           "0"          "4.426622178"
##  [6,] "0"          "0"          NA           "0"          "1.638141422"
##  [7,] "0"          "0"          NA           "0"          "0"
##  [8,] "0"          "0"          NA           "0"          "0"
##  [9,] "0"          NA           NA           "0"          "0"
## [10,] "7.759635339" NA           NA           "0"          "0"
## [11,] "16.38145248" "0"          NA           "0"          "0"
```

```
##  [12,] "0"          "0"          NA           "0"          "0"
##  [13,] "0"          "0"          NA           NA           "0"
##  [14,] "0"          "0"          NA           NA           "0.442922036"
##  [15,] "0"          "0"          NA           NA           "11.51597322"
##  [16,] "0"          "0"          "6.080671687" NA          "0"
##  [17,] "0"          "0"          "6.080671687" "0"         "0"
##  [18,] "0"          "0"          "0.760083954" "0"         "0"
##  [19,] "0"          "0"          "0.157300376" "0"         "12.68042824"
##  [20,] "0.506338462" "2.5531"     "1.258403017" "0"         "9.948786825"
##  [21,] "0.327630769" "3.945700005" "4.047141927" "0"         "2.55087942"
##  [22,] "0"          "0"          "8.69504012"  "0"         "0.827281327"
##  [23,] "0"          "0"          "9.687754077" "0"         "0.95455538"
##  [24,] "4.655781214" "0"          "10.28338245" "0"         "0"
##  [25,] "10.79508687" "0"          "3.686858773" "0"         "0"
##  [26,] "6.446338445" "0"          "2.401435979" "0"         "0"
##  [27,] "17.07021538" "0"          "0"          NA           "0"
##  [28,] "6.256923075" "5.836725008" "11.07733175" NA          "0.304508899"
##  [29,] "0"          "9.069698273" "29.53955136" NA          "7.917231608"
##  [30,] "0"          "0"          "6.080671687" "0"         "0"
##  [31,] "0"          "0"          "6.080671687" "0"         "0"
##  [32,] "0"          "0"          "0.760083953" "0"         "0"
##  [33,] "0"          "0"          "0.157300376" "0"         "6.370635823"
##  [34,] "1.265846155" "3.186822215" "1.258403017" "1.053499999" "10.11375239"
##  [35,] "0.819076923" "4.925088891" "4.047141931" "3.862833341" "4.591582986"
##  [36,] "0"          "0"          "8.69504012"  "0"         "8.574896718"
##  [37,] "0"          "0"          "9.687754076" "0"         "13.98781485"
##  [38,] "0"          "0"          "10.28338245" "0"         "6.32663234"
##  [39,] "2.415538448" "9.123800023" "3.686858772" "0"         "0"
##  [40,] "16.11584608" "3.041266654" "2.401435979" "4.624615374" "0"
##  [41,] "42.67553854" "8.785000034" "0"          "9.018000006" "0"
##  [42,] "15.64230771" "8.826524997" "11.07733176" "3.377541605" "0"
##  [43,] "0"          "5.212272984" "29.53955136" "40.5304994"  "0"
##  [44,] "0"          "0"          "6.080671687" "0"         "13.56654278"
##  [45,] "0"          "0"          "6.080671687" "0.164999998" "2.777148843"
##  [46,] "0"          "0"          "0.760083953" "0.091666666" "0.263430117"
##  [47,] "0"          "0"          "0.157300376" "0"         "17.75692307"
##  [48,] "3.842000002" "1.768433327" "1.258403017" "1.053499999" "37.8640484"
##  [49,] "2.485999999" "2.733033341" "4.047141931" "3.862833341" "47.1706607"
##  [50,] "0"          "0"          "8.69504012"  "0"         "31.45031961"
##  [51,] "0"          "0"          "9.687754076" "0"         "12.26961517"
##  [52,] "0"          "0"          "10.28338245" "0"         "6.32663234"
##  [53,] "6.15138458"  "17.17807349" "3.686858772" "0"         "0"
##  [54,] "33.17184605" "15.16238932" "2.401435979" "4.383749992" "0"
##  [55,] "73.49092315" "24.76432538" "0"          "7.890750012" "0.226066914"
##  [56,] "49.3769231"  "16.36025298" "11.07733176" "3.124447631" "1.883890992"
##  [57,] "29.4"       "16.70838274" "29.53955136" "37.49337173" "0"
```

```
##  [58,] "0"          "0"          "8.636063609" "0"          "7.536968158"
##  [59,] "2.886853854" "0"          "8.636063616" "1.32"       "1.542860461"
##  [60,] "0.627576917" "0.684123704" "1.944547658" "0.733333333" "0.146350065"
##  [61,] "0"          "1.659"      "1.014725161" "0"          "10.76458533"
##  [62,] "1.536800001" "0"          "1.258403017" "0"          "27.43806588"
##  [63,] "1.520847058" "0"          "4.047141927" "0"          "49.84205153"
##  [64,] "0"          "0"          "8.69504012"  "0"          "43.31471879"
##  [65,] "0"          "0"          "8.100999614" "0"          "0"
##  [66,] "0"          "0"          "7.833681378" "0"          "0"
##  [67,] "2.460553833" "19.33025636" "2.780287151" "0"          "0"
##  [68,] "13.26873844" "29.09069445" "1.873656579" "0"          "0"
##  [69,] "32.16361405" "38.35038088" "0.580636739" "0"          "0.227740194"
##  [70,] "43.74447335" "19.89480554" "9.966387292" "1.352789812" "1.973728319"
##  [71,] "36.02658082" "29.76269586" "25.60930489" "16.23347773" "1.973226712"
##  [72,] "0"          NA           "16.30223946" "0"          "0"
##  [73,] "10.49016131" NA           "16.30223946" "0.910344827" "0"
##  [74,] "5.385685624" "8.690000003" "5.497938807" "0.458333332" "0"
##  [75,] "1.232791175" "6.082999997" "3.954467303" "0"          "1.210600961"
##  [76,] "0"          "0"          NA           "0"          "2.734126348"
##  [77,] "0"          "0"          NA           "0"          "3.304575218"
##  [78,] NA           NA           NA           "0"          NA
##  [79,] "0"          NA           "0.484578109" "0"          NA
##  [80,] "0"          NA           "0.484578109" "0"          NA
##  [81,] "0"          NA           "0.060572263" "0"          "0"
##  [82,] "0"          NA           "0.290318368" "0"          "0"
##  [83,] "4.612074724" NA           "2.322546973" "0"          "0.148578131"
##  [84,] "39.98950687" "0"          "6.633553885" "1.352789806" "1.348541406"
##  [85,] "40.44430135" "0"          "13.8185654"  "16.23347773" "2.870147933"
##  [86,] NA           NA           "16.30223946" NA           NA
##  [87,] "36.16793403" NA           "16.30223946" "0"          NA
##  [88,] "22.74458196" "0"          "5.497938807" "0"          "0"
##  [89,] "19.5485458"  "0"          "3.954467303" "0"          "0"
##  [90,] NA           "0"          NA           NA           "0.043550508"
##  [91,] NA           NA           NA           NA           "0.06660666"
##  [92,] NA           NA           NA           NA           NA
##  [93,] NA           NA           "0.484578109" NA           NA
##  [94,] NA           NA           "0.484578109" NA           NA
##  [95,] NA           NA           "0.060572263" "0"          NA
##  [96,] NA           NA           "0.290318369" "0"          NA
##  [97,] NA           NA           "2.322546973" NA           NA
##  [98,] NA           NA           "6.633553882" NA           NA
##  [99,] NA           NA           "13.8185654"  NA           NA
##       V6           V7           V8           V9           V10
##  [1,] "KAL_0168"    "KAL_0401"    "KAL_0722"    "KAL_0122"    "KAL_0674"
##  [2,] NA           NA           NA           NA           NA
##  [3,] NA           "0"          "0"          NA           NA
```

```
##  [4,] "0"          "0"          "0"          "2.74E-08"    "0"
##  [5,] "0"          "33.48952594" "0"         "10.57444445" "0"
##  [6,] "1.908533291" "1.240352837" "0"         "8.014111109" "0"
##  [7,] "1.234933302" "0"          "0"         "3.642666667" "0"
##  [8,] "0"          "0.784"      "0"          "2.219999988" "0"
##  [9,] "6.835133022" "24.65458827" "0"         "23.20533334" NA
## [10,] "11.7067877"  "51.07894112" "0"         "13.18044441" NA
## [11,] "2.413845031" "21.78209473" "0"         "0"          NA
## [12,] "0"          "60.99053965" "0"         "0"          NA
## [13,] "0"          "59.93058906" "0"         "0"          NA
## [14,] "0"          "40.82007213" NA          "0"          NA
## [15,] "0"          "26.37425377" NA          "0"          NA
## [16,] "0"          NA           NA          NA           NA
## [17,] "0"          "24.86130374" "0"         "0"          "0"
## [18,] "0"          "25.73755731" "0"         "0"          "0"
## [19,] "0"          "11.47764705" "0"         "0"          "0"
## [20,] "1.145119974" "6.606352942" "0"         "5.219555551" "1.135749999"
## [21,] "0.74095998"  "4.610823536" "0"         "7.500666667" "0.309749998"
## [22,] "0"          "0.958"      "0"          "2.478999987" "2.779000007"
## [23,] "4.101079807" "0.619882351" "0"         "22.26933334" "0.883499995"
## [24,] "7.450534161" "1.043764727" "0"         "22.61066665" "9.146999981"
## [25,] "2.348614711" "44.60117647" "0"         "3.187555536" "12.7055"
## [26,] "0"          "72.29917647" "0"         "2.031999998" "10.3205"
## [27,] "0"          "62.50870588" "0"         "13.55822222" "14.796"
## [28,] "0"          "64.39870588" NA          "6.358666656" "2.785499984"
## [29,] "0"          "64.44"      NA          "8.137500007" "6.370619198"
## [30,] "0"          "84.74978388" NA          "0"          "0"
## [31,] "0"          "60.79208904" "0"         "0"          "0"
## [32,] "0"          "29.76235293" "0"         "0"          "0.9275"
## [33,] "0"          "2.785764708" "0"         "0"          "2.75"
## [34,] "0"          "6.487882412" "0"         "0"          "1.012500002"
## [35,] "0"          "29.3696471"  "0"         "0"          "1.722500002"
## [36,] "0"          "3.338"      "0"         "0"          "0.783499999"
## [37,] "0"          "2.159882346" "0"         "0"          "0.470812497"
## [38,] "1.066153845" "0"          "8.116500013" "0"         "4.386312492"
## [39,] "2.250769236" "3.151647059" "4.509166659" "0"         "5.613999999"
## [40,] "0"          "37.28870588" "0"         "0.510222222" "0.826"
## [41,] "0"          "74.126"     "3.24077419"  "1.275555556" "0"
## [42,] "0"          "76.26485314" "13.156"     "0"          "0"
## [43,] "0"          "76.65684027" "0"         "0"          "0"
## [44,] "0"          "86.62455367" "0"         "0"          "0"
## [45,] "0"          "47.55211766" "0.94559999"  "0"         "0"
## [46,] "0"          "18.15929412" "1.420749988" "5.584444441" "2.194500001"
## [47,] "0"          "8.299529407" "1.010333326" "0"         "9.700499998"
## [48,] "0"          "26.47470135" "0"         "0"          "41.79274998"
## [49,] "0"          "42.25676911" "0"         "0"          "10.60199994"
```

```
##  [50,] "0"          NA           "0"          "0"          "2.551500006"
##  [51,] "0"          NA           "0"          "0"          "0.893249995"
##  [52,] "0"          "25.14721562" "9.701249997" "0"          "3.275250006"
##  [53,] "0"          "16.2163305"  "7.178666636" "0"          "2.726500001"
##  [54,] "3.069230739" "30.59282447" "1.78908332"  "0"          "3.637750001"
##  [55,] "15.39923078" "35.10740116" "3.539749998" "0"          "0.506250001"
##  [56,] "10.67307693" "31.83284725" "12.97908334" "0"          "3.357250004"
##  [57,] "0"          "31.69270527" "0"          "9.712500008" "0.275466663"
##  [58,] "0"          "13.75387278" "0"          "0"          "0.351374998"
##  [59,] "0"          "37.19026446" "5.910000002" "0"          "1.288375001"
##  [60,] "3.632676931" "33.18830413" "11.366"     "0"          "1.00625"
##  [61,] "3.987230771" "54.03442905" "8.082666672" "0"          "1.00625"
##  [62,] "9.132632189" "21.62911378" "0"          "0"          "3.907749997"
##  [63,] "7.736373693" "87.88997276" "0"          "0"          "10.83200001"
##  [64,] "0"          NA           "0"          "0"          "13.16908898"
##  [65,] "0"          NA           "0"          NA           "0"
##  [66,] "0"          NA           "12.678"     NA           "0.805878723"
##  [67,] "2.349484124" NA           "21.356"     "0"          "3.331831708"
##  [68,] "11.10351577" NA           "14.31266668" "0"          "2.189500001"
##  [69,] "23.20877135" NA           "3.201999997" "0"          "4.425499997"
##  [70,] "37.96233188" NA           "11.74066667" "13.14933334" "4.676749998"
##  [71,] "0"          NA           "0"          "2.465499985" "27.67827695"
##  [72,] NA           NA           "0"          NA           "0"
##  [73,] "0"          NA           "3.69375"    "0"          "0"
##  [74,] "13.44211478" NA           "7.103750002" "0"          "0"
##  [75,] "9.142229283" "49.40718872" "5.987160502" "0"          "4.520249998"
##  [76,] "12.71308876" "65.17676664" "0"          "0"          "8.51175"
##  [77,] "15.94161854" "14.55257608" "0"          "0"          "1.712573247"
##  [78,] NA           NA           "0"          "0"          "0.093628623"
##  [79,] NA           NA           "0"          NA           NA
##  [80,] "0"          NA           "7.923750001" NA           NA
##  [81,] "3.915806859" NA           "15.208625"  "0"          NA
##  [82,] "16.45970577" NA           "11.41029168" "0"          NA
##  [83,] "28.41513171" NA           "3.087999998" "0"          NA
##  [84,] "56.15516848" NA           "7.337916665" "0"          NA
##  [85,] "0"          NA           "0"          "0"          NA
##  [86,] NA           NA           "0"          NA           NA
##  [87,] NA           NA           "0"          NA           NA
##  [88,] "50.02858776" NA           "0"          "0"          NA
##  [89,] "19.12960793" "0"          "0"          "0"          "0.862019752"
##  [90,] "2.681229225" "7.760724196" NA           "0"          "4.12763117"
##  [91,] "0"          NA           "0"          "0"          "1.052249994"
##  [92,] NA           NA           "0"          NA           "0"
##  [93,] NA           NA           "0"          NA           NA
##  [94,] NA           NA           "0"          NA           NA
##  [95,] NA           NA           "4.963000003" NA           NA
```

```
## [96,] NA           NA           "6.573000002" NA           NA
## [97,] NA           NA           "2.898000001" NA           NA
## [98,] NA           NA           "0"           NA           NA
## [99,] NA           NA           "0"           NA           NA
##       V11          V12          V13          V14          V15
##  [1,] "KAL_0877"    "KAL_1459"    "KAL_0109"    "KAL_0869"    "KAL_1310"
##  [2,] NA           NA           NA           NA           NA
##  [3,] "0"          NA           "0"          NA           NA
##  [4,] "0"          "0"          "0"          "2.428"      NA
##  [5,] "0"          "0"          "0"          "2.428"      NA
##  [6,] "0"          "0"          "0"          NA           NA
##  [7,] "0"          "0"          "0"          NA           "0"
##  [8,] "0"          "0"          "0"          NA           "0"
##  [9,] "0"          "0"          "3.229380003" NA           "0"
## [10,] "0"          "0"          "3.387850013" NA           "0"
## [11,] "0"          "0"          "9.119469998" "0.478"      "0.40122115"
## [12,] "0"          "0"          "1.253559995" "0.478"      "1.203663453"
## [13,] "0"          "0"          "0"           NA           "0"
## [14,] "0"          "0"          "4.814451601" NA           "0"
## [15,] "0"          "0"          "2.394352945" NA           "0"
## [16,] "0"          "0"          NA           "1.096"      NA
## [17,] "0"          "0.433749999" "0.666156535" "5.507576936" NA
## [18,] "2.756988716" "0.78075"    "1.490579995" "1.657653835" NA
## [19,] "3.515096255" "0"          "1.754090001" "0.604"      NA
## [20,] "0.421170856" "0"          "0.625210013" NA           NA
## [21,] "0"          "0"          "0"          NA           "7.210000006"
## [22,] "0"          "0"          "0"          NA           "4.029999986"
## [23,] "0"          "0"          "0"          "0.48"       "1.456586536"
## [24,] "0"          "0"          "0.336150007" "0.595269231" "1.493221149"
## [25,] "0"          "0"          "4.369949997" "0.813"      "1.471144224"
## [26,] "9.96591067"  "0"          "0"          NA           "4.413432705"
## [27,] "5.53661705"  "0.47575"    "0"          "2.998"      "0"
## [28,] "0"          "1.257500001" "1.865599997" "2.936038463" "0"
## [29,] "8.225751831" "0.092533332" "0.598588237" "1.387"      "0"
## [30,] "3.43862455"  "3.012"      "0"          "0.706"      "0"
## [31,] "5.076047842" "2.070750002" "17.22008354" "1.877230773" "0"
## [32,] "8.031829572" "0"          "1.322509995" "2.03"       "0"
## [33,] "3.751491509" "0"          "5.363300004" NA           "0"
## [34,] "0.334027876" "0"          "8.175050004" "0.39"       NA
## [35,] "0.791781146" "0"          "0"          "0.435576923" "23.072"
## [36,] "0.317163026" "0.162749997" "0"          "1.286"      "13.22416"
## [37,] "6.509371437" "2.115750003" "0"          "3.020230769" "8.587067303"
## [38,] "12.00504484" "0"          "0"          "3.014115386" "6.419134622"
## [39,] "7.778758441" "1.632750001" "0"          "0.64"       "0"
## [40,] "19.46528188" "1.632750001" "0"          NA           "0"
## [41,] "22.39512424" "0"          "0"          NA           "0"
```

```
##  [42,] "66.62719709" "0"          "0"          NA           "0"
##  [43,] "26.10667207" "32.27456697" "0"          NA           "0"
##  [44,] "0.69345952"  "0"          "0"          NA           "0"
##  [45,] "2.810961248" "0"          "9.442299967" NA           "0"
##  [46,] "3.037515748" "0"          "1.322509995" NA           "0"
##  [47,] "2.554618364" "0"          "1.937669994" "0.079"      "0"
##  [48,] "0"           "0"          "0.047320001" "2.062115385" NA
##  [49,] "3.986660183" "0"          "0"          "3.112"      NA
##  [50,] "0.306666162" "0"          "0"          NA           "14.068"
##  [51,] "0"           "0"          "0"          NA           "13.63476923"
##  [52,] "0"           "0"          "0"          NA           "8.528769241"
##  [53,] "5.568323121" "0"          "0"          NA           "0"
##  [54,] "11.22916152" "0"          "0"          NA           "0"
##  [55,] "3.144910227" "0"          "5.878294114" NA           "0"
##  [56,] "13.82916668" "0"          "25.66952249" NA           "0"
##  [57,] "13.52150214" "0"          "10.41790085" NA           "0"
##  [58,] "0"           "0"          "0"          NA           "0.682186131"
##  [59,] "0"           "0"          "6.299439983" NA           "8.245261906"
##  [60,] "0"           "0"          "0"          "0.852"      "7.772451946"
##  [61,] "0"           "0"          "1.577549999" "4.638846162" "5.721096572"
##  [62,] "0"           "0"          "1.438469999" "2.789615383" "1.664307686"
##  [63,] "49.53914362" NA           "0"          "1.072"      "0"
##  [64,] "60.97125361" NA           "0"          NA           "0"
##  [65,] NA            NA           NA           NA           NA
##  [66,] NA            NA           "0"          NA           NA
##  [67,] NA            NA           "0"          NA           NA
##  [68,] NA            NA           "0"          NA           NA
##  [69,] NA            NA           "9.043529398" NA           NA
##  [70,] NA            NA           "39.49157296" NA           NA
##  [71,] NA            NA           "42.20585458" NA           NA
##  [72,] "8.69646797"  NA           "0"          NA           "0.992270733"
##  [73,] "0"           "0"          "0"          NA           "2.481332706"
##  [74,] "0"           "0"          "0"          NA           "2.119759595"
##  [75,] "0"           "0"          "2.83959001"  NA          "1.20398076"
##  [76,] "0"           "0"          "0.218430005" NA          "0.312057688"
##  [77,] "0"           NA           "0"          NA           "0"
##  [78,] NA            NA           "0"          NA           "0"
##  [79,] NA            NA           NA           NA           NA
##  [80,] NA            NA           NA           NA           NA
##  [81,] NA            NA           NA           NA           NA
##  [82,] NA            NA           NA           NA           NA
##  [83,] NA            NA           NA           NA           NA
##  [84,] NA            NA           NA           NA           NA
##  [85,] NA            NA           NA           NA           NA
##  [86,] NA            NA           NA           NA           NA
##  [87,] NA            NA           NA           NA           "0"
```

```
##  [88,] "0"           "0"           "0"           NA            "0"
##  [89,] "0"           "0"           "0"           "0.22"        "0"
##  [90,] "2.849810254" "0"           "0"           "0.22"        "0"
##  [91,] "4.145178554" NA            "0"           NA            "0"
##  [92,] NA            NA            NA            NA            "0"
##  [93,] NA            NA            NA            NA            NA
##  [94,] NA            NA            NA            NA            NA
##  [95,] NA            NA            NA            NA            NA
##  [96,] NA            NA            NA            NA            NA
##  [97,] NA            NA            NA            NA            NA
##  [98,] NA            NA            NA            NA            NA
##  [99,] NA            NA            NA            NA            NA
##        V16
##   [1,] "KAL_1412"
##   [2,] "0"
##   [3,] "0"
##   [4,] "0"
##   [5,] "0"
##   [6,] "0"
##   [7,] "0"
##   [8,] "0"
##   [9,] "0"
##  [10,] "0"
##  [11,] "0"
##  [12,] "0"
##  [13,] "0"
##  [14,] "0"
##  [15,] "0"
##  [16,] "2.869166664"
##  [17,] "0.981557013"
##  [18,] "0"
##  [19,] "0"
##  [20,] "5.554421048"
##  [21,] "5.965859644"
##  [22,] "0"
##  [23,] "0"
##  [24,] "0"
##  [25,] "0"
##  [26,] "0"
##  [27,] "0"
##  [28,] "0"
##  [29,] "0"
##  [30,] "2.973627631"
##  [31,] "3.169289472"
##  [32,] "0.203644739"
##  [33,] "0"
```

```
##  [34,] "1.514842096"
##  [35,] "3.722078937"
##  [36,] "4.422833333"
##  [37,] "0"
##  [38,] "0"
##  [39,] "0"
##  [40,] "0"
##  [41,] "0"
##  [42,] "0"
##  [43,] "0"
##  [44,] "0"
##  [45,] "1.697039478"
##  [46,] "0.203644739"
##  [47,] "0"
##  [48,] "0"
##  [49,] "0"
##  [50,] "0"
##  [51,] "0"
##  [52,] "0"
##  [53,] "0"
##  [54,] "0"
##  [55,] "0"
##  [56,] "0"
##  [57,] "0"
##  [58,] "0"
##  [59,] "0"
##  [60,] "0"
##  [61,] "0"
##  [62,] "0"
##  [63,] NA
##  [64,] NA
##  [65,] NA
##  [66,] NA
##  [67,] NA
##  [68,] NA
##  [69,] NA
##  [70,] NA
##  [71,] NA
##  [72,] "0.905171053"
##  [73,] "0"
##  [74,] "0"
##  [75,] "0"
##  [76,] "0"
##  [77,] NA
##  [78,] NA
##  [79,] NA
```

```
##  [80,] NA
##  [81,] NA
##  [82,] NA
##  [83,] NA
##  [84,] NA
##  [85,] NA
##  [86,] "2.413789475"
##  [87,] "0"
##  [88,] "0"
##  [89,] "0"
##  [90,] "0"
##  [91,] NA
##  [92,] NA
##  [93,] NA
##  [94,] NA
##  [95,] NA
##  [96,] NA
##  [97,] NA
##  [98,] NA
##  [99,] NA
```

```
colnames(GI_Ext)<-GI_Ext[1,]



data_numGI<-as.data.frame(apply(GI_Ext,2, as.numeric))
```

```
## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion
```

```
## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(GI_Ext, 2, as.numeric): NAs introduced by coercion
```

```
sapply(data_numGI, class
)
```

```
## KAL_0169 KAL_0707 KAL_1418 KAL_1800 KAL_0028 KAL_0168 KAL_0401 KAL
_0722
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
## KAL_0122 KAL_0674 KAL_0877 KAL_1459 KAL_0109 KAL_0869 KAL_1310 KAL
_1412
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
```

```
View(data_numGI
)
```

```
Estim.NA <- estim_ncpPCA(data_numGI, ncp.min = 0, ncp.max = 10, method =
                      "Regularized", scale = TRUE, method.cv = "gcv",
                    nbsim = 100, pNA = 0.05, threshold=1e-4, verbose =
                      TRUE)
Estim.NA
```

```
## $ncp
## [1] 1
##
## $criterion
##        0        1        2        3        4        5        6
##  99.33192  92.40852  95.77007 103.95122 121.87954 150.89299 181.08828
##        7        8        9       10
## 221.28094 348.71781 564.52860 1440.43102
```

```
Impute.PCA <- imputePCA(data_numGI, ncp = 2) #based on the results of Estim.NA

GI.complete.sample_HS <- Impute.PCA$completeObs
GI.complete.sample_HS <- as.data.frame(GI.complete.sample_HS)

write.table(GI.complete.sample_HS, "~/Desktop/GI_NA_Test.txt")
("~/Desktop/GI_NA_Test.txt")
```

```
## [1] "~/Desktop/GI_NA_Test.txt"

#AG 3
library(readxl)
WE_Ext <- read.table("~/Desktop/WE_NA.txt")
as.matrix((WE_Ext))

##       V1              V2              V3              V4
##  [1,] "1892-93-286-180" "1892-93-307-197" "1894-95-142-277" "1898-99-232-476"
##  [2,] NA              NA              NA              NA
##  [3,] NA              NA              "0"             "0"
##  [4,] NA              "0"             "0"             "0"
##  [5,] "0"             "0"             "0"             "0"
##  [6,] "0"             "0"             "0"             "0"
##  [7,] "0"             "0"             "0"             "0"
##  [8,] "0"             "0"             "0"             "0"
##  [9,] "0"             "0"             "0"             "0"
## [10,] "0"             "0"             "0"             "0"
## [11,] "0"             "0"             "0"             "4.458108382"
## [12,] "0"             "0"             "0"             "15.58317653"
## [13,] "0"             "0"             "0"             "23.61415341"
## [14,] "0"             "20.59779424"   "0"             "15.46149862"
## [15,] "0"             "0"             "0"             "27.62063228"
## [16,] NA              NA              NA              "0"
## [17,] NA              NA              "0"             "0"
## [18,] "0"             "0"             "0"             "0"
## [19,] "0"             "0"             "0"             "0"
## [20,] "0"             "0"             "0"             "0"
## [21,] "0"             "0"             "0"             "0"
## [22,] "0"             "0"             "0"             "0"
## [23,] "2.193230767"   "0"             "0"             "0"
## [24,] "3.389538467"   "0"             "0"             "7.430399998"
## [25,] "1.368230765"   "0"             "0"             "9.737265027"
## [26,] "4.10469232"    "0"             "0"             "13.45497259"
## [27,] "0"             "0"             "0"             "17.33809205"
## [28,] "1.309340289"   "20.59779424"   "0"             "13.60436793"
## [29,] "34.04284845"   "0"             "0"             "21.75640443"
## [30,] "0"             NA              "0"             "0"
## [31,] "0"             "2.035005443"   "16.97492495"   "0"
## [32,] "2.985073179"   "3.745000005"   "10.70388015"   "0"
## [33,] "0.915294866"   "0"             "1.568069876"   "0"
## [34,] "0"             "0"             "0"             "0"
## [35,] "0"             "0"             "0"             "0"
## [36,] "5.205666678"   "1.400000002"   "0"             "0"
## [37,] "7.724512827"   "0"             "0"             "0"
## [38,] "2.655051282"   "4.168500006"   "0"             "18.57600001"
## [39,] "2.939371778"   "5.309500007"   "5.078957006"   "17.65599999"
```

```
##  [40,] "9.69535899"     "1.725500002"    "14.60730978"    "10.26266667"
##  [41,] "4.486282061"    "0"              "17.15103504"    "7.923999999"
##  [42,] "5.825302133"    "34.9602243"     "0.173744286"    "10.81867189"
##  [43,] "53.74952342"    "8.39593872"     "2.084931445"    "12.96006265"
##  [44,] "7.456039157"    "21.16758797"    "0"              "6.46"
##  [45,] "5.217099957"    "2.001129763"    "18.79366689"    "1.614999999"
##  [46,] "3.693012387"    "3.895500004"    "16.91225692"    "0"
##  [47,] "2.77510256"     "0"              "8.80839038"     "0"
##  [48,] "1.971794867"    "0.558999995"    "3.443022792"    "0"
##  [49,] "2.402948717"    "0.159420288"    "0"              "0"
##  [50,] "6.676500007"    "1.400000002"    "0"              "0"
##  [51,] "6.506051293"    "0"              "0"              "0"
##  [52,] "0.771974362"    "4.168500006"    "0"              "11.391"
##  [53,] "2.179243576"    "5.309500007"    "5.804522285"    "23.83533334"
##  [54,] "8.785522138"    "2.082499999"    "16.69406829"    "31.51533333"
##  [55,] "11.22312821"    "0.191499998"    "18.92528003"    "34.574"
##  [56,] "8.995966557"    "34.10304532"    "0.175573174"    "36.44278462"
##  [57,] "42.516089"      "8.39593872"     "2.38277879"     "25.51741545"
##  [58,] "17.894494"      "21.16758797"    NA               "6.46"
##  [59,] "12.52103992"    "1.764"          NA               "0.76"
##  [60,] "1.790932499"    "1.204"          "73.06327"       "0"
##  [61,] "4.009000011"    "0"              "57.92256451"    "0"
##  [62,] "7.056538461"    "4.472"          "27.54418261"    "0"
##  [63,] "6.392038458"    "1.275362319"    "0"              "0"
##  [64,] "3.025714287"    "0"              "0"              "0"
##  [65,] "0"              "0"              "0"              "0"
##  [66,] "0"              "0"              NA               "11.39099999"
##  [67,] "0"              "0"              NA               "23.83533333"
##  [68,] "3.289314647"    "2.856"          NA               "31.51533333"
##  [69,] "16.1684308"     "1.532"          "0"              "34.574"
##  [70,] "9.355381678"    "13.74036231"    "0"              "36.44278462"
##  [71,] "18.4302219"     "0"              NA               "25.51741545"
##  [72,] NA               "21.16758797"    NA               NA
##  [73,] NA               "1.764"          NA               "0"
##  [74,] "0"              "0.7525"         "45.66454375"    "0"
##  [75,] "3.214653868"    "7.43366286"     "44.5313724"     "0"
##  [76,] "13.23420512"    "4.472"          "35.54063382"    "0"
##  [77,] "7.960740727"    "1.275362319"    "0"              NA
##  [78,] "0"              "0"              "0"              NA
##  [79,] NA               "0"              "0"              NA
##  [80,] NA               "0"              NA               NA
##  [81,] NA               "0"              NA               NA
##  [82,] NA               "2.856"          NA               NA
##  [83,] NA               "1.532"          "0"              NA
##  [84,] NA               "13.74036231"    "0"              NA
##  [85,] NA               "0"              NA               NA
```

```
##  [86,] NA              NA              NA              NA
##  [87,] NA              NA              NA              "0"
##  [88,] "0"             "0"             "0"             "0"
##  [89,] "9.476615404"   "19.82310094"   "22.21271885"   "0"
##  [90,] "4.488923067"   NA              "53.3105253"    NA
##  [91,] "0"             NA              NA              NA
##  [92,] NA              NA              NA              NA
##  [93,] NA              NA              NA              NA
##  [94,] NA              NA              NA              NA
##  [95,] NA              NA              NA              NA
##  [96,] NA              NA              NA              NA
##  [97,] NA              NA              NA              NA
##  [98,] NA              NA              NA              NA
##  [99,] NA              NA              NA              NA
##        V5          V6          V7             V8
##   [1,] "1879-73-121" "1893-94-8"   "1893-94-86-210" "1899-288-512"
##   [2,] NA           NA           NA             NA
##   [3,] NA           NA           "0"            NA
##   [4,] "1.43E-08"   "0"          "0"            "0.581410122"
##   [5,] "1.603713397" "0"          "1.890000001"  "0.657897377"
##   [6,] "15.07892398" "0"          "0.070000001"  "1.104480515"
##   [7,] "20.3619384"  "0"          "0"            "0"
##   [8,] "2.677211744" "0"          "0"            "0"
##   [9,] "0.612166127" "0"          "0.492470589"  "0"
##  [10,] "3.537478042" "0"          "5.942352933"  "0"
##  [11,] "1.088502609" "2.392"      "5.641411765"  "0"
##  [12,] "0"           "10.076"     "0"            "0"
##  [13,] "0"           "25.72"      "0"            "3.38877572"
##  [14,] "0"           "25.416"     "0"            "2.541581789"
##  [15,] "0"           "0"          "0"            "0"
##  [16,] NA            NA           NA             NA
##  [17,] "0"           "0"          "0"            "1.94146778"
##  [18,] "4.688852479" "0"          "0"            "4.403517952"
##  [19,] "16.25163887" "9.704999999" "1.527882354"  "2.751738041"
##  [20,] "3.785902776" "0.855"      "0.056588236"  "4.144537867"
##  [21,] "0.178749999" "0"          "0.213529411"  "0"
##  [22,] "3.490666678" "0"          "0.486"        "0.318560042"
##  [23,] "0"           "0"          "0.100941176"  "0"
##  [24,] "0.074305556" "7.874999999" "7.597529403"  "0"
##  [25,] "0.029722222" "0.598"      "10.53"        "0"
##  [26,] "0.014722222" "2.519000001" "0"            "0"
##  [27,] "0.036805555" "13.795"     "1.013147059"  "13.81296898"
##  [28,] "0"           "27.684"     "6.215205892"  "12.14089718"
##  [29,] "0"           "4.853999999" "0.754838716"  "0.333969462"
##  [30,] "0"           "0"          NA             "3.354479882"
##  [31,] "1.136851853" "1.803666668" "0"            "2.758102109"
```

```
##  [32,] "7.491555605" "0"           "0.968470588"    "5.660994601"
##  [33,] "25.49758336" "22.21966668" "2.019058824"    "3.614750042"
##  [34,] "5.820569444" "8.133000006" "0.062823531"    "0.572629613"
##  [35,] "0.415375"    "0"           "0.207058823"    "0.064771634"
##  [36,] "0"           "0"           "1.042"          "0.961181903"
##  [37,] "0"           "0"           "0.467176469"    "0"
##  [38,] "0.166034483" "4.374999994" "1.964588236"    "1.435037274"
##  [39,] "0.0535"      "0"           "6.082941176"    "4.966510162"
##  [40,] "0.191777777" "0"           "1.168470588"    "0.862731275"
##  [41,] "8.836666672" "13.705"      "11.4562353"     "4.888909224"
##  [42,] "8.920916677" "19.90233333" "8.924764686"    "6.677035163"
##  [43,] "0.497437495" "8.987333336" "3.694903228"    "0.802150497"
##  [44,] "4.292725383" "0"           "0"              "13.53962048"
##  [45,] "3.124638888" "1.803666668" "0"              "4.399816098"
##  [46,] "0.719055567" "1.304999998" "0"              "7.46490157"
##  [47,] "9.940098311" "20.42583334" "6.896117652"    "4.411678392"
##  [48,] "14.08964256" "12.26633333" "0.25541177"     "0.402032903"
##  [49,] "0.214"       "0"           "0"              "0.064771634"
##  [50,] "0"           "0"           "0"              "0.775547651"
##  [51,] NA            "0"           "0"              "0"
##  [52,] "0"           "0"           "0"              "3.280085204"
##  [53,] "0"           "0"           "3.264470588"    "8.137811273"
##  [54,] "0.29382716"  "0"           "9.793411766"    "6.942458015"
##  [55,] "11.4196627"  "9.613333342" "8.858470594"    "13.33599408"
##  [56,] "23.01519891" "8.052333341" "18.92258824"    "5.478857501"
##  [57,] "5.449974502" "6.290666672" "1.938193564"    "0.676673566"
##  [58,] "2.384847419" NA            "3.291176478"    "15.37064077"
##  [59,] "13.90984527" "0"           "0"              "4.511103329"
##  [60,] "6.253472823" "3.132"       "0"              "3.074833253"
##  [61,] "12.3535639"  "8.5341"      "0"              "4.657495207"
##  [62,] "6.858787428" "8.672709656" "0"              "5.047161792"
##  [63,] NA            "0"           "0"              "6.98837787"
##  [64,] NA            NA            NA               "0"
##  [65,] NA            NA            NA               NA
##  [66,] NA            NA            NA               NA
##  [67,] NA            "0"           NA               "4.923599322"
##  [68,] "0"           "0"           NA               "10.80840309"
##  [69,] "4.70977072"  "0"           NA               "19.45690502"
##  [70,] "25.05650174" "0"           "0"              "6.899550899"
##  [71,] "32.28626058" "0"           "0"              "0.845632184"
##  [72,] "0"           NA            "51.82717974"    "15.71643885"
##  [73,] "18.39654256" NA            "10.2422353"     "5.193494389"
##  [74,] "24.68989029" NA            "27.86376471"    "13.33344478"
##  [75,] "23.15041668" "7.545999983" "14.62294116"    "8.691009346"
##  [76,] "0"           "1.384935403" "0.259310351"    "13.48326886"
##  [77,] NA            NA            NA               "16.51798398"
```

```
##  [78,] NA           NA           NA             NA
##  [79,] NA           NA           NA             NA
##  [80,] NA           NA           NA             NA
##  [81,] NA           NA           NA             NA
##  [82,] NA           NA           NA             NA
##  [83,] NA           NA           NA             NA
##  [84,] NA           NA           NA             NA
##  [85,] NA           NA           NA             NA
##  [86,] NA           NA           NA             "15.88115852"
##  [87,] NA           NA           NA             "5.425645729"
##  [88,] "50.40326749" NA          "21.30390884"  "16.41039466"
##  [89,] "1.738573721" "0"         "47.5525009"   "54.85091689"
##  [90,] NA           "0"          "23.86690195"  "44.15755933"
##  [91,] NA           NA           NA             NA
##  [92,] NA           NA           NA             NA
##  [93,] NA           NA           NA             NA
##  [94,] NA           NA           NA             NA
##  [95,] NA           NA           NA             NA
##  [96,] NA           NA           NA             NA
##  [97,] NA           NA           NA             NA
##  [98,] NA           NA           NA             NA
##  [99,] NA           NA           NA             NA
##        V9           V10          V11            V12
##   [1,] "1886-87-99"  "1890-91-21-129" "1892-93-285-179" "1906-07-37"
##   [2,] NA           NA             NA             NA
##   [3,] NA           NA             "0"            "0"
##   [4,] "7.91E-09"    "0"            "0.380719422"  "4.014266635"
##   [5,] "4.512399991" "9.537844947"  "0.206718748"  "17.7464"
##   [6,] "19.53301657" "27.08418119"  "0"            "14.11333334"
##   [7,] "16.06786242" "5.941466816"  "0"            "3.202800001"
##   [8,] "2.380599994" "3.221125301"  "0.015703125"  "0"
##   [9,] "0"           "41.87462969"  "0.204140624"  "0"
##  [10,] "0"           "9.758281084"  "0.280312497"  "0"
##  [11,] "3.897324665" "21.40958262"  "0.792656244"  "0"
##  [12,] "3.626133365" "3.844676602"  "4.062733791"  "0"
##  [13,] "0"           "0"            "34.26600001"  "0"
##  [14,] "0"           "0"            "7.795499956"  "0"
##  [15,] "0"           "0"            "0"            NA
##  [16,] NA           NA             "0"            "0"
##  [17,] "0"           "7.91251574"   "0"            "0.234666663"
##  [18,] "17.9758353"  "4.450790102"  "1.348593748"  "1.723466664"
##  [19,] "13.07677781" "1.788199902"  "1.348593749"  "1.932000004"
##  [20,] "17.04622221" "2.580927933"  "0"            "0"
##  [21,] "13.43906665" "0.432950054"  "0"            "0"
##  [22,] "1.999844437" "1.090077781"  "0.057578124"  "0"
##  [23,] "0"           "14.17101142"  "0.748515628"  "0"
```

```
##  [24,] "0"          "0"            "1.027812497"     "0"
##  [25,] "3.80E-09"    "0.937487198"    "2.906406252"     "3.570000007"
##  [26,] "5.244399979" "0.937487198"    "9.454609383"     "8.387466671"
##  [27,] "11.75688888" "0.364972829"    "31.40498438"     "1.564666644"
##  [28,] "2.099999996" "24.35777378"    "15.74049991"     "0"
##  [29,] "0"          "1.990419312"    "3.171285362"     "0"
##  [30,] NA           "0"              "0"               "0"
##  [31,] "0"          "0"              "0"               "0.330666662"
##  [32,] "14.99477786" "1.174947861"    "1.538578126"     "1.52106667"
##  [33,] "35.00026671" "1.533004755"    "2.822203127"     "0"
##  [34,] "7.609666668" "0.198920497"    "0.713125001"     "0"
##  [35,] "0"          "0"              "0"               "0"
##  [36,] "0"          "0"              "0"               "0"
##  [37,] "0"          "0"              "0"               "0"
##  [38,] "0"          "0"              "0"               "0"
##  [39,] "0"          "0"              "0"               "0"
##  [40,] "0"          "6.349147198"    "4.958268182"     "28.33600004"
##  [41,] "0"          "47.59085963"    "30.80734341"     "23.85666663"
##  [42,] "0"          "46.64195324"    "31.83339061"     "2.12359998"
##  [43,] "0"          "2.840994414"    "3.608361039"     "0"
##  [44,] "0"          "0"              "0"               "7.612781963"
##  [45,] "0"          "0"              "0"               "0"
##  [46,] "6.268888907" "0"              "0.475453125"     "0"
##  [47,] "7.41859999"  "0"              "3.403124999"     "0"
##  [48,] "1.976333329" "11.71236281"    "1.626484378"     "0"
##  [49,] "0"          "3.19428075"     "0"               "0"
##  [50,] "0"          "0"              "0"               NA
##  [51,] "0"          NA               "0"               NA
##  [52,] "0"          NA               "0"               "0"
##  [53,] "0"          NA               "0"               "19.89314629"
##  [54,] "0"          NA               "5.712362992"     "47.26633121"
##  [55,] "0"          NA               "20.24043562"     "37.63401249"
##  [56,] "0"          NA               "13.14742186"     "3.456030625"
##  [57,] "0"          NA               "3.290277856"     "0"
##  [58,] "8.878011089" "4.015721886"    "0"               "2.055200001"
##  [59,] "4.713546769" "0"              "0"               "8.092694767"
##  [60,] "9.254311103" "12.95325842"    "7.460578135"     "31.30772978"
##  [61,] "2.408041428" "52.61619827"    "9.247731554"     "4.340730401"
##  [62,] "5.262057567" "77.02713693"    "1.882762451"     "14.4691014"
##  [63,] "8.505629108" "79.9886114"     "0"               NA
##  [64,] "0"          NA               "0"               NA
##  [65,] NA           NA               "0"               NA
##  [66,] NA           NA               "0"               NA
##  [67,] NA           NA               NA                NA
##  [68,] NA           NA               NA                NA
##  [69,] NA           NA               "0"               NA
```

```
##  [70,] NA           NA             "0"             NA
##  [71,] NA           NA             "0"             NA
##  [72,] "31.53950961" "0"            "0"             "2.206800002"
##  [73,] "6.773120427" "0"            "0.587031249"    "0.797999988"
##  [74,] "1.02E-08"    "27.58469089"   "12.61934375"    "7.181066648"
##  [75,] "4.559454564" "83.83469087"   "21.75250108"    "13.19600001"
##  [76,] "11.30813809" "81.73160578"   "18.11522416"    "8.884"
##  [77,] "11.14118569" "73.4277902"    NA              NA
##  [78,] NA           NA             NA              NA
##  [79,] NA           NA             NA              NA
##  [80,] NA           NA             NA              NA
##  [81,] NA           NA             NA              NA
##  [82,] NA           NA             NA              NA
##  [83,] NA           NA             NA              NA
##  [84,] NA           NA             NA              NA
##  [85,] NA           NA             NA              NA
##  [86,] NA           NA             "0"             NA
##  [87,] NA           NA             "0.722499998"    NA
##  [88,] "14.82972693" "27.8320161"    "11.66515037"    "19.67280888"
##  [89,] "25.18068669" "66.2065951"    "6.270780242"    "13.77096624"
##  [90,] "53.87786531" "96.0534899"    "1.451615671"    "0"
##  [91,] "0.532"      NA             NA              NA
##  [92,] NA           NA             NA              NA
##  [93,] NA           NA             NA              NA
##  [94,] NA           NA             NA              NA
##  [95,] NA           NA             NA              NA
##  [96,] NA           NA             NA              NA
##  [97,] NA           NA             NA              NA
##  [98,] NA           NA             NA              NA
##  [99,] NA           NA             NA              NA
##       V13           V14           V15           V16
##   [1,] "1902-144-594" "ULAC_131_199" "ULAC_186_201" "ULAC_201_12"
##   [2,] NA            "0"           NA            NA
##   [3,] "0"           "0"           NA            NA
##   [4,] "0"           "0"           NA            NA
##   [5,] "0"           "0"           "0"           "0"
##   [6,] "0.886795836"  "0"           "0"           "0"
##   [7,] "9.237392886"  "0"           "0"           "0"
##   [8,] "28.85007411"  "0"           "0"           "0"
##   [9,] "13.99543742"  "0"           "0"           "0"
##  [10,] "20.55027533"  "0"           "0"           "0"
##  [11,] "1.898018551"  "0"           "0"           "0"
##  [12,] "0"           "0"           "0"           "0"
##  [13,] "0"           "0"           "0"           "0"
##  [14,] "0"           "0"           "2.855101925"  "0"
##  [15,] "0"           "0"           "0.375768254"  "0"
```

```
##  [16,] NA           "0"          NA           NA
##  [17,] "0"          "1.627999978" NA           NA
##  [18,] "0"          "7.488800028" NA           "0"
##  [19,] "0"          "2.294099988" "0"          "0.434920307"
##  [20,] "5.393405155" "4.843100012" "0"          "0.221603238"
##  [21,] "15.42560659" "0"          "0"          "0"
##  [22,] "14.88650392" "0"          "0"          "0"
##  [23,] "12.43147386" "0"          "0"          "0"
##  [24,] "9.293264945" "0"          "0"          "0"
##  [25,] "16.11775321" "0"          "0"          "0"
##  [26,] "0"          "0"          "0"          "0"
##  [27,] "0"          "0"          "0"          "0"
##  [28,] "0"          "0"          "0"          "0"
##  [29,] "0"          "0"          "0"          "0"
##  [30,] "0"          "0"          NA           NA
##  [31,] "0.239791107" "0.90444443"  NA           NA
##  [32,] "2.944101865" "4.160444446" "0"          "0"
##  [33,] "4.03E-09"    "1.274499989" "0"          "2.054866392"
##  [34,] "13.58909796" "2.690611114" "0"          "1.329619434"
##  [35,] "29.42633491" "0"          "0"          "0"
##  [36,] "7.418020704" "0"          "0"          "0"
##  [37,] "16.97875454" "0.663755557" "0"          "0"
##  [38,] "12.70463809" "2.895900002" "0"          "0"
##  [39,] "24.45671122" "1.168299997" "0"          "0"
##  [40,] "22.30686562" "0"          "0"          "0"
##  [41,] "1.248803186" "0"          "0"          "0"
##  [42,] "0"          "0"          "0"          "0"
##  [43,] "0"          "0"          "0"          "0"
##  [44,] "0"          "0.172666665" "0"          NA
##  [45,] "0.762439632" "1.438888891" "0"          "0"
##  [46,] "5.895908753" "0"          "0"          "0.086040486"
##  [47,] "2.791142816" "0"          "0"          "1.063384614"
##  [48,] "0"          "0"          "0"          "0.521052632"
##  [49,] NA           "0"          "0"          "0"
##  [50,] NA           "0"          NA           "0"
##  [51,] NA           "0.663755558" NA           NA
##  [52,] NA           "2.895900002" NA           NA
##  [53,] NA           "1.168299997" NA           NA
##  [54,] "0"          "0"          NA           NA
##  [55,] "0"          "0"          NA           NA
##  [56,] "0"          "0"          NA           NA
##  [57,] "0"          "0"          NA           NA
##  [58,] "0"          "0.401032255" "0"          "0"
##  [59,] "2.144361472" "2.59000001"  "0"          "0"
##  [60,] "10.33609189" "0"          "0"          "0.472882589"
##  [61,] "2.574177014" "0"          "0"          "1.552858298"
```

```
##  [62,] "0"           "0"           "0"           "0.086842105"
##  [63,] NA            "1.102097085"  NA            "0"
##  [64,] NA            "1.091499998"  NA            NA
##  [65,] NA            "0"            NA            NA
##  [66,] NA            "0"            NA            NA
##  [67,] NA            "0"            NA            NA
##  [68,] NA            "0"            NA            NA
##  [69,] NA            "0"            NA            NA
##  [70,] NA            "0"            NA            NA
##  [71,] NA            "0"            NA            NA
##  [72,] NA            "0"            "0"           "3.267076026"
##  [73,] "0"           "0"           "0"           "0"
##  [74,] "0"           "0"           "0"           "0.244886486"
##  [75,] "0"           "0"           "0"           "1.555246962"
##  [76,] "0"           "0"           "0"           "0.650291496"
##  [77,] NA            "8.732"       NA            "0"
##  [78,] NA            "4.366"       NA            NA
##  [79,] NA            "0"            NA            NA
##  [80,] NA            "0"            NA            NA
##  [81,] NA            "0"            NA            NA
##  [82,] NA            "0"            NA            NA
##  [83,] NA            "0"            NA            NA
##  [84,] NA            "0"            NA            NA
##  [85,] NA            "0"            NA            NA
##  [86,] NA            NA             NA            NA
##  [87,] "0"           "0"           NA            NA
##  [88,] "0.358734047"  "0"           "0"           "0"
##  [89,] "1.024954422"  "0"           "4.818519033"  "0"
##  [90,] NA            NA             "0.20923327"   "0"
##  [91,] NA            NA             NA            "0"
##  [92,] NA            NA             NA            NA
##  [93,] NA            NA             NA            NA
##  [94,] NA            NA             NA            NA
##  [95,] NA            NA             NA            NA
##  [96,] NA            NA             NA            NA
##  [97,] NA            NA             NA            NA
##  [98,] NA            NA             NA            NA
##  [99,] NA            NA             NA            NA
```

```
colnames(WE_Ext)<-WE_Ext[1,]
```

```
data_numWE<-as.data.frame(apply(WE_Ext,2, as.numeric))
```

```
## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion
```

```
## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

## Warning in apply(WE_Ext, 2, as.numeric): NAs introduced by coercion

sapply(data_numWE, class
)

## 1892-93-286-180 1892-93-307-197 1894-95-142-277 1898-99-232-476     1879-73-121
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric"
##       1893-94-8  1893-94-86-210    1899-288-512      1886-87-99  1890-91-21-129
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric"
## 1892-93-285-179      1906-07-37    1902-144-594    ULAC_131_199    ULAC_186_201
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric"
##     ULAC_201_12
##       "numeric"

View(data_numWE
)

Estim.NA <- estim_ncpPCA(data_numWE, ncp.min = 0, ncp.max = 10, method =
                         "Regularized", scale = TRUE, method.cv = "gcv",
```

```
                    nbsim = 100, pNA = 0.05, threshold=1e-4, verbose =
                     TRUE)
Estim.NA

## $ncp
## [1] 1
##
## $criterion
##         0         1         2         3         4         5         6
##  111.01937  99.54354 107.94164 123.29005 127.92520 151.92569 195.00295
##         7         8         9        10
##  268.49140 446.81677 852.29455 4390.63806

Impute.PCA <- imputePCA(data_numWE, ncp = 2) #based on the results of Estim.NA

WE.complete.sample_HS <- Impute.PCA$completeObs
WE.complete.sample_HS <- as.data.frame(WE.complete.sample_HS)

write.table(WE.complete.sample_HS, "~/Desktop/WE_NA_Test.txt")
("~/Desktop/WE_NA_Test.txt")

## [1] "~/Desktop/WE_NA_Test.txt"

SA_NA_Imp<-read.table("~/Desktop/AG1_NA_Test.txt")
SA_NA_Imp<-t(SA_NA_Imp)

GI_NA_Imp<-read.table("~/Desktop/GI_NA_Test.txt")
GI_NA_Imp<-GI_NA_Imp[-1,]
GI_NA_Imp<-t(GI_NA_Imp)

WE_NA_Imp<-read.table("~/Desktop/WE_NA_Test.txt")
WE_NA_Imp<-WE_NA_Imp[-1,]
WE_NA_Imp<-t(WE_NA_Imp)


NA_All_Ext<-rbind(GI_NA_Imp,SA_NA_Imp,WE_NA_Imp)

NA_All_Ext<-as.data.frame(apply(NA_All_Ext,2, as.numeric))
sapply(NA_All_Ext, class
)

##         2         3         4         5         6         7         8         9
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##        10        11        12        13        14        15        16        17
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##        18        19        20        21        22        23        24        25
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##        26        27        28        29        30        31        32        33
```

```
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       34       35       36       37       38       39       40       41
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       42       43       44       45       46       47       48       49
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       50       51       52       53       54       55       56       57
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       58       59       60       61       62       63       64       65
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       66       67       68       69       70       71       72       73
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       74       75       76       77       78       79       80       81
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       82       83       84       85       86       87       88       89
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       90       91       92       93       94       95       96       97
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       98       99
## "numeric" "numeric"

View(NA_All_Ext
)

Population=c("GI","GI","GI","GI","GI","GI","GI","GI","GI","GI","GI","GI","GI",
             "GI","GI","GI","SA","SA","SA","SA","SA","SA","SA","SA","SA","SA",
             "SA","SA","SA","SA","SA","SA","WE","WE","WE","WE","WE","WE","WE",
             "WE","WE","WE","WE","WE","WE","WE","WE","WE")
NA_All_Ext<-cbind(NA_All_Ext,Population)

Age=c("1","1","1","1","2","2","2","2","3","3","3","3","4","4","4","4","1","1",
      "1","1","2","2","2","2","3","3","3","3","4","4","4","4","1","1","1","1",
      "2","2","2","2","3","3","3","3","4","4","4","4")
NA_All_Ext<-cbind(NA_All_Ext,Age)
Data_all<-NA_All_Ext
BM_data <- as.matrix(Data_all[, 1:98])
Population <- Data_all$Population
Age <- Data_all$Age

Man_pop <- manova(BM_data ~ Population*Age, data = NA_All_Ext)
summary.aov(Man_pop)

##  Response 2 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2    5.36   2.682  0.0949 0.90970
## Age             3  204.80  68.267  2.4154 0.08242 .
## Population:Age  6  194.65  32.442  1.1478 0.35536
## Residuals      36 1017.48  28.263
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 3 :
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  86.89  43.444  4.0363 0.02621 *
## Age             3  48.63  16.210  1.5061 0.22947
## Population:Age  6  89.61  14.935  1.3876 0.24621
## Residuals      36 387.48  10.763
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 4 :
##             Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  277.62 138.808  2.1142 0.1355
## Age             3  359.65 119.885  1.8260 0.1599
## Population:Age  6  731.85 121.975  1.8578 0.1153
## Residuals      36 2363.56  65.654
##
##  Response 5 :
##             Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  113.52  56.762  0.6540 0.5260
## Age             3  670.07 223.358  2.5736 0.0691 .
## Population:Age  6  715.58 119.263  1.3742 0.2514
## Residuals      36 3124.33  86.787
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 6 :
##             Df  Sum Sq Mean Sq F value   Pr(>F)
## Population      2  110.18  55.088  1.7926 0.181068
## Age             3  494.91 164.968  5.3682 0.003693 **
## Population:Age  6  299.11  49.852  1.6222 0.169361
## Residuals      36 1106.30  30.731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 7 :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  74.47  37.236  2.3139 0.1134
## Age             3  57.21  19.068  1.1849 0.3291
## Population:Age  6  56.42   9.403  0.5843 0.7404
## Residuals      36 579.33  16.093
##
##  Response 8 :
##             Df Sum Sq Mean Sq F value Pr(>F)
```

```
## Population      2  37.19  18.597  1.0038 0.3765
## Age             3  44.01  14.668  0.7918 0.5065
## Population:Age  6  90.88  15.146  0.8175 0.5636
## Residuals      36 666.94  18.526
##
##  Response 9 :
##                 Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2   11.04   5.521  0.0748 0.92811
## Age             3  648.86 216.286  2.9290 0.04667 *
## Population:Age  6  271.80  45.300  0.6135 0.71797
## Residuals      36 2658.34  73.843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 10 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  204.89 102.445  1.4202 0.2549
## Age             3  150.08  50.027  0.6935 0.5620
## Population:Age  6  503.19  83.864  1.1626 0.3476
## Residuals      36 2596.83  72.134
##
##  Response 11 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  86.34  43.172  1.6715 0.2022
## Age             3  20.06   6.688  0.2589 0.8545
## Population:Age  6 142.62  23.770  0.9203 0.4919
## Residuals      36 929.80  25.828
##
##  Response 12 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  143.79  71.895  0.8351 0.4421
## Age             3  219.44  73.147  0.8496 0.4760
## Population:Age  6  425.49  70.914  0.8237 0.5592
## Residuals      36 3099.44  86.096
##
##  Response 13 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  129.4  64.692  0.5001 0.6106
## Age             3  359.2 119.734  0.9255 0.4383
## Population:Age  6  431.9  71.988  0.5564 0.7617
## Residuals      36 4657.3 129.370
##
##  Response 14 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  113.69  56.844  0.9120 0.4108
## Age             3  169.30  56.433  0.9054 0.4480
```

```
## Population:Age  6  324.17  54.029  0.8669 0.5285
## Residuals      36 2243.80  62.328
##
##  Response 15 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2   26.20  13.100  0.3936 0.67747
## Age             3   55.26  18.418  0.5534 0.64914
## Population:Age  6  403.43  67.239  2.0204 0.08823 .
## Residuals      36 1198.07  33.280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 16 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   8.30   4.149  0.2009 0.8189
## Age             3 125.73  41.911  2.0294 0.1271
## Population:Age  6  50.87   8.479  0.4106 0.8671
## Residuals      36 743.47  20.652
##
##  Response 17 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   63.74  31.871  0.8823 0.4226
## Age             3   82.49  27.496  0.7611 0.5233
## Population:Age  6  105.65  17.609  0.4875 0.8133
## Residuals      36 1300.46  36.124
##
##  Response 18 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2   84.25  42.124  1.0880 0.34772
## Age             3  379.86 126.619  3.2702 0.03218 *
## Population:Age  6  506.44  84.406  2.1800 0.06777 .
## Residuals      36 1393.86  38.718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 19 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   52.35  26.174  0.6854 0.5103
## Age             3  255.92  85.307  2.2338 0.1010
## Population:Age  6  371.63  61.938  1.6219 0.1694
## Residuals      36 1374.78  38.188
##
##  Response 20 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  27.46  13.731  0.6725 0.5167
## Age             3 116.23  38.745  1.8977 0.1474
```

```
## Population:Age  6 115.42  19.236  0.9422 0.4774
## Residuals      36 734.99  20.416
##
##  Response 21 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   3.90  1.9503  0.1436 0.8668
## Age             3  37.17 12.3909  0.9121 0.4448
## Population:Age  6  49.88  8.3137  0.6120 0.7191
## Residuals      36 489.05 13.5846
##
##  Response 22 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   2.646  1.3232  0.1521 0.8595
## Age             3  29.986  9.9952  1.1488 0.3426
## Population:Age  6  32.221  5.3701  0.6172 0.7151
## Residuals      36 313.218  8.7005
##
##  Response 23 :
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  16.46   8.230  0.3647 0.69693
## Age             3 189.12  63.040  2.7937 0.05416 .
## Population:Age  6  90.24  15.041  0.6665 0.67702
## Residuals      36 812.34  22.565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 24 :
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  41.90  20.949  0.8445 0.43810
## Age             3 170.33  56.775  2.2887 0.09496 .
## Population:Age  6 261.95  43.658  1.7599 0.13540
## Residuals      36 893.03  24.806
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 25 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   61.10  30.550  0.4504 0.6409
## Age             3  103.53  34.510  0.5088 0.6787
## Population:Age  6  207.86  34.643  0.5108 0.7961
## Residuals      36 2441.72  67.826
##
##  Response 26 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  263.6 131.794  1.1156 0.3388
## Age             3  205.5  68.484  0.5797 0.6321
```

```
## Population:Age  6  622.6 103.766  0.8783 0.5205
## Residuals      36 4253.0 118.140
##
##  Response 27 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  276.8 138.395  1.1323 0.3335
## Age            3  469.7 156.573  1.2810 0.2956
## Population:Age  6  291.0  48.493  0.3968 0.8762
## Residuals      36 4400.1 122.224
##
##  Response 28 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  403.9 201.950  1.7017 0.1967
## Age            3  415.0 138.328  1.1656 0.3363
## Population:Age  6  510.7  85.124  0.7173 0.6382
## Residuals      36 4272.3 118.676
##
##  Response 29 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  470.7  235.37  1.8534 0.1714
## Age            3  614.9  204.96  1.6139 0.2032
## Population:Age  6  687.4  114.56  0.9021 0.5042
## Residuals      36 4571.7  126.99
##
##  Response 30 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  266.2  133.11  0.8566 0.4331
## Age            3  622.5  207.51  1.3354 0.2781
## Population:Age  6  672.2  112.04  0.7210 0.6353
## Residuals      36 5594.2  155.40
##
##  Response 31 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  178.7  89.353  0.7842 0.4641
## Age            3  676.8 225.591  1.9799 0.1344
## Population:Age  6  445.9  74.313  0.6522 0.6881
## Residuals      36 4101.9 113.942
##
##  Response 32 :
##               Df  Sum Sq Mean Sq F value Pr(>F)
## Population     2  184.00  92.000  1.3920 0.2616
## Age            3  271.19  90.396  1.3677 0.2682
## Population:Age  6  158.21  26.369  0.3990 0.8748
## Residuals      36 2379.39  66.094
##
##  Response 33 :
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  290.57 145.286  2.0458 0.1440
## Age             3  318.20 106.068  1.4936 0.2327
## Population:Age  6  234.03  39.005  0.5493 0.7672
## Residuals      36 2556.57  71.016
##
##  Response 34 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   91.46  45.728  1.2477 0.2993
## Age             3   99.76  33.255  0.9073 0.4471
## Population:Age  6   90.10  15.017  0.4097 0.8677
## Residuals      36 1319.41  36.650
##
##  Response 35 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  120.01  60.003  0.9743 0.3872
## Age             3  331.52 110.508  1.7943 0.1657
## Population:Age  6  101.55  16.925  0.2748 0.9451
## Residuals      36 2217.14  61.587
##
##  Response 36 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  39.08 19.5408  1.7584 0.1868
## Age             3  63.56 21.1864  1.9065 0.1460
## Population:Age  6  31.93  5.3215  0.4789 0.8195
## Residuals      36 400.06 11.1128
##
##  Response 37 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  40.77 20.3864  0.9064 0.4130
## Age             3  28.19  9.3969  0.4178 0.7413
## Population:Age  6 179.90 29.9839  1.3331 0.2680
## Residuals      36 809.68 22.4912
##
##  Response 38 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  11.30   5.651  0.2420 0.7864
## Age             3  26.65   8.884  0.3804 0.7677
## Population:Age  6 294.45  49.075  2.1012 0.0772 .
## Residuals      36 840.79  23.355
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 39 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   69.59  34.796  0.4586 0.6358
```

```
## Age            3  122.14  40.715  0.5366 0.6602
## Population:Age  6  726.86 121.144  1.5967 0.1765
## Residuals      36 2731.30  75.869
##
##  Response 40 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2    5.7   2.833  0.0162 0.9840
## Age            3  533.9 177.973  1.0157 0.3971
## Population:Age  6  795.1 132.511  0.7562 0.6088
## Residuals      36 6308.2 175.227
##
##  Response 41 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  276.9  138.43  0.5730 0.5689
## Age            3 1205.0  401.68  1.6627 0.1923
## Population:Age  6 1518.6  253.11  1.0477 0.4114
## Residuals      36 8697.1  241.58
##
##  Response 42 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  818.8  409.41  1.4782 0.2416
## Age            3  888.7  296.24  1.0696 0.3741
## Population:Age  6 1076.3  179.38  0.6477 0.6916
## Residuals      36 9970.9  276.97
##
##  Response 43 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2 1224.0  611.98  2.6832 0.0820 .
## Age            3  931.1  310.38  1.3609 0.2702
## Population:Age  6  937.5  156.26  0.6851 0.6628
## Residuals      36 8210.8  228.08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 44 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  157.6  78.779  0.4568 0.6369
## Age            3  524.3 174.761  1.0134 0.3981
## Population:Age  6 1559.2 259.862  1.5068 0.2039
## Residuals      36 6208.4 172.456
##
##  Response 45 :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Population     2  120.7   60.37  0.5615 0.5753
## Age            3 1023.9  341.29  3.1745 0.0357 *
## Population:Age  6  754.6  125.77  1.1698 0.3439
```

```
## Residuals      36 3870.4  107.51
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 46 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  277.5 138.743  1.5455 0.2270
## Age           3  537.1 179.039  1.9943 0.1322
## Population:Age  6  826.6 137.770  1.5346 0.1950
## Residuals     36 3231.8  89.773
##
##  Response 47 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2  398.64 199.318  2.5079 0.09557 .
## Age           3  818.68 272.894  3.4337 0.02698 *
## Population:Age  6  272.13  45.356  0.5707 0.75087
## Residuals     36 2861.15  79.476
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 48 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  160.7  80.348  0.6808 0.5126
## Age           3  691.1 230.369  1.9518 0.1387
## Population:Age  6  231.1  38.518  0.3263 0.9188
## Residuals     36 4249.0 118.028
##
##  Response 49 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2  534.6  267.30  2.4723 0.09861 .
## Age           3 1046.7  348.92  3.2272 0.03371 *
## Population:Age  6  779.5  129.91  1.2016 0.32788
## Residuals     36 3892.2  108.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 50 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2  201.95 100.974  1.9077 0.16313
## Age           3  508.48 169.493  3.2022 0.03464 *
## Population:Age  6  367.93  61.321  1.1585 0.34974
## Residuals     36 1905.47  52.930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 51 :
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   48.42  24.209  0.7646 0.4729
## Age             3   97.73  32.578  1.0289 0.3914
## Population:Age  6  253.36  42.226  1.3336 0.2678
## Residuals      36 1139.89  31.664
##
##  Response 52 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2   31.48  15.741  0.3632 0.69794
## Age             3   36.39  12.130  0.2799 0.83952
## Population:Age  6  540.80  90.133  2.0799 0.07997 .
## Residuals      36 1560.05  43.335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 53 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   52.99  26.493  0.6602 0.5229
## Age             3  145.52  48.507  1.2087 0.3205
## Population:Age  6  242.59  40.432  1.0075 0.4358
## Residuals      36 1444.72  40.131
##
##  Response 54 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  266.4  133.18  1.2480 0.2992
## Age             3  652.7  217.57  2.0388 0.1258
## Population:Age  6  674.0  112.33  1.0526 0.4086
## Residuals      36 3841.8  106.72
##
##  Response 55 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  465.4  232.70  1.3270 0.2779
## Age             3  910.5  303.50  1.7307 0.1781
## Population:Age  6 2003.4  333.90  1.9041 0.1069
## Residuals      36 6313.0  175.36
##
##  Response 56 :
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  514.3  257.13  2.0018 0.14986
## Age             3 1052.8  350.94  2.7320 0.05797 .
## Population:Age  6  488.6   81.44  0.6340 0.70215
## Residuals      36 4624.3  128.45
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 57 :
```

```
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## Population      2  747.77  373.88  5.6255 0.0074828 **
## Age             3 1524.61  508.20  7.6464 0.0004438 ***
## Population:Age  6 1039.68  173.28  2.6072 0.0334542 *
## Residuals      36 2392.66   66.46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 58 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  256.46 128.229  3.1027 0.05712 .
## Age             3  228.90  76.300  1.8462 0.15627
## Population:Age  6 1151.20 191.867  4.6426 0.00137 **
## Residuals      36 1487.80  41.328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 59 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  149.2  74.583  0.7211 0.4931
## Age             3  640.8 213.601  2.0651 0.1221
## Population:Age  6  465.2  77.531  0.7496 0.6137
## Residuals      36 3723.5 103.431
##
##  Response 60 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  385.8  192.91  0.8525 0.4348
## Age             3  597.7  199.24  0.8804 0.4604
## Population:Age  6 2437.1  406.18  1.7948 0.1279
## Residuals      36 8146.9  226.30
##
##  Response 61 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  143.7   71.84  0.2998 0.7428
## Age             3  574.6  191.52  0.7994 0.5024
## Population:Age  6 1972.1  328.69  1.3719 0.2523
## Residuals      36 8625.1  239.59
##
##  Response 62 :
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  244.8  122.38  0.6818 0.51211
## Age             3  341.9  113.98  0.6350 0.59734
## Population:Age  6 2149.0  358.17  1.9954 0.09196 .
## Residuals      36 6462.0  179.50
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Response 63 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2   381.7  190.87  0.5369 0.58916
## Age           3  2723.1  907.69  2.5532 0.07068 .
## Population:Age  6  2867.9  477.99  1.3445 0.26333
## Residuals    36 12798.2  355.51
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 64 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  822.2  411.11  2.2546 0.1195
## Age           3 1055.9  351.98  1.9304 0.1421
## Population:Age  6 1428.8  238.13  1.3059 0.2796
## Residuals    36 6564.2  182.34
##
##  Response 65 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2   6.53  3.2636  0.1295 0.8789
## Age           3  71.27 23.7569  0.9427 0.4302
## Population:Age  6 140.63 23.4379  0.9300 0.4854
## Residuals    36 907.27 25.2020
##
##  Response 66 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2   5.75   2.874  0.1094 0.8967
## Age           3  31.13  10.378  0.3948 0.7575
## Population:Age  6 225.90  37.651  1.4324 0.2294
## Residuals    36 946.25  26.285
##
##  Response 67 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   17.83   8.917  0.2391 0.7885
## Age           3  141.08  47.027  1.2613 0.3022
## Population:Age  6  303.88  50.647  1.3584 0.2577
## Residuals    36 1342.27  37.285
##
##  Response 68 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2   74.68  37.341  0.6840 0.51105
## Age           3  387.17 129.056  2.3638 0.08731 .
## Population:Age  6  376.86  62.811  1.1505 0.35398
## Residuals    36 1965.47  54.596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Response 69 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  213.3  106.63  0.9249 0.4058
## Age           3  614.0  204.67  1.7754 0.1693
## Population:Age  6  797.6  132.94  1.1532 0.3526
## Residuals    36 4150.2  115.28
##
##  Response 70 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2  970.6  485.30  3.2151 0.05191 .
## Age           3  756.3  252.11  1.6702 0.19064
## Population:Age  6  584.8   97.47  0.6457 0.69308
## Residuals    36 5434.0  150.94
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 71 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2 2294.6 1147.31  4.6082 0.01652 *
## Age           3  674.2  224.72  0.9026 0.44941
## Population:Age  6  369.2   61.54  0.2472 0.95732
## Residuals    36 8963.0  248.97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 72 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  245.8  122.88  0.8694 0.4278
## Age           3  589.6  196.53  1.3904 0.2614
## Population:Age  6  635.7  105.94  0.7495 0.6138
## Residuals    36 5088.6  141.35
##
##  Response 73 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2   49.5  24.752  0.1845 0.8323
## Age           3  837.9 279.306  2.0820 0.1198
## Population:Age  6  323.0  53.830  0.4012 0.8733
## Residuals    36 4829.6 134.155
##
##  Response 74 :
##              Df Sum Sq Mean Sq F value   Pr(>F)
## Population    2  367.8  183.88  1.1944 0.314610
## Age           3 2089.9  696.65  4.5249 0.008593 **
## Population:Age  6 1128.9  188.14  1.2220 0.317883
## Residuals    36 5542.5  153.96
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 75 :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  726.6  363.31  1.4087 0.2576
## Age             3 1191.4  397.14  1.5399 0.2209
## Population:Age  6 2819.7  469.94  1.8222 0.1223
## Residuals      36 9284.5  257.90
##
##  Response 76 :
##             Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2   443.3  221.65  0.7876 0.46264
## Age             3   387.2  129.06  0.4586 0.71292
## Population:Age  6  3573.6  595.61  2.1163 0.07531 .
## Residuals      36 10132.0  281.44
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 77 :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  491.7  245.87  1.2886 0.2881
## Age             3  617.2  205.72  1.0782 0.3706
## Population:Age  6 1695.8  282.64  1.4813 0.2123
## Residuals      36 6868.9  190.80
##
##  Response 78 :
##             Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  278.23 139.114  1.7401 0.18994
## Age             3  930.43 310.144  3.8794 0.01679 *
## Population:Age  6  949.29 158.215  1.9790 0.09448 .
## Residuals      36 2878.06  79.946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 79 :
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2   8.44   4.218  0.2394 0.78831
## Age             3 141.13  47.042  2.6703 0.06208 .
## Population:Age  6 114.46  19.076  1.0828 0.39105
## Residuals      36 634.22  17.617
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 80 :
##             Df Sum Sq Mean Sq F value  Pr(>F)
```

```
## Population      2 122.83  61.415  2.2706 0.11784
## Age             3 296.26  98.753  3.6511 0.02138 *
## Population:Age  6 272.84  45.474  1.6812 0.15392
## Residuals      36 973.72  27.048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 81 :
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  86.51  43.257  1.9208 0.16120
## Age             3 420.57 140.191  6.2251 0.00162 **
## Population:Age  6 351.65  58.609  2.6025 0.03371 *
## Residuals      36 810.73  22.520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 82 :
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  42.52  21.260  1.1272 0.33509
## Age             3 342.37 114.124  6.0509 0.00191 **
## Population:Age  6 284.82  47.470  2.5169 0.03882 *
## Residuals      36 678.99  18.861
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 83 :
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  14.53   7.263  0.2872 0.75206
## Age             3 279.92  93.308  3.6897 0.02052 *
## Population:Age  6 302.71  50.451  1.9950 0.09202 .
## Residuals      36 910.40  25.289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 84 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  147.1  73.573  0.7282 0.4897
## Age             3  659.5 219.833  2.1759 0.1078
## Population:Age  6  831.9 138.657  1.3724 0.2521
## Residuals      36 3637.1 101.032
##
##  Response 85 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  319.0 159.516  1.4373 0.2509
## Age             3  293.9  97.975  0.8828 0.4592
## Population:Age  6  372.7  62.117  0.5597 0.7593
```

```
## Residuals      36 3995.5 110.986
##
##  Response 86 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2   19.63   9.813  0.1461 0.86458
## Age           3  615.14 205.046  3.0526 0.04077 *
## Population:Age  6  780.52 130.086  1.9367 0.10131
## Residuals     36 2418.14  67.170
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 87 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2  364.14 182.072  2.8322 0.07206 .
## Age           3  295.92  98.639  1.5344 0.22226
## Population:Age  6  353.92  58.987  0.9176 0.49373
## Residuals     36 2314.33  64.287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 88 :
##              Df Sum Sq Mean Sq F value   Pr(>F)
## Population    2  254.5  127.25  1.1434 0.330037
## Age           3 1816.5  605.50  5.4409 0.003439 **
## Population:Age  6 1241.7  206.96  1.8597 0.114990
## Residuals     36 4006.3  111.29
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 89 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2 1757.4  878.72  3.9759 0.02753 *
## Age           3 1335.2  445.05  2.0137 0.12936
## Population:Age  6 2317.5  386.25  1.7476 0.13814
## Residuals     36 7956.5  221.01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 90 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2  2141.7 1070.86  3.2856 0.0489 *
## Age           3  1381.2  460.40  1.4126 0.2550
## Population:Age  6  3203.9  533.99  1.6384 0.1650
## Residuals     36 11733.2  325.92
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Response 91 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2  743.4  371.72  3.1582 0.05449 .
## Age           3  189.0   62.99  0.5351 0.66118
## Population:Age  6  187.9   31.32  0.2661 0.94912
## Residuals    36 4237.3  117.70
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 92 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2   3.63   1.816  0.0678 0.9346
## Age           3 154.55  51.518  1.9225 0.1434
## Population:Age  6 186.35  31.058  1.1590 0.3495
## Residuals    36 964.69  26.797
##
##  Response 93 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2  33.38  16.690  0.8639 0.43005
## Age           3 153.16  51.054  2.6428 0.06399 .
## Population:Age  6 144.89  24.149  1.2501 0.30462
## Residuals    36 695.46  19.318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 94 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2  33.38  16.690  0.8639 0.43005
## Age           3 153.16  51.054  2.6428 0.06399 .
## Population:Age  6 144.89  24.149  1.2501 0.30462
## Residuals    36 695.46  19.318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 95 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2  12.75   6.376  0.3626 0.69835
## Age           3 166.20  55.401  3.1512 0.03661 *
## Population:Age  6 171.78  28.630  1.6284 0.16767
## Residuals    36 632.92  17.581
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 96 :
##              Df Sum Sq Mean Sq F value Pr(>F)
```

```
## Population      2   5.52   2.762  0.1643 0.8491
## Age             3 160.77  53.591  3.1875 0.0352 *
## Population:Age  6 190.36  31.727  1.8871 0.1099
## Residuals      36 605.25  16.813
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 97 :
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2   6.17   3.086  0.1639 0.84949
## Age             3 155.65  51.884  2.7546 0.05655 .
## Population:Age  6 167.10  27.850  1.4786 0.21323
## Residuals      36 678.09  18.836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 98 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  192.35  96.176  2.1659 0.1294
## Age             3  278.03  92.677  2.0871 0.1191
## Population:Age  6  266.27  44.379  0.9994 0.4408
## Residuals      36 1598.57  44.405
##
##  Response 99 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  376.88 188.438  2.4305 0.1023
## Age             3  440.25 146.750  1.8928 0.1483
## Population:Age  6  459.20  76.534  0.9871 0.4484
## Residuals      36 2791.09  77.530
```

```
library(missMDA)
library(Morpho)
library(geomorph)
library(ggplot2)
library(scales)
library(readr)
library(RColorBrewer)
library(fields)
library(raster)
library(colorRamps)
library(vegan)
library(proxy)
library(base)
library(pairwiseAdonis)
```

```
## Loading required package: cluster
```

```
library(readxl)
SA_Int <- read.table("~/Desktop/SA_NA_Int.txt")
as.matrix((SA_Int))
```

```
##       V1           V2           V3           V4           V5
##  [1,] "MMK_146"    "MMK_238"    "UCT_189"    "UCT_195"    "SAM_AP_1448"
##  [2,] NA           NA           "0"          "0"          NA
##  [3,] "59.92377896" "0"          "0"          "0"          NA
##  [4,] "59.92377896" "0"          "0"          "0"          NA
##  [5,] "78.6808641"  "0"          "0"          "0"          NA
##  [6,] "52.99885562" "0"          "0"          "0"          NA
##  [7,] "28.70201333" "0"          "0"          "0"          NA
##  [8,] "0"          NA           "0"          "0"          NA
##  [9,] "0"          NA           "0"          "0"          "0"
## [10,] "0"          NA           "0"          "0"          "0"
## [11,] "0"          NA           "0"          "0"          "0"
## [12,] "0"          NA           "0"          "0"          "0"
## [13,] "0"          NA           "0"          "0"          "0"
## [14,] "0"          NA           "0"          NA           "0"
## [15,] "0"          NA           "0"          NA           "0"
## [16,] "18.832"     "0"          "0"          "0"          NA
## [17,] "43.4754171"  "2.0472"     "0"          "0"          NA
## [18,] "58.26146737" "1.066249997" "0"          "0"          "0"
## [19,] "58.95251844" "0"          "0"          "0"          "17.16197181"
## [20,] "36.49691336" "0"          "0"          "0"          "34.18360843"
## [21,] "17.22120801" "0"          "0"          "0"          "12.18327634"
## [22,] "0"          "0"          "0"          "0.857997807" "1.348714168"
## [23,] "0"          "8.649666646" "0"          "1.715995614" "0.11239285"
## [24,] "0"          "8.245"      "0"          "1.429930944" "0"
## [25,] "0"          "3.26999999"  "0"          "0.863563316" "0"
## [26,] "0"          "0"          "0"          "0"          "0"
## [27,] "0"          "0"          "0"          "0"          "0"
## [28,] "0"          "0"          "0"          "0"          "0"
## [29,] "0"          "0"          "0"          "0"          "0"
## [30,] "18.832"     "0"          "0"          "0"          NA
## [31,] "33.60640002" "1.066249997" "0"          "0"          "0"
## [32,] "55.768"     "0.592361109" "1.010319123" "0"          "16.17000001"
## [33,] "29.36"      "0"          "2.020638247" "0"          "16.17000001"
## [34,] "11.744"     "0"          "1.010319123" "0"          "0.401"
## [35,] "0"          "0.287388889" "0"          "0"          "1.470333334"
## [36,] "0"          "3.448666671" "0"          "2.144994519" "0"
## [37,] "0"          "6.07949999"  "0"          "4.289989038" "0"
## [38,] "0"          "12.91691668" "0"          "3.574827363" "0"
## [39,] "0"          "8.443583332" "0"          "2.158908295" "9.909666676"
## [40,] "0"          "2.145111116" "7.490617495" "0"          "12.11633334"
## [41,] "0"          "0"          "14.981235"   "0"          "3.972"
## [42,] "0"          "0"          "7.490617495" "0"          "0"
```

```
##  [43,] "0"          "0"          "0"          "0"          "0"
##  [44,] "0"          "0"          "0"          "0"          "0"
##  [45,] "10.97600001" "0"          "0"          "0"          "0"
##  [46,] "27.44"      "0"          "1.010319123" "0"          "6.731666673"
##  [47,] "0"          "0"          "2.020638247" "12.7549632"  "10.61833333"
##  [48,] "0"          "0"          "1.922753197" "20.15413499" "6.995999999"
##  [49,] "0"          "0.287388888" "1.82486814"  "7.954545455" "0"
##  [50,] "0"          "3.448666671" "0.912434074" "3.977272728" "0"
##  [51,] "0"          "2.475472227" "0"          "0"          "11.16866664"
##  [52,] "0"          "9.481500024" "0"          "0"          "39.232"
##  [53,] "0"          "7.081083337" "0"          "0"          "61.52466668"
##  [54,] "0"          "2.145111116" "7.65780118"  "0"          "41.42133337"
##  [55,] "0"          "0"          "15.31560237" "0"          "0"
##  [56,] "0"          "0"          "7.65780118"  "0"          "0"
##  [57,] NA           "0"          "0"          "0"          "0"
##  [58,] "8.577600009" "0"          "0"          "0"          "0"
##  [59,] "9.536959988" "0"          "0"          "0"          "0"
##  [60,] "10.976"     "0"          "0"          "0"          "0"
##  [61,] "0"          "0"          "0"          "5.101985265" "3.086666663"
##  [62,] "0"          "0"          "7.29947265"  "8.339642071" "5.555999999"
##  [63,] "0"          "0"          "14.5989453"  "7.954545455" "0"
##  [64,] "0"          "0"          "7.29947265"  "3.977272719" "0"
##  [65,] "0"          "0"          "0"          "0"          "0"
##  [66,] "0"          "0"          "0"          "0"          NA
##  [67,] "0"          "0"          "0"          "0"          NA
##  [68,] "0"          "0"          "1.337469485" "0"          NA
##  [69,] "0"          "0"          "2.67493897"  "0"          NA
##  [70,] "0"          "0"          "1.337469485" "0"          NA
##  [71,] NA           "0"          "0"          "0"          NA
##  [72,] "11.4368"    NA           "0"          "0"          "24.52809361"
##  [73,] "6.862080002" "0"          "0"          "0"          "6.132023404"
##  [74,] "0"          "0"          "0.839706965" "0"          "0"
##  [75,] "0"          "0"          "1.679413936" "0"          "0"
##  [76,] "0"          "0"          "7.045432947" "0"          "0.149"
##  [77,] "0"          NA           "12.41145196" NA           "0.596"
##  [78,] NA           NA           "7.637816591" NA           NA
##  [79,] NA           NA           "0"          NA           NA
##  [80,] NA           NA           "0"          NA           NA
##  [81,] NA           NA           "0"          NA           NA
##  [82,] NA           NA           "1.337469485" NA           NA
##  [83,] NA           NA           "2.67493897"  "0"          NA
##  [84,] NA           NA           "1.337469485" "0"          NA
##  [85,] NA           NA           "0"          "0"          NA
##  [86,] "0"          NA           "0"          "0"          NA
##  [87,] "0"          NA           "0"          "0"          NA
##  [88,] "0"          "0"          "2.239218575" "0"          NA
```

```
##  [89,] "0"          "0"          "4.47843715"  "0"          NA
##  [90,] "0"          "0"          "6.62203344"  NA           "0"
##  [91,] NA           NA           "8.76562973"  NA           "0"
##  [92,] NA           NA           "8.76562973"  NA           NA
##  [93,] NA           NA           NA            NA           NA
##  [94,] NA           NA           NA            NA           NA
##  [95,] NA           NA           NA            NA           NA
##  [96,] NA           NA           NA            NA           NA
##  [97,] NA           NA           NA            NA           NA
##  [98,] NA           NA           NA            NA           NA
##  [99,] NA           NA           NA            NA           NA
##       V6           V7           V8           V9           V10
##   [1,] "SAM_AP_3027" "SAM_AP_6052" "SAM_AP_6340" "SAM_AP_1273" "SAM_AP_3
691"
##   [2,] NA           NA           NA           NA           NA
##   [3,] "0"          "0"          "0"          "51.94"      NA
##   [4,] "17.46774919" "0"          "0"          "59.80706661" "69.84190248"
##   [5,] "54.1219924"  "0"          "21.63071745" "82.44840001" "32.48089614"
##   [6,] "37.81237798" "0"          "1.347854865" "50.13883336" "20.54191656"
##   [7,] "8.593722278" "0"          "1.755926251" "4.113433335" "30.52792222"
##   [8,] "0"          "0"          "0"          "0"          "14.23826146"
##   [9,] "0"          "0"          "0"          "0"          "0"
##  [10,] "0"          "0"          "0"          "2.505466668" "0"
##  [11,] "0"          "0"          "15.91938174" "18.37320003" "0"
##  [12,] "0"          "0"          "0.953048353" "24.9108"     "0"
##  [13,] "0"          "0"          "0"          "11.08333329" "0"
##  [14,] "0"          "0"          "0"          "0.829599992" "0"
##  [15,] NA           "0"          "6.271695672" "0"          NA
##  [16,] NA           "0"          NA           "20.468"     NA
##  [17,] "69.848"     "0"          "0"          "23.74266662" "0"
##  [18,] "71.88563635" "11.95600003" "0.051748655" "37.14520002" "0"
##  [19,] "74.67763638" "15.09350003" "0"          "36.18839996" "7.317737729"
##  [20,] "41.616"     "5.647500002" "12.62346536" "35.93026668" "0"
##  [21,] "17.81600001" "0"          "2.341234995" "0"          "0"
##  [22,] "6.358"      "0"          "0"          "5.592"      "0"
##  [23,] "1.468181818" "0"          "11.64952918" "17.93413334" "0"
##  [24,] "0"          "0"          "4.549427854" "33.01906667" "0"
##  [25,] "0"          "0"          "0"          "28.38439998" "0"
##  [26,] "0"          "0"          "0"          "31.07213334" "0"
##  [27,] "0"          "0"          "0"          "26.16399998" "0"
##  [28,] "0"          "0"          "0"          "10.92399999" "0"
##  [29,] "0"          "0"          "4.782943899" "4.460000004" "0"
##  [30,] "0"          "0"          "1.25E-08"    "11.66159996" "0"
##  [31,] "7.057818188" "0"          "1.90825343"  "44.38666672" "5.6797099"
##  [32,] "13.38472726" "6.642222221" "0.020699462" "28.64493333" "0"
##  [33,] "15.97636364" "10.30055555" "10.82812404" "26.90000002" "0"
```

```
##  [34,] "5.568"       "7.919666665" "6.580946332" "9.287200005" "0"
##  [35,] "12.54109091" "5.98286112"  "0"           "0"           "0"
##  [36,] "21.782"      "13.06900002" "0"           "0.604"       "14.54494303"
##  [37,] "39.13109092" "15.86573941" "4.659811645" "9.583333336" "0"
##  [38,] "73.672"      "4.029997097" "1.81977113"  "15.9196"     "29.86488783"
##  [39,] "10.39899996" "0"           "5.171326911" "4.763999979" "22.02210041"
##  [40,] "2.350090893" "0"           "3.871025284" "17.47693336" "0"
##  [41,] "0"           "0"           "0"           "14.18933331" "0"
##  [42,] "0"           "0"           "0"           "2.182799989" "0"
##  [43,] "0"           NA            "0"           "0.03586207"  "0"
##  [44,] "5.551042513" "0"           "4.267461344" "5.486400006" "16.93816993"
##  [45,] "2.270881025" "0"           "1.649651356" "0"           "0"
##  [46,] "0"           "0"           "1.720359404" "0"           "0"
##  [47,] "0"           "1.915277778" "21.69895265" "1.419599994" "0"
##  [48,] "0"           "4.782166673" "0"           "2.996933335" "0"
##  [49,] "0"           "6.198071948" "1.540789647" "0"           "5.08492569"
##  [50,] "0"           "22.404"      NA            "0"           "0"
##  [51,] "0"           "27.19841038" NA            NA            "20.37248816"
##  [52,] "0"           "6.908566442" NA            "0"           "54.02829028"
##  [53,] "0"           "0"           NA            "0"           "0"
##  [54,] "0"           "0"           "0"           "0"           "0"
##  [55,] "0"           "0"           "0"           "0"           "0"
##  [56,] "0.819272725" "0"           "0"           "0"           "0"
##  [57,] "6.008"       NA            "0"           "0"           NA
##  [58,] "5.92"        "0"           "0"           "0"           NA
##  [59,] "2.421818179" "0"           "4.124128415" "0"           NA
##  [60,] "0"           "0"           "13.2917"     "0"           "0"
##  [61,] "0"           "0.55601177"  "26.41004401" "0"           "0"
##  [62,] "0"           "1.067542599" "15.46695209" "4.683066663" NA
##  [63,] "0"           "0"           "48.29641855" "14.42242967" NA
##  [64,] "0"           NA            NA            "16.58068386" NA
##  [65,] NA            NA            NA            NA            NA
##  [66,] NA            NA            NA            NA            NA
##  [67,] NA            "0"           NA            NA            NA
##  [68,] NA            "0"           NA            NA            "0"
##  [69,] NA            "0"           NA            NA            "0"
##  [70,] NA            "0"           NA            NA            "0"
##  [71,] NA            NA            NA            NA            NA
##  [72,] "0"           NA            NA            NA            NA
##  [73,] "0"           NA            "0"           NA            NA
##  [74,] "0"           "3.167501102" "21.90024617" "0"           "0"
##  [75,] "0"           "3.886418801" "43.06107216" "0.734399997" "0"
##  [76,] "0"           "4.892903568" "26.13083837" "11.89837994" NA
##  [77,] "0"           NA            "33.33333334" "28.22176349" NA
##  [78,] "0"           NA            NA            NA            NA
##  [79,] NA            NA            NA            NA            NA
```

```
##  [80,] NA           NA           NA           NA           NA
##  [81,] NA           NA           NA           NA           NA
##  [82,] NA           NA           NA           NA           NA
##  [83,] NA           NA           NA           NA           "0"
##  [84,] NA           NA           NA           NA           "0"
##  [85,] NA           NA           NA           NA           NA
##  [86,] NA           NA           NA           NA           NA
##  [87,] NA           NA           NA           NA           NA
##  [88,] NA           "38.01001252" "22.7357176"  "0"          "0"
##  [89,] "0"          "22.17250733" "5.721304586" "0"          "0"
##  [90,] "0"          "0"          "44.42209585" "0"          NA
##  [91,] "0"          NA           "0"          NA           NA
##  [92,] "0"          NA           NA           NA           NA
##  [93,] NA           NA           NA           NA           NA
##  [94,] NA           NA           NA           NA           NA
##  [95,] NA           NA           NA           NA           NA
##  [96,] NA           NA           NA           NA           NA
##  [97,] NA           NA           NA           NA           NA
##  [98,] NA           NA           NA           NA           NA
##  [99,] NA           NA           NA           NA           NA
##       V11          V12          V13          V14          V15
##   [1,] "SAM_AP_3737a" "SAM_AP_6348a" "SAM_AP_34"   "SAM_AP_1879" "SAM_AP_
4790"
##   [2,] NA           NA           NA           NA           NA
##   [3,] NA           NA           NA           "0"          NA
##   [4,] "10.11403551"  "0"          NA           "0"          "0"
##   [5,] "0"          "2.402399993"  NA           "0"          "0"
##   [6,] "11.384"      "15.08613332"  NA           "0"          "0"
##   [7,] NA           "19.32306667"  NA           "0"          "0"
##   [8,] NA           "4.437999992"  NA           "3.764000001" "0"
##   [9,] NA           "0"          NA           "0"          "0"
##  [10,] NA           "0"          NA           "4.170666668" "0"
##  [11,] "0"          "0"          NA           "0"          "0"
##  [12,] "0"          "0"          "1.1"        "0"          "0"
##  [13,] "3.288"       "0"          "1.1"        "0"          "0"
##  [14,] "0"          "0"          NA           "0"          "0"
##  [15,] "0"          "0"          NA           "0"          "1.265939394"
##  [16,] NA           NA           NA           NA           NA
##  [17,] "0"          "0"          NA           "0"          NA
##  [18,] "19.84750888"  "0"          NA           "0"          "0"
##  [19,] "0"          "1.067733331"  NA           "0"          "0"
##  [20,] "2.846000001"  "6.704948126"  "1.402"      "22.76666671" "0"
##  [21,] "6.78"        "8.588029623"  "1.402"      "9.606666686" "0"
##  [22,] "0"          "1.972444441"  NA           "5.018666654" "0"
##  [23,] "10.984"      "0"          NA           "0"          "0.153742687"
##  [24,] "17.828"      "0"          NA           "5.560888875" "4.151052642"
```

```
##  [25,] "0"          "0"          NA          "0"          "0"
##  [26,] "4.047"      "4.605111125"  "2.838"      "0"          "0"
##  [27,] "0.822"      "1.001111099"  "1.281333329" "0"          "0"
##  [28,] "0"          "0"          "0.97"       "0"          "0"
##  [29,] "0"          "0"          "3.812"      "0"          "1.687919188"
##  [30,] NA           "0"          NA          "0"          "0"
##  [31,] "0"          "0"          NA          "0"          "0"
##  [32,] "9.621666654"  "0"          NA          "0"          "3.926999988"
##  [33,] "0"          "1.560533331"  NA          "1.73E-08"    "1.062526316"
##  [34,] "0"          "3.294459273"  "11.267"     "9.106666632" "0"
##  [35,] "2.824999996"  "1.08E-08"    "8.077666668" "17.11155552" "1.15E-08"
##  [36,] "0"          "1.24E-08"    "3.907"      "5.31E-08"    "2.42E-08"
##  [37,] "4.576666661"  "0"          "2.173"      "7.80E-08"    "0.061497075"
##  [38,] "7.428333324"  "3.526962966"  "3.712633334" "4.45E-08"    "1.660421047"
##  [39,] "0"          "1.670666658"  "4.186799999" "0"          "0"
##  [40,] "2.24833333"   "1.842044434"  "6.28346667"  "0"          "0"
##  [41,] "0"          "0.400444443"  "6.976"      "0"          "0"
##  [42,] "0"          "0"          NA          "0"          "0"
##  [43,] "0"          "0"          NA          "0"          "1.10E-08"
##  [44,] NA           "0"          "3.13"       "0"          "0"
##  [45,] "0"          "0"          "3.13"       "0"          "0"
##  [46,] "0"          "0"          NA          "0"          "0"
##  [47,] "0"          "3.610933314"  NA          "7.575555567" "0"
##  [48,] "0"          "7.623081489"  "0.245"      "14.04977776" "0"
##  [49,] "4.274999995"  "4.718564199"  "0.771066666" "0"          "5.016368315"
##  [50,] "0"          "6.125059298"  "1.3305"     "29.86566698" "11.91387472"
##  [51,] "0"          "0"          "1.202"      "43.85322823" "0"
##  [52,] "0"          "0"          "4.997"      "25.05413323" NA
##  [53,] "0"          "0"          "9.465100008" "0"          NA
##  [54,] "0"          "0"          "10.30053333" "0"          "0"
##  [55,] "0"          "0"          "9.972"      "0"          "0"
##  [56,] "0"          "0"          NA          "0"          "0"
##  [57,] "0"          "0"          NA          "0"          "6.169414068"
##  [58,] "26.49075421"  "0"          "1.31"       "0"          "0"
##  [59,] "1.478863333"  "0"          NA          "0"          "0"
##  [60,] "0"          "0"          "2.126"      "0"          "0"
##  [61,] "0"          "4.534266646"  "4.645399989" "0"          "0"
##  [62,] "0"          "9.572340757"  "8.775000002" "21.60844445" "5.81810527"
##  [63,] "7.694999999"  "0"          "5.754"      "0"          "0.215485374"
##  [64,] "0"          "0"          NA          NA          "0"
##  [65,] "0"          NA           NA          NA          NA
##  [66,] NA           NA           NA          NA          NA
##  [67,] NA           NA           NA          NA          NA
##  [68,] "0"          "0"          NA          NA          NA
##  [69,] "0"          "0"          NA          NA          NA
##  [70,] "0"          "0"          NA          NA          NA
```

```
##  [71,] "0"           "0"           NA           "0"           NA
##  [72,] "27.07340564"  NA            "5.774"      "0"           "0"
##  [73,] "5.422498875"  NA            "6.157"      "0"           "0"
##  [74,] "0"           "0"           "6.157"      "0"           "0"
##  [75,] "0"           "1.131199995"  "12.327"     "0"           "0"
##  [76,] "0"           "2.388088887"  "9.591300002" "6.581333335" "12.23431581"
##  [77,] "0.794551665"  "0"           "4.866"      "0"           "0.53808332"
##  [78,] NA            "0"           NA           NA           "0"
##  [79,] NA            NA            NA           NA           NA
##  [80,] NA            NA            NA           NA           NA
##  [81,] NA            NA            NA           NA           NA
##  [82,] NA            NA            NA           NA           NA
##  [83,] NA            NA            NA           NA           NA
##  [84,] NA            NA            NA           NA           NA
##  [85,] NA            NA            NA           NA           NA
##  [86,] "33.48257115"  NA            NA           "0"           NA
##  [87,] "0"           NA            NA           "0"           NA
##  [88,] "0"           "0"           NA           "0"           NA
##  [89,] "0"           "0"           "4.656"      "0"           "0"
##  [90,] "0"           "0"           "4.656"      "0"           "1.475763149"
##  [91,] "9.53461975"   "0"           NA           "0"           "0.071620689"
##  [92,] NA            NA            NA           NA           NA
##  [93,] NA            NA            NA           NA           NA
##  [94,] NA            NA            NA           NA           NA
##  [95,] NA            NA            NA           NA           NA
##  [96,] NA            NA            NA           NA           NA
##  [97,] NA            NA            NA           NA           NA
##  [98,] NA            NA            NA           NA           NA
##  [99,] NA            NA            NA           NA           NA
##        V16
##   [1,] "SAM_AP_4844"
##   [2,] NA
##   [3,] "0"
##   [4,] "0"
##   [5,] "0.549175778"
##   [6,] "1.234769518"
##   [7,] "0.253546872"
##   [8,] "0.134203122"
##   [9,] "1.770515614"
##  [10,] "1.482843734"
##  [11,] "2.498343733"
##  [12,] "1.024312493"
##  [13,] "2.200499976"
##  [14,] "0.444937494"
##  [15,] "2.839861739"
##  [16,] "0"
```

```
##  [17,] "0"
##  [18,] "0"
##  [19,] "4.203175782"
##  [20,] "5.743894539"
##  [21,] "0.929671868"
##  [22,] "6.141765612"
##  [23,] "14.14329687"
##  [24,] "15.60279683"
##  [25,] "15.99832815"
##  [26,] "4.184562504"
##  [27,] "8.068500017"
##  [28,] "1.631437487"
##  [29,] "14.70046086"
##  [30,] "0"
##  [31,] "1.532890622"
##  [32,] "5.240593751"
##  [33,] "17.68675"
##  [34,] "15.8069375"
##  [35,] "16.18465625"
##  [36,] "40.84825002"
##  [37,] "48.09189062"
##  [38,] "46.80664062"
##  [39,] "35.28082812"
##  [40,] "10.53926562"
##  [41,] "0.288203125"
##  [42,] "0"
##  [43,] "0"
##  [44,] "0"
##  [45,] "1.532890622"
##  [46,] "6.375578126"
##  [47,] "25.22579687"
##  [48,] "26.58853477"
##  [49,] "33.43614045"
##  [50,] "70.12300002"
##  [51,] "78.44425002"
##  [52,] "46.48499368"
##  [53,] "22.97290625"
##  [54,] "9.767515619"
##  [55,] "0.288203125"
##  [56,] "3.191906256"
##  [57,] "3.928500006"
##  [58,] "0.947375005"
##  [59,] "0.464921875"
##  [60,] "10.50889064"
##  [61,] "16.66460935"
##  [62,] "33.61615129"
```

```
##  [63,] "43.24565185"
##  [64,] NA
##  [65,] NA
##  [66,] "0"
##  [67,] "0"
##  [68,] "0"
##  [69,] "0"
##  [70,] "3.627812498"
##  [71,] "4.464999997"
##  [72,] "0.258374998"
##  [73,] "0.629065812"
##  [74,] "3.598177822"
##  [75,] "2.805468729"
##  [76,] "7.202718677"
##  [77,] "10.20224994"
##  [78,] NA
##  [79,] NA
##  [80,] NA
##  [81,] NA
##  [82,] "0"
##  [83,] "0"
##  [84,] "2.697500004"
##  [85,] "3.32"
##  [86,] "0"
##  [87,] "0.618177154"
##  [88,] "0.156171069"
##  [89,] "1.650600001"
##  [90,] "4.562015509"
##  [91,] "5.301840739"
##  [92,] NA
##  [93,] NA
##  [94,] NA
##  [95,] NA
##  [96,] NA
##  [97,] NA
##  [98,] NA
##  [99,] NA
```

```
colnames(SA_Int)<-SA_Int[1,]
SA_Int<-SA_Int[-1,]

data_numSA_Int<-as.data.frame(apply(SA_Int,2, as.numeric))
sapply(data_numSA_Int, class
)
```

```
##     MMK_146     MMK_238     UCT_189     UCT_195  SAM_AP_1448  SAM_AP_3027
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
```

```
## SAM_AP_6052 SAM_AP_6340 SAM_AP_1273 SAM_AP_3691 SAM_AP_3737a SAM_
AP_6348a
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##   SAM_AP_34 SAM_AP_1879 SAM_AP_4790 SAM_AP_4844
##   "numeric"   "numeric"   "numeric"   "numeric"

View(data_numSA_Int
)

Estim.NA <- estim_ncpPCA(data_numSA_Int, ncp.min = 0, ncp.max = 10, method =
                   "Regularized", scale = TRUE, method.cv = "gcv",
                  nbsim = 100, pNA = 0.05, threshold=1e-4, verbose =
                   TRUE)
Estim.NA

## $ncp
## [1] 2
##
## $criterion
##        0        1        2        3        4        5        6
##  123.30995  107.46408   97.74805  108.14239  125.01927  152.38486  192.81865
##        7        8        9       10
##  271.27916  407.13090  797.14016 3226.68423

Impute.PCA <- imputePCA(data_numSA_Int, ncp = 2) #based on the results of
#Estim.NA

SA.complete.sample_HS_Int <- Impute.PCA$completeObs
SA.complete.sample_HS_Int <- as.data.frame(SA.complete.sample_HS_Int)

write.table(SA.complete.sample_HS_Int, "~/Desktop/SA_NA_Test_Int.txt")
("~/Desktop/SA_NA_Test_Int.txt")

## [1] "~/Desktop/SA_NA_Test_Int.txt"

#AG 2
library(readxl)
GI_Int<- read.table("~/Desktop/GI_NA_Int.txt")
as.matrix((GI_Int))

##       V1          V2          V3          V4          V5
##   [1,] "KAL_0169"    "KAL_0707"    "KAL_1418"    "KAL_1800"    "KAL_0028"
##   [2,] NA          "0"          NA          NA          NA
##   [3,] "0"          "0"          NA          NA          "100"
##   [4,] "2.26526958"  "0"          NA          "0"          "86.37879143"
##   [5,] "6.795808788" "0"          NA          "0"          "30.12879145"
##   [6,] "0"          "0"          NA          "0"          "0"
##   [7,] "0"          "0"          NA          "0"          "0"
```

```
##   [8,] "19.23861319" "0"          NA           NA           "0"
##   [9,] "37.7128965"  "0"          NA           NA           "0"
##  [10,] "35.11417492" "0"          NA           NA           "0"
##  [11,] "15.67408865" "0"          NA           NA           "0"
##  [12,] "19.58044416" "0"          NA           NA           "0"
##  [13,] "12.90274345" "0"          NA           NA           "0"
##  [14,] "0"           "0"          NA           NA           "0"
##  [15,] "0"           NA           NA           NA           "0"
##  [16,] "0"           "0"          "3.376939"   NA           "100"
##  [17,] "0.925975902" "0"          "3.376939"   NA           "85.19174871"
##  [18,] "3.031307203" "7.476000021" "10.2527345" "0"          "29.59514763"
##  [19,] "6.730721638" "12.06600001" "17.12853"   "0"          "0"
##  [20,] "1.217599999" "8.262000003" "20.624105"  "0"          "0"
##  [21,] "7.017381811" "0"          "24.11968"   "0"          "0"
##  [22,] "16.89016791" "0"          "15.5696475" NA           "0"
##  [23,] "24.44081064" "0"          "7.019615"   NA           "1.164143759"
##  [24,] "22.34530496" "0"          "9.4929575"  NA           "4.268527149"
##  [25,] "12.08328956" "0"          "11.9663"    NA           "0"
##  [26,] "14.57323013" "0"          "11.23133"   NA           "2.816273902"
##  [27,] "9.432827888" "0"          "10.49636"   NA           "13.64840626"
##  [28,] "0"           "0"          "10.94741"   NA           "14.05196311"
##  [29,] "0"           "0"          "11.39846"   NA           "0.882574648"
##  [30,] "0"           "0"          "3.376939"   "0"          "7.519147644"
##  [31,] "1.746727274" "0"          "3.376939"   "0"          "33.61752198"
##  [32,] "4.180363628" "4.153333333" "10.2527345" "0"         "10.8851648"
##  [33,] "6.633090911" "18.53138888" "17.12853"   "0"          "0"
##  [34,] "3.044"       "25.88050005" "20.624105"  "0"          "0"
##  [35,] "17.54345454" "0"          "24.11968"   "0"          "0"
##  [36,] "13.3675"     "0"          "15.5696475" "0"          "0"
##  [37,] "4.532681818" "0"          "7.019615"   "0"          "0"
##  [38,] "3.192"       "0"          "9.4929575"  "0"          "31.24999987"
##  [39,] "6.697090911" "0"          "11.9663"    "0"          "56.24999998"
##  [40,] "7.062409092" "0"          "11.23133"   "0"          "0"
##  [41,] "4.22795455"  "0"          "10.49636"   "0"          "11.57341125"
##  [42,] "0"           "0"          "10.94741"   "0"          "59.5574889"
##  [43,] "0"           "0"          "11.39846"   "0"          "43.4078231"
##  [44,] "0"           "0"          "3.376939"   "0"          "15.19137321"
##  [45,] "0"           "0"          "3.376939"   "0"          "6.920836238"
##  [46,] "0"           "0"          "10.2527345" "0"          "22.70456029"
##  [47,] "0"           "12.35097222" "17.12853"  "0"          "15.128836"
##  [48,] "0"           "22.23175006" "20.624105"  "0"          "19.94234549"
##  [49,] "11.3930909"  "0"          "24.11968"   "0"          "4.699157692"
##  [50,] "12.176"      "0"          "15.5696475" "0"          "0"
##  [51,] "7.424363633" "0"          "7.019615"   "0"          "0"
##  [52,] "0"           "5.35001873" "9.4929575"  "0"          "0"
##  [53,] "0"           "2.972232624" "11.9663"   "0.425541664" "17.08181944"
```

```
##  [54,] "0"          "0"          "11.23133"    "0.425541664" "17.08181944"
##  [55,] "0"          "0"          "10.49636"    "0"           "32.66485658"
##  [56,] "0"          "0"          "10.94741"    "0"           "37.04463967"
##  [57,] "0"          NA           "11.39846"    "0"           "2.308598321"
##  [58,] "0"          "0"          "4.69863675"  "0"           "14.58079834"
##  [59,] "0"          "1.248"      "4.69863674"  "0"           "14.47815719"
##  [60,] "0"          "0.992999999" "13.04284463" "0"           "18.43652269"
##  [61,] "0"          "1.284249997" "21.38705247" "0"           "11.86627308"
##  [62,] "0"          "1.8072"     "22.55817857" "0"           "5.816326137"
##  [63,] "8.496542362" "0"          "24.11968"    "0"           "5.20465773"
##  [64,] "12.176"     "0"          "15.5696475"  "0"           NA
##  [65,] "7.424363633" "0"          "7.019615"    "0"           NA
##  [66,] "0"          "12.84004499" "9.4929575"   "0"           NA
##  [67,] "0"          "7.133358302" "11.9663"     "3.404333336" NA
##  [68,] "0"          "0"          "11.23133"    "3.404333336" NA
##  [69,] "0"          "0"          "10.49636"    "0"           NA
##  [70,] "0"          "0"          "14.08129857" "0"           NA
##  [71,] "0"          NA           "16.77000246" "0"           NA
##  [72,] "0"          NA           "8.66373"     "0"           NA
##  [73,] "0"          "5.719999986" "8.66373"     "2.60565518"  NA
##  [74,] "0"          "4.042753915" "21.413175"   "1.311874998" "82.1374541"
##  [75,] "0"          "1.719111982" "34.16262"    "0"           "41.38042372"
##  [76,] "0"          "0"          "34.16262"    "0"           "9.68051118"
##  [77,] "0"          NA           NA            "0"           NA
##  [78,] NA           NA           NA            "0"           NA
##  [79,] NA           NA           NA            "0"           NA
##  [80,] NA           NA           NA            "0"           NA
##  [81,] NA           NA           NA            "2.127708337" NA
##  [82,] NA           "0"          NA            "2.127708337" NA
##  [83,] NA           "0"          NA            "0"           NA
##  [84,] NA           "0"          "32.88463"    "0"           NA
##  [85,] NA           NA           "32.88463"    "0"           NA
##  [86,] NA           NA           "8.66373"     NA            NA
##  [87,] "0"          "0"          "8.66373"     "8.396"       NA
##  [88,] "0"          "4.821047015" "21.413175"   "3.49833333"  "30.7530722"
##  [89,] "0"          "8.264652015" "34.16262"    "0"           "13.45446909"
##  [90,] NA           NA           "34.16262"    "0"           "0"
##  [91,] NA           NA           NA            "0"           NA
##  [92,] NA           NA           NA            "0"           NA
##  [93,] NA           NA           NA            "0"           NA
##  [94,] NA           NA           NA            "0"           NA
##  [95,] NA           NA           NA            "0"           NA
##  [96,] NA           NA           NA            "0"           NA
##  [97,] NA           NA           NA            "0"           NA
##  [98,] NA           NA           "32.88463"    "0"           NA
##  [99,] NA           NA           "32.88463"    "0"           NA
```

```
##       V6           V7           V8           V9           V10
##  [1,] "KAL_0168"    "KAL_0401"    "KAL_0722"    "KAL_0122"    "KAL_0674"
##  [2,] NA           NA           NA           NA           NA
##  [3,] NA           NA           "0"          NA           NA
##  [4,] "33.7111399"  "35.18778825" "0"          NA           NA
##  [5,] "34.55668304" "32.06199849" "0"          NA           "20.90380665"
##  [6,] "52.31308923" "16.51824994" "0"          "0"          "34.124"
##  [7,] "11.88933846" "8.072249954" "0"          "0"          "27.09687995"
##  [8,] "0"          "16.59784922" "0"          "0"          "12.44345626"
##  [9,] "0"          "14.212"      "0"          "0.334999998" "0"
## [10,] "0"          "2.74"        "0"          "1.314166666" "0"
## [11,] "0"          "0"           "0"          "0.1545"      "0"
## [12,] "0"          "20.516"      "0"          "0"          "0"
## [13,] "0"          NA           "0"          "0"          "0"
## [14,] "0"          NA           "0"          "0"          "0"
## [15,] "0"          NA           "0"          "0"          "0"
## [16,] NA           NA           NA           NA           NA
## [17,] "0"          "21.256"      "0"          NA           NA
## [18,] "5.968903789" "37.85100001" "0"          "0"          "39"
## [19,] "15.81435257" "51.2785"     "0"          "0"          "13.52695166"
## [20,] "21.11827228" "65.08825"    "0"          "0"          "19.166"
## [21,] "4.799607345" "46.03074999" "1.781400007" "0"          "6.77421999"
## [22,] "0"          "13.01768749" "8.907000006" "0"          "3.110864066"
## [23,] "0"          "24.47949998" "1.78140001"  "0.446666664" "0.543"
## [24,] "0"          "12.92249998" "1.268400006" "1.752222223" "0"
## [25,] "0"          "17.26084965" "3.171000001" "0.206"       "0"
## [26,] "0"          "16.15100001" "0"          "0"          "0"
## [27,] "1.077545453" "19.404"      "0"          "0"          "0"
## [28,] "2.274818183" "0"          "0"          "0"          "0"
## [29,] "0"          "0"          "0"          "0"          "9.612229447"
## [30,] "0"          "29.364"      NA           "0"          NA
## [31,] "0"          "17.406"      "19.81233333" "0"          "0"
## [32,] "1.261060595" "17.27075"    "19.81233335" "2.30844444"  "35.63056821"
## [33,] "5.249287858" "44.20525"    "0"          "8.326888885" "25.88700001"
## [34,] "32.25433337" "71.978"      "0"          "3.50388889"  "20.42166667"
## [35,] "32.23398491" "29.57849999" "5.130705465" "0"          "1.782666668"
## [36,] "21.11083335" "27.67474422" "25.65352725" "0"          "0"
## [37,] "12.19124245" "47.08848557" "10.27897213" "0"          "0.301666666"
## [38,] "19.66300003" "23.10836147" "4.565866662" "0"          "10.20833334"
## [39,] "15.62495456" "37.77996297" "1.761666664" "0"          "13.14833335"
## [40,] "4.910393908" "10.3245"     "0"          "0"          "0"
## [41,] "0.598636361" "22.84975"    "0"          "0"          "0"
## [42,] "1.263787876" "15.12700001" "0.485800001" "0"          "0"
## [43,] "0"          "0"          "2.429"      "0"          "15.83402423"
## [44,] "0"          "31.84421736" "0"          "0"          "0"
## [45,] "0"          "37.01024999" "12.1922051"  "0"          "0"
```

```
##  [46,] "0"          "16.418"     "19.81233335" "0"          "19.38056824"
##  [47,] "1.263075745" "18.82300001" "0"          "0.764999997" "21.27533335"
##  [48,] "27.78766671" "32.99450002" "0"          "0.424999995" "15.34000001"
##  [49,] "31.21883339" "0"          "4.517496868" "0"          "1.782666668"
##  [50,] "21.11083335" "12.55024423" "35.49461814" "0"          "0"
##  [51,] "12.66283335" "20.61273558" "10.13378778" "0"          "0"
##  [52,] "21.73800003" "3.238611478" "3.861199999" "0"          "10.20833334"
##  [53,] "19.65359092" "6.710433558" "0"          "0"          "13.14833335"
##  [54,] "6.221833291" "6.641250003" "0"          "0"          "0"
##  [55,] "0"          "28.09675"    "0"          "0"          "0"
##  [56,] "0"          "25.8595"     "0.485800001" "0.050555555" "0"
##  [57,] "0"          "0"          "2.429"      "0.004148148" "10.49389677"
##  [58,] "0"          "10.55414845" "0"          "0"          "1.933000001"
##  [59,] "0"          "17.53374987" "1.325999999" "0"          "2.563000001"
##  [60,] "0"          "13.15507715" "3.314999997" "0"          "2.092000001"
##  [61,] "0"          "0"          "7.302744691" "5.757500005" "1.523"
##  [62,] "0"          "0"          "3.436585737" "3.198611112" "13.84975695"
##  [63,] "0"          "0"          "0"          "0"          "0"
##  [64,] "0"          "0"          NA           "0"          "0"
##  [65,] "1.131818183" "0"          "0"          NA           "0"
##  [66,] "4.98"       "0"          "0"          NA           "0"
##  [67,] "9.668727276" "0"          "0"          "0"          "0"
##  [68,] "3.147454521" "0"          NA           "0"          "0"
##  [69,] "0"          "8.978749931" "0"          "0"          "0"
##  [70,] "0"          "5.962499954" "0"          "0.2275"     "0"
##  [71,] "0"          "0"          "0"          "0.018666666" "0"
##  [72,] NA           "0"          "0"          "0"          "9.612182128"
##  [73,] "0"          "0"          "4.861999982" "0"          "9.60974019"
##  [74,] "0"          "1.433202863" "12.15499997" "0"          "8.368"
##  [75,] "0"          "3.237000004" "26.7767305"  "2.883749998" "6.092"
##  [76,] "0"          "0"          "26.7767305"  "1.602083325" "49.44702777"
##  [77,] "0"          "0"          NA           "0"          NA
##  [78,] "0"          "0"          NA           "0"          NA
##  [79,] NA           "0"          NA           NA           NA
##  [80,] NA           "0"          NA           NA           NA
##  [81,] NA           "0"          NA           NA           NA
##  [82,] NA           "0"          NA           NA           "0"
##  [83,] "0"          "0"          NA           NA           "0"
##  [84,] "0"          "0"          NA           NA           NA
##  [85,] "0"          "0"          NA           NA           "0"
##  [86,] NA           "0"          "0"          "0"          "30.29418498"
##  [87,] NA           "0"          "0"          "0"          "2.544882473"
##  [88,] "0"          "0"          "0"          "0"          NA
##  [89,] "0"          "3.984"      "0"          "0"          NA
##  [90,] NA           "0"          "0"          "0"          NA
##  [91,] NA           "0"          NA           "0"          NA
```

```
## [92,] NA          NA          NA          "0"         NA
## [93,] NA          NA          NA          NA          NA
## [94,] NA          NA          NA          NA          NA
## [95,] NA          NA          NA          NA          NA
## [96,] NA          "0"         NA          NA          NA
## [97,] NA          "0"         NA          NA          NA
## [98,] NA          NA          NA          NA          NA
## [99,] NA          "0"         NA          NA          NA
##       V11         V12         V13         V14         V15
##  [1,] "KAL_0877"   "KAL_1459"   "KAL_0109"   "KAL_0869_"   "KAL_1310"
##  [2,] NA          NA          "0"         "0.200865263" "0"
##  [3,] "0"         NA          "0"         "1.673877206" "0"
##  [4,] "7.193812829" "0"         "26.74658826" "0"         "3.044941654"
##  [5,] "21.58143854" "0"         "10.76389604" "0"         "13.08845"
##  [6,] "0"         "0"         "16.57511281" "0"         "8.346541674"
##  [7,] "0"         "0"         "22.16515442" "0"         "1.388616669"
##  [8,] "0"         "0"         "22.73925005" "0"         "5.495750008"
##  [9,] "0"         "0"         "5.406661757" "0"         "3.374366672"
## [10,] "0"         "0"         "0.297830883" "0"         "0"
## [11,] "0"         "0"         "0.95949264"  "1.347157763" "0"
## [12,] "0"         "0"         "0"         "0.449052587" "0"
## [13,] "1.875919406" "0"         "0"         "0"         "0"
## [14,] "6.87636529"  "0"         "0"         "0"         "0"
## [15,] "7.390283286" "0"         "0"         "0"         "0"
## [16,] NA          NA          "0"         "1.487311751" "0"
## [17,] "55.55555556" NA          "0"         "12.39426473" "0"
## [18,] "54.68251361" "2.812500004" "18.38827944" "0"         "1.65806666"
## [19,] "36.26976295" "1.640625005" "16.26891912" "0"         "7.601825007"
## [20,] "0"         "0"         "9.890617661" "0"         "7.204999999"
## [21,] "0.656299641" "0"         "7.053830894" "1.636199405" "2.562041667"
## [22,] "0.757268817" "0"         "1.974500003" "3.296744579" "0"
## [23,] "3.312734649" "0"         "0"         "2.605694337" "2.33891666"
## [24,] "7.364363539" "0"         "0.130301474" "2.201117491" "8.163416655"
## [25,] "11.0173732"  "0"         "3.518139713" "5.557795783" "14.34387499"
## [26,] "11.50906496" "0"         "0"         "1.64652616"  "9.455416624"
## [27,] "7.741761562" "0"         "0"         "0"         "5.669374982"
## [28,] "0.703073591" "0"         "0"         "0"         "1.986749991"
## [29,] "0"         "0"         "0"         "0"         "0.055775863"
## [30,] "29.837587"   NA          "0"         "1.351450411" "3.584853207"
## [31,] "36.62584922" "0"         "0"         "12.07903367" "0"
## [32,] "56.25468412" "1.102941175" "0"         "4.875711411" "0"
## [33,] "23.80736399" "0.911458333" "0.787963236" "5.350492862" "0"
## [34,] "1.47E-08"    "0"         "1.174919127" "4.216362801" "2.984025"
## [35,] "0.262519854" "0"         "7.071955884" "8.750369564" "4.611674999"
## [36,] "7.506817632" "0"         "12.021625"   "12.63246019" "0"
## [37,] "7.568482607" "0"         "13.34642646" "4.690249822" "8.94724167"
```

```
##  [38,] "2.945745402" "0"          "16.77385293" "3.962011495" "20.56458334"
##  [39,] "26.14682937" "0"          "5.307286767" "1.112791086" "27.14774999"
##  [40,] "23.76126561" "0"          "2.270985298" "0"           "31.89197903"
##  [41,] "4.685564885" "0"          "2.620367641" "0"           "13.43796822"
##  [42,] "3.411056804" "0"          "0"           "0"           "3.576149995"
##  [43,] "0.124516716" "0"          "0"           "0"           "0.100396554"
##  [44,] "22.92625109" "0"          "0"           "0"           "28.53963598"
##  [45,] "21.03237669" "0"          "0"           "1.264438979" "0"
##  [46,] "8.864970157" "4.237916666" "0"          "11.17044814" "0"
##  [47,] "3.171083067" "9.897638867" "0.787963236" "19.07841346" "0"
##  [48,] "6.694508733" "10.18749997" "11.84044851" "8.022427345" "0"
##  [49,] "0"           "0"           "7.071955884" "13.26905283" "0"
##  [50,] "0"           "0"           "12.021625"   "15.31044558" "0"
##  [51,] "0"           "0"           "13.34642646" "0"           "10.82786667"
##  [52,] NA            "0"           "19.89235292" "0"           "13.41813333"
##  [53,] NA            "0"           "5.307286767" "0"           "3.037199987"
##  [54,] "0"           "0"           "2.270985298" "0"           "33.9936664"
##  [55,] "0"           "0"           "2.620367641" "0"           "7.389927373"
##  [56,] "0"           "0"           "0"           "0"           "0"
##  [57,] "0"           "0"           "0"           "0"           "0"
##  [58,] "11.98103893" "0"           "0"           "0"           "36.36398214"
##  [59,] "19.23664463" "0"           "0"           "0.635723659" "0"
##  [60,] "1.830163541" "8.22325002"  "0"           "6.247765564" "0"
##  [61,] "0.318682861" "18.41074999" "0"           "13.27430328" "1.364549999"
##  [62,] "0.672774933" "19.56"       "8.786413038" "6.039126737" "2.880716679"
##  [63,] "0"           "0"           "0"           NA            "0"
##  [64,] NA            "0"           "0"           NA            NA
##  [65,] NA            "0"           "0"           NA            NA
##  [66,] NA            "0"           "5.2070718"   NA            NA
##  [67,] NA            "0"           "0"           NA            NA
##  [68,] NA            "0"           "0"           "0"           NA
##  [69,] NA            "0"           "0"           "0"           NA
##  [70,] NA            "0"           "0"           "0"           NA
##  [71,] NA            "0"           "0"           "0"           NA
##  [72,] "5.286950254" NA            NA            "0"           "0"
##  [73,] "20.87399451" "0"           "0"           "0.105577332" "0"
##  [74,] "0.58758098"  "2.260143886" "4.976608016" "0.75018807"  "0"
##  [75,] "0.832901686" "2.181666666" "1.143522059" "1.13371001"  "0.372149996"
##  [76,] "1.758348014" "0"           "2.029699838" NA            "1.119026699"
##  [77,] "0"           NA            NA            NA            "0"
##  [78,] NA            NA            NA            NA            NA
##  [79,] NA            NA            NA            NA            NA
##  [80,] NA            NA            "0"           NA            NA
##  [81,] NA            NA            "0"           NA            NA
##  [82,] NA            NA            "0"           NA            NA
##  [83,] NA            NA            "0"           NA            NA
```

```
##  [84,] NA           NA           "0"          NA           NA
##  [85,] NA           NA           "0"          NA           NA
##  [86,] "1.062477771" NA           NA           "0"          NA
##  [87,] "1.881471075" NA           NA           "0"          NA
##  [88,] "0"          "0"          "0.455733325" "4.307328638" "0"
##  [89,] "11.33438452" "0"          "0.093827449" "5.250838711" "0"
##  [90,] "33.39194857" "0"          "0.114781768" NA           "0"
##  [91,] "25.8103728"  NA           NA           NA           NA
##  [92,] NA           NA           NA           NA           NA
##  [93,] NA           NA           NA           NA           NA
##  [94,] NA           NA           NA           NA           NA
##  [95,] NA           NA           NA           NA           NA
##  [96,] NA           NA           NA           NA           NA
##  [97,] NA           NA           NA           NA           NA
##  [98,] NA           NA           NA           NA           NA
##  [99,] NA           NA           NA           NA           NA
##       V16
##   [1,] "KAL_1412"
##   [2,] NA
##   [3,] "0"
##   [4,] "0"
##   [5,] "0"
##   [6,] "0"
##   [7,] "0"
##   [8,] "0"
##   [9,] "0"
##  [10,] "0"
##  [11,] "0"
##  [12,] "0"
##  [13,] "0"
##  [14,] "0"
##  [15,] "0"
##  [16,] NA
##  [17,] "0"
##  [18,] "0"
##  [19,] "0"
##  [20,] "0"
##  [21,] "0"
##  [22,] "0"
##  [23,] "0"
##  [24,] "0"
##  [25,] "0"
##  [26,] "0"
##  [27,] "0"
##  [28,] "0"
##  [29,] "4.25211111"
```

```
##  [30,] "0"
##  [31,] "0"
##  [32,] "0"
##  [33,] "0"
##  [34,] "0"
##  [35,] "0"
##  [36,] "0"
##  [37,] "0"
##  [38,] "1.390520828"
##  [39,] "5.132093762"
##  [40,] "2.62915626"
##  [41,] "0"
##  [42,] "0"
##  [43,] "9.439221236"
##  [44,] "0"
##  [45,] "0"
##  [46,] "0"
##  [47,] "0"
##  [48,] "0"
##  [49,] "0"
##  [50,] "0"
##  [51,] "0"
##  [52,] "4.767499974"
##  [53,] "17.59575"
##  [54,] "9.014250015"
##  [55,] "0"
##  [56,] "0"
##  [57,] "0"
##  [58,] "0"
##  [59,] "0"
##  [60,] "0"
##  [61,] "0"
##  [62,] "0"
##  [63,] "0"
##  [64,] NA
##  [65,] NA
##  [66,] NA
##  [67,] NA
##  [68,] NA
##  [69,] NA
##  [70,] NA
##  [71,] NA
##  [72,] "0"
##  [73,] "0"
##  [74,] "0"
##  [75,] "0"
```

```
##  [76,] "0"
##  [77,] "0"
##  [78,] "0"
##  [79,] NA
##  [80,] NA
##  [81,] NA
##  [82,] NA
##  [83,] NA
##  [84,] NA
##  [85,] NA
##  [86,] NA
##  [87,] "0"
##  [88,] "0"
##  [89,] "0"
##  [90,] "0"
##  [91,] "0"
##  [92,] "0"
##  [93,] NA
##  [94,] NA
##  [95,] NA
##  [96,] NA
##  [97,] NA
##  [98,] NA
##  [99,] NA

colnames(GI_Int)<-GI_Int[1,]
GI_Int<-GI_Int[-1,]


data_numGI_Int<-as.data.frame(apply(GI_Int,2, as.numeric))
sapply(data_numGI_Int, class
)

## KAL_0169 KAL_0707 KAL_1418 KAL_1800 KAL_0028 KAL_0168 KAL_0401 KAL
_0722
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
## KAL_0122 KAL_0674 KAL_0877 KAL_1459 KAL_0109 KAL_0869_ KAL_1310 KA
L_1412
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"

View(data_numGI_Int
)

Estim.NA <- estim_ncpPCA(data_numGI_Int, ncp.min = 0, ncp.max = 10, method =
                           "Regularized", scale = TRUE, method.cv = "gcv",
                         nbsim = 100, pNA = 0.05, threshold=1e-4, verbose =
```

```
                    TRUE)
Estim.NA

## $ncp
## [1] 0
##
## $criterion
##        0        1        2        3        4        5        6        7
##  105.1723  116.0845  125.5997  117.9628  136.2760  146.2522  180.6628  233.5594
##        8        9       10
##  330.5976  542.6597 1097.8544

Impute.PCA <- imputePCA(data_numGI_Int, ncp = 2) #based on the results of
#Estim.NA

GI.complete.sample_HS_Int <- Impute.PCA$completeObs
GI.complete.sample_HS_Int <- as.data.frame(GI.complete.sample_HS_Int)

write.table(GI.complete.sample_HS_Int, "~/Desktop/GI_NA_Test_Int.txt")
("~/Desktop/GI_NA_Test_Int.txt")

## [1] "~/Desktop/GI_NA_Test_Int.txt"

#AG 3
library(readxl)
WE_Int<- read.table("~/Desktop/WE_NA_Int.txt")
as.matrix((WE_Int))

##       V1              V2              V3              V4
##   [1,] "1892-93-286-180" "1892-93-307-197" "1894-95-142-277" "1898-99-232-476"
##   [2,] NA              NA              "0"             NA
##   [3,] NA              "34.15714286"   "4.338011728"   "0"
##   [4,] "0"             "34.15714286"   "26.0280702"    "0"
##   [5,] "0.7950909"     "33.66640021"   "10.1220273"    "0.35466144"
##   [6,] "17.492"        "13.46656007"   "0"             "11.13269028"
##   [7,] "3.975454545"   "0"             "0"             "14.65443425"
##   [8,] "0"             "0"             "0"             "0"
##   [9,] "0"             "0"             "0"             "0"
##  [10,] "0"             "0"             "0"             "0"
##  [11,] "0"             "0"             "0"             "0"
##  [12,] "0"             "0"             "0"             "0"
##  [13,] "0"             "0"             "0"             "0"
##  [14,] "0"             "0"             "0"             "0"
##  [15,] "0"             "0"             "0"             "0"
##  [16,] "0"             NA              "0"             "0"
##  [17,] "0"             "34.15714286"   "4.338011702"   "0"
##  [18,] "0"             "34.15714286"   "26.0280702"    "0"
##  [19,] "2.294032603"   "33.66640021"   "10.12202729"   "0.08866536"
```

```
## [20,] "50.46871792"    "13.46656006"    "0"            "2.783172561"
## [21,] "11.47016318"    "0"              "0"            "3.663608555"
## [22,] "0"              "0"              "0"            "0"
## [23,] "0"              "0"              "0"            "0"
## [24,] "0"              "0"              "0"            "0"
## [25,] "0"              "0"              "0"            "0"
## [26,] "0"              "0"              "0"            "0"
## [27,] "0"              "0"              "0"            "0"
## [28,] "0"              "0"              "0"            "0"
## [29,] "0"              "0"              "0"            "0"
## [30,] "0"              "0"              "0"            "0"
## [31,] "0"              "1.846332031"    "0.672694652"  "2.437866665"
## [32,] "0"              "4.269642809"    "4.036167844"  "6.094666673"
## [33,] "1.164033262"    "4.680799979"    "2.626110706"  "2.592333336"
## [34,] "25.6087321"     "1.872319998"    "2.025439393"  "1.036933333"
## [35,] "11.65657549"    "0"              "7.068281615"  "0"
## [36,] "8.297736685"    "0"              "12.99037812"  "0"
## [37,] "7.514487001"    "0"              "18.91247464"  "0"
## [38,] "2.319333337"    "0"              "13.86376328"  "0"
## [39,] "2.377242428"    "0"              "8.29081015"   "0"
## [40,] "0.757272723"    "0"              "0"            "0"
## [41,] "1.639909089"    "7.529301656"    "0"            "0"
## [42,] "5.279594953"    "10.03906887"    "6.18490112"   "0"
## [43,] "13.32880745"    "0"              "22.26564395"  "0"
## [44,] "0"              "0.474999995"    "0"            "0"
## [45,] "0"              "0.284999998"    "0.130443185"  "3.284199994"
## [46,] "0"              "0"              "0.782659108"  "8.210500003"
## [47,] "0"              "0.472500001"    "1.360857301"  "2.592333336"
## [48,] "0"              "0.189"          "2.039603305"  "1.03693333"
## [49,] "18.58080563"    "0"              "8.07803612"   "0"
## [50,] "20.89782432"    "0"              "14.84614641"  "0"
## [51,] "12.8819777"     "0"              "21.6142567"   "0"
## [52,] "3.976"          "0"              "15.84430086"  "0"
## [53,] "4.075272727"    "0"              "8.976516246"  "0"
## [54,] "1.298181809"    "0"              "0"            "0"
## [55,] "2.811272719"    "7.529301656"    "0"            "0"
## [56,] "9.050734193"    "10.36377543"    "6.18490112"   "0"
## [57,] "22.84938415"    "1.623532792"    "22.26564395"  "0"
## [58,] NA               "3.8"            "0"            "0"
## [59,] "0"              "2.280000002"    "0"            "1.792235293"
## [60,] "0"              "0"              "0"            "3.808500001"
## [61,] "0"              "0"              "0"            "0"
## [62,] "0"              "0"              "0"            "0"
## [63,] "28.43256923"    "0"              NA             "0"
## [64,] "39.58259639"    "0"              NA             NA
## [65,] NA               "0"              NA             NA
```

```
##  [66,] NA              "0"              NA               NA
##  [67,] NA              "0"              "0"              NA
##  [68,] NA              "0"              "0"              NA
##  [69,] NA              "0"              "0"              NA
##  [70,] NA              "2.597652489"    "0"              NA
##  [71,] NA              "12.98826246"    "0"              NA
##  [72,] NA              "2.375"          "0"              NA
##  [73,] NA              "1.425000003"    "0.127478444"    "0"
##  [74,] "0"             "0"              "0.764870664"    "0"
##  [75,] "0"             "0"              "0.71652414"     "0"
##  [76,] "0"             "0"              "0.685758173"    "0"
##  [77,] "0"             "0"              NA               NA
##  [78,] NA              "0"              NA               NA
##  [79,] NA              "0"              NA               NA
##  [80,] NA              "0"              NA               NA
##  [81,] NA              "0"              "0"              NA
##  [82,] NA              "0"              "0"              NA
##  [83,] NA              "0"              "0"              NA
##  [84,] NA              "2.597652498"    "0"              NA
##  [85,] NA              "12.98826246"    "0"              NA
##  [86,] NA              "0"              "0"              NA
##  [87,] NA              "0"              "0.339942517"    "0"
##  [88,] "0"             "0"              "2.0396551"      "0"
##  [89,] "0"             "0"              "1.910731042"    "0"
##  [90,] NA              "0"              "1.82868846"     "0"
##  [91,] NA              NA               NA               NA
##  [92,] NA              NA               NA               NA
##  [93,] NA              NA               NA               NA
##  [94,] NA              NA               NA               NA
##  [95,] NA              NA               NA               NA
##  [96,] NA              NA               NA               NA
##  [97,] NA              NA               NA               NA
##  [98,] NA              NA               NA               NA
##  [99,] NA              NA               NA               NA
##       V5           V6           V7           V8           V9
##  [1,] "1879-73-121" "1893-94-8"   "1893-94-86"  "1899-288-512" "1886-87-99"
##  [2,] NA            NA            NA            NA            NA
##  [3,] NA            "0"           NA            "0.635"       "0"
##  [4,] "0"           "12.02492203" "26.77646481" "8.914442697"  "22.54040464"
##  [5,] "10.89389525" "47.1916915"  "14.726"      "25.28282833"  "29.59873797"
##  [6,] "24.3852585"  "26.64571547" "17.23212492" "4.132131256"  "12.70500002"
##  [7,] "14.67277888" "4.909370201" "0"           "6.38602104"   "0"
##  [8,] "13.73375002" "0"           "0"           "0"            "0"
##  [9,] "4.501250005" "0"           "0"           "21.38602079"  "0"
## [10,] "0"           "0"           "0"           "19.86008621"  "0"
## [11,] "0.595466668" "0"           "0"           "2.852560774"  "0"
```

```
##  [12,] "0.372166667" "0"          "1.455749992" "0"          "0"
##  [13,] "0"           "0"          "0.641999996" "0"          "0"
##  [14,] "0"           "0"          "0"           "0"          "0"
##  [15,] "0"           "0"          "0"           "0"          "0"
##  [16,] NA            "0"          NA            NA           NA
##  [17,] "0"           "1.534666644" "0"          "13.0129959"   "0"
##  [18,] "8.430522711" "11.01586665" "7.838000005" "13.24332018"  "15.49652823"
##  [19,] "23.19907091" "18.36839999" "12.26650001" "15.44799746"  "20.3491324"
##  [20,] "18.20743271" "21.19466667" "3.761249993" "11.92340249"  "8.734687529"
##  [21,] "4.066888878" "11.55093334" "2.394999996" "0"          "0"
##  [22,] "27.01902783" "12.658"      "3.427499994" "8.032565005"  "0"
##  [23,] "19.9852778"  "10.97026666" "7.334999987" "9.747428493"  "0"
##  [24,] "3.377638895" "0"          "3.422499994" "1.802623159"  "0"
##  [25,] "1.216222223" "0"          "2.391249996" "0"          "0.489999998"
##  [26,] "0.49622222"  "3.164800005" "5.591500004" "0"          "0.489999998"
##  [27,] "0"           "0.68799999"  "4.690249998" "0"          "0"
##  [28,] "0"           "0"          "0.759999999" "0"          "0.1640625"
##  [29,] "0"           "0"          "0"           "0"          "1.968749997"
##  [30,] "0"           "0"          "14.33314496" "0"          "5.07731645"
##  [31,] "0.199193317" "0"          "1.228500001" "1.113127912"  "2.549185731"
##  [32,] "2.832926611" "0"          "1.3458125"   "6.634978472"  "2.2999375"
##  [33,] "7.45490681"  "8.701199962" "1.18575"     "10.77490882"  "1.391395832"
##  [34,] "3.601861101" "29.43666667" "7.040187498" "17.86904287"  "0"
##  [35,] "0.235999997" "19.13053334" "4.310999998" "9.88605677"   "0"
##  [36,] "8.366055515" "6.766"      "6.169499998" "3.000162062"  "0"
##  [37,] "7.193888869" "5.265599999" "28.1845378"  "0"          "1.565666662"
##  [38,] "4.032994375" "5.714533335" "6.160499998" "14.80904333"  "0.426999999"
##  [39,] "5.115489898" "2.26919999"  "4.304249998" "12.29732495"  "0.882000001"
##  [40,] "0"           "0"          "0.45675"     "1.760838317"  "0.882000001"
##  [41,] "0"           "8.13666669"  "4.205249999" "0"          "0"
##  [42,] "0"           "0.976399991" "1.368"       "0"          "0.2953125"
##  [43,] "0"           "0"          "0"           "0"          "3.543749997"
##  [44,] "4.083584507" "0"          "13.45275092" "0"          "9.46407595"
##  [45,] "0.746166664" "0"          "1.228500001" "0"          "4.001393988"
##  [46,] "0"           "0"          "0.7608125"   "0.687401886"  "6.444812506"
##  [47,] "10.50175002" "0"          "0"           "2.062205677"  "8.489395833"
##  [48,] "5.834305569" "0"          "0.2699375"   "0"          "5.315624997"
##  [49,] "0"           "0"          "0"           "2.221631146"  "0"
##  [50,] "0"           "0"          "0"           "2.563420554"  "0"
##  [51,] NA            "0"          "34.24351496" "3.146607959"  "3.23234407"
##  [52,] NA            "5.542933336" "0"          "11.83856809"  "0.975999998"
##  [53,] NA            "6.783599996" "0"          "4.643302549"  "0"
##  [54,] NA            "1.385999993" "0"          "0"          "0"
##  [55,] "0"           "0"          "0"           "0"          "0"
##  [56,] "0"           "0"          "0"           "0"          "0"
##  [57,] "0"           "0"          "0"           "0"          "0"
```

```
##  [58,] "10.20896131" "0"          "3.989999969" "23.81266976"  "30.43286079"
##  [59,] "1.865416668" "0"          "0"           "8.6525597"    "7.608215166"
##  [60,] "0"           "0"          "1.765500006" "0"           "2.302708312"
##  [61,] "0"           "0"          "11.04400004" "0"           "3.9433333"
##  [62,] "0"           "0"          "0"           "0"           "3.566037713"
##  [63,] "0"           "0"          NA            "0"           "0"
##  [64,] NA            NA           NA            "0"           "0"
##  [65,] NA            NA           NA            NA            "0"
##  [66,] NA            NA           NA            NA            NA
##  [67,] NA            NA           NA            NA            NA
##  [68,] NA            NA           NA            NA            NA
##  [69,] NA            NA           "0"           "0"           "0"
##  [70,] NA            NA           "0"           "0"           "0"
##  [71,] NA            NA           "0"           "0"           "0"
##  [72,] "0"           NA           "0"           NA            "38.41325196"
##  [73,] "0"           "0"          "0"           "0"           "9.603312947"
##  [74,] "0"           "0"          "0.481499997" "0"           "0"
##  [75,] "0"           "0"          "3.011999983" "0.479725869"  "0"
##  [76,] "0"           "0"          NA            "1.847584059"  "0"
##  [77,] "0"           NA           NA            "2.276807573"  NA
##  [78,] NA            NA           NA            NA            NA
##  [79,] NA            NA           NA            NA            NA
##  [80,] NA            NA           NA            NA            NA
##  [81,] NA            NA           NA            NA            NA
##  [82,] NA            NA           NA            NA            NA
##  [83,] NA            NA           NA            NA            NA
##  [84,] NA            NA           NA            NA            NA
##  [85,] NA            NA           NA            NA            NA
##  [86,] NA            NA           NA            NA            NA
##  [87,] NA            "0"          NA            NA            NA
##  [88,] "0"           "0"          "0"           "0"           "0"
##  [89,] "0"           "0"          "0"           "0"           "0"
##  [90,] "0"           NA           NA            "0"           "0"
##  [91,] NA            NA           NA            "0"           NA
##  [92,] NA            NA           NA            NA            NA
##  [93,] NA            NA           NA            NA            NA
##  [94,] NA            NA           NA            NA            NA
##  [95,] NA            NA           NA            NA            NA
##  [96,] NA            NA           NA            NA            NA
##  [97,] NA            NA           NA            NA            NA
##  [98,] NA            NA           NA            NA            NA
##  [99,] NA            NA           NA            NA            NA
##        V10           V11          V12           V13
##   [1,] "1890-91-21-129" "1892-93-285-179" "1906-07-37"  "1902-144-594"
##   [2,] NA              NA               NA            NA
##   [3,] NA              "0"              "0"           "4.575862992"
```

```
##  [4,] NA             "0"             "8.216913296" "18.78552739"
##  [5,] "0"            "7.794661277"   "30.54798207" "11.62992204"
##  [6,] "100"          "27.10874987"   "31.33079606" "1.004743819"
##  [7,] "7.122789622"  "41.67467851"   "14.50283332" "0.318396688"
##  [8,] "0"            "30.37215847"   "0.818416654" "1.439684368"
##  [9,] "14.21618852"  "0"             "1.817833337" "1.157535627"
## [10,] "0"            "0"             "0"           "2.953088324"
## [11,] "0"            "0"             "0"           "0"
## [12,] "0"            "0"             "0"           "0"
## [13,] "0"            "0"             "0"           "0"
## [14,] "0"            "0"             "0"           "0"
## [15,] "0"            "0"             "0"           "0"
## [16,] NA             NA              NA            "0"
## [17,] NA             "0"             "6.326666684" "0.586649099"
## [18,] "98.35376532"  "4.512499992"   "14.07071146" "10.77840204"
## [19,] "65.46363132"  "6.294499999"   "27.37371147" "5.371745705"
## [20,] "36.14554822"  "34.36675003"   "22.40936113" "0.48652831"
## [21,] "0"            "9.472499997"   "16.07727778" "0.195101447"
## [22,] "0"            "3.876249993"   "1.16477776"  "15.69983764"
## [23,] "100"          "8.204999986"   "4.042499997" "31.3306151"
## [24,] "100"          "7.186249987"   "11.40625"    "27.69633381"
## [25,] "93.71558535"  "6.358749989"   "11.97819449" "18.2285282"
## [26,] "30.28056404"  "4.206249993"   "0.824444445" "0"
## [27,] "0"            "0"             "0"           "0"
## [28,] "0"            "0"             "0"           "0"
## [29,] "46.84219028"  "0"             "0"           "0"
## [30,] NA             "0"             "5.17E-09"    "0"
## [31,] "0"            "0"             "9.700655693" "0"
## [32,] "100"          "8.122499997"   "21.42211108" "0"
## [33,] "100"          "4.162499999"   "31.94661109" "0"
## [34,] "100"          "0"             "47.10130557" "0"
## [35,] "0"            "7.100999997"   "25.8761667"  "0"
## [36,] "0"            "10.61621799"   "13.84138892" "0"
## [37,] "100"          "19.4670981"    "8.688506945" NA
## [38,] "77.62297413"  "12.93525"      "11.31414581" "0"
## [39,] "0"            "16.93275756"   "9.127777757" "0"
## [40,] "0"            "7.571249997"   "2.753527775" "0"
## [41,] "0"            "0"             "1.346527784" "0"
## [42,] "0"            "0"             "1.45E-09"    "0"
## [43,] "0"            "0"             "0"           "0"
## [44,] "99.12767645"  "0"             "7.775501361" "0"
## [45,] "0"            "0"             "6.242749986" "0"
## [46,] "0"            "0"             "0.950277774" "0"
## [47,] "56.47753362"  "0"             "12.8335278"  "0"
## [48,] "0"            "0"             "29.1557892"  "0"
## [49,] "0"            "0"             "44.39835166" NA
```

```
##  [50,] NA             "8.317641115"    "24.90976529" NA
##  [51,] NA             "10.73850996"    NA            NA
##  [52,] NA             "0"              NA            NA
##  [53,] NA             "12.54173158"    "0"           "0"
##  [54,] NA             "0"              "0"           "0"
##  [55,] NA             "0"              "2.986499999" "0"
##  [56,] NA             "2.832948348"    "0.814499995" "0"
##  [57,] NA             "0"              "0"           "0"
##  [58,] "100"          "0"              "22.45011134" "0"
##  [59,] "53.89299261"  "0"              "27.52030419" "0"
##  [60,] "0"            "0"              "3.944904886" "0"
##  [61,] "2.508133067"  "0"              "0.448194444" "0"
##  [62,] "0"            "0"              "0.799195103" "0"
##  [63,] NA             NA               "0.341158382" NA
##  [64,] NA             NA               "0"           NA
##  [65,] NA             NA               NA            NA
##  [66,] NA             NA               NA            NA
##  [67,] NA             NA               NA            NA
##  [68,] NA             "0"              NA            NA
##  [69,] NA             "0"              NA            NA
##  [70,] NA             "5.03635262"     NA            NA
##  [71,] NA             "0"              NA            NA
##  [72,] NA             NA               "8.62946145"  "0"
##  [73,] NA             "0"              "33.60287188" "0"
##  [74,] "0"            "0"              "1.883052526" "0"
##  [75,] "0"            "0"              "1.6815"      "0"
##  [76,] NA             "0"              "0.934166671" NA
##  [77,] NA             NA               "0"           NA
##  [78,] NA             NA               NA            NA
##  [79,] NA             NA               NA            NA
##  [80,] NA             NA               NA            NA
##  [81,] NA             NA               NA            NA
##  [82,] NA             NA               NA            NA
##  [83,] NA             NA               NA            NA
##  [84,] NA             NA               NA            NA
##  [85,] NA             NA               NA            NA
##  [86,] NA             NA               NA            "0"
##  [87,] NA             NA               NA            "0"
##  [88,] "0"            "0"              "0"           "0"
##  [89,] "0"            "0"              "0.315281248" "0"
##  [90,] NA             "0"              "0.439607839" NA
##  [91,] NA             NA               "0"           NA
##  [92,] NA             NA               NA            NA
##  [93,] NA             NA               NA            NA
##  [94,] NA             NA               NA            NA
##  [95,] NA             NA               NA            NA
```

```
## [96,] NA             NA             NA          NA
## [97,] NA             NA             NA          NA
## [98,] NA             NA             NA          NA
## [99,] NA             NA             NA          NA
##       V14          V15          V16
##  [1,] "ULAC_131_199" "ULAC_186_201" "ULAC_201_12"
##  [2,] NA           NA           NA
##  [3,] NA           "0"          NA
##  [4,] "0"          "0"          "0"
##  [5,] "14.23192772"  "0"          "4.503099302"
##  [6,] "13.10690249"  "0"          "0.138882355"
##  [7,] "0"          "0"          "0"
##  [8,] "0"          "0"          "0"
##  [9,] "0"          "0"          "0"
## [10,] "0"          "0"          "0"
## [11,] "0"          "0"          "0"
## [12,] "0"          "0"          "0"
## [13,] "0"          "0"          "0"
## [14,] "3.367307704"  "0"          "0"
## [15,] "0.677806457"  "0"          "0"
## [16,] "0"          "0"          NA
## [17,] "0"          "0"          NA
## [18,] "0"          "0.615547072"  "0"
## [19,] "0"          "0.205182356"  "13.74935296"
## [20,] "0"          "0.270117649"  "0.509235306"
## [21,] "0"          "1.242541177"  "0"
## [22,] "0"          "0"          "0"
## [23,] "0"          "4.854589226"  "0"
## [24,] "0"          "4.675184109"  "0.576470585"
## [25,] "0"          "0"          "0"
## [26,] "0"          "0"          "3.295882351"
## [27,] "0"          "0.959164705"  "3.802941168"
## [28,] "2.288888889"  "0.454341173"  "0"
## [29,] "0.301247314"  "0"          "0"
## [30,] "0"          "0"          NA
## [31,] "0"          "0"          NA
## [32,] "0"          "8.002111784"  "0"
## [33,] "0"          "2.66737059"   "0"
## [34,] "0"          "0"          "0"
## [35,] "0"          "0"          "0"
## [36,] "0"          "0"          "0"
## [37,] "0"          "2.499549201"  "0"
## [38,] "0"          "10.86579964"  "1.037647056"
## [39,] "0"          "0"          "0"
## [40,] "0"          "0"          "7.70240441"
## [41,] "0"          "0"          "8.88738971"
```

```
##  [42,] "0"           "0"           "0"
##  [43,] "0"           "0"           "0"
##  [44,] "0"           "0"           "0"
##  [45,] "0"           "0"           "0"
##  [46,] "0"           "0"           "0"
##  [47,] "0"           "0"           "0"
##  [48,] "0"           "0"           "0"
##  [49,] "0"           NA             "0"
##  [50,] "0"           NA             "0"
##  [51,] "0"           NA             "0"
##  [52,] "0"           NA             "0"
##  [53,] "0"           NA             "0"
##  [54,] "0"           NA             "2.134580885"
##  [55,] "0"           NA             "2.462977937"
##  [56,] "0"           NA             "0"
##  [57,] "0"           NA             "0"
##  [58,] "0"           "0"           "0"
##  [59,] "0"           "0"           "0"
##  [60,] "0"           "0"           "0"
##  [61,] "0"           "8.784052935"  "0"
##  [62,] "0"           "10.46298096"  "0"
##  [63,] NA             NA             "0"
##  [64,] NA             NA             "0"
##  [65,] NA             NA             "0"
##  [66,] NA             NA             "0"
##  [67,] NA             NA             "0"
##  [68,] NA             NA             "0.202647057"
##  [69,] NA             NA             "0.233823528"
##  [70,] NA             NA             "0"
##  [71,] NA             NA             "0"
##  [72,] "4.756862758"  NA              "0"
##  [73,] "0"           "0"           "0"
##  [74,] "0"           "0"           "0"
##  [75,] "0"           "11.83362354"  "0"
##  [76,] "0"           "34.96699616"  "0"
##  [77,] NA             NA             NA
##  [78,] NA             NA             NA
##  [79,] NA             NA             NA
##  [80,] NA             NA             "0"
##  [81,] NA             NA             "0"
##  [82,] NA             NA             "0"
##  [83,] NA             NA             "0"
##  [84,] NA             NA             "0"
##  [85,] NA             NA             "0"
##  [86,] "0.89191176"   NA              "0"
##  [87,] "0"           NA             "0"
```

```
##  [88,] "0"          "0"          "0"
##  [89,] "0"          "38.39737416"  "0"
##  [90,] "0"          "21.96100145"  "0"
##  [91,] NA           NA           NA
##  [92,] NA           NA           NA
##  [93,] NA           NA           NA
##  [94,] NA           NA           NA
##  [95,] NA           NA           NA
##  [96,] NA           NA           NA
##  [97,] NA           NA           NA
##  [98,] NA           NA           NA
##  [99,] NA           NA           NA
```

```
colnames(WE_Int)<-WE_Int[1,]
WE_Int<-WE_Int[-1,]


data_numWE_Int<-as.data.frame(apply(WE_Int,2, as.numeric))
sapply(data_numWE_Int, class
)
```

```
## 1892-93-286-180 1892-93-307-197 1894-95-142-277 1898-99-232-476     1879-73-121
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric"
##       1893-94-8      1893-94-86    1899-288-512      1886-87-99  1890-91-21-129
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric"
## 1892-93-285-179      1906-07-37    1902-144-594    ULAC_131_199    ULAC_186_201
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric"
##     ULAC_201_12
##       "numeric"
```

```
View(data_numWE_Int
)
```

```
Estim.NA <- estim_ncpPCA(data_numWE_Int, ncp.min = 0, ncp.max = 10, method =
                           "Regularized", scale = TRUE, method.cv = "gcv",
                         nbsim = 100, pNA = 0.05, threshold=1e-4, verbose =
                           TRUE)
Estim.NA
```

```
## $ncp
## [1] 1
##
## $criterion
##        0        1        2        3        4        5        6        7
##  129.3730  118.4657  138.0043  133.3506  160.1878  186.0831  230.8818  345.5044
##        8        9       10
##  537.5612 1032.4624 5573.5584
```

```
Impute.PCA <- imputePCA(data_numWE_Int, ncp = 2) #based on the results of
Estim.NA

## $ncp
## [1] 1
##
## $criterion
##        0        1        2        3        4        5        6        7
##  129.3730  118.4657  138.0043  133.3506  160.1878  186.0831  230.8818  345.5044
##        8        9       10
##  537.5612 1032.4624 5573.5584

WE.complete.sample_HS_Int <- Impute.PCA$completeObs
WE.complete.sample_HS_Int <- as.data.frame(WE.complete.sample_HS_Int)

write.table(WE.complete.sample_HS_Int, "~/Desktop/WE_NA_Test_Int.txt")
("~/Desktop/WE_NA_Test_Int.txt")

## [1] "~/Desktop/WE_NA_Test_Int.txt"

SA_NA_Imp_Int<-read.table("~/Desktop/SA_NA_Test_Int.txt")
SA_NA_Imp_Int<-t(SA_NA_Imp_Int)

GI_NA_Imp_Int<-read.table("~/Desktop/GI_NA_Test_Int.txt")

GI_NA_Imp_Int<-t(GI_NA_Imp_Int)

WE_NA_Imp_Int<-read.table("~/Desktop/WE_NA_Test_Int.txt")


WE_NA_Imp_Int<-t(WE_NA_Imp_Int)


NA_All_Int<-rbind(GI_NA_Imp_Int,SA_NA_Imp_Int, WE_NA_Imp_Int)

NA_All_Int<-as.data.frame(apply(NA_All_Int,2, as.numeric))
sapply(NA_All_Int, class
)

##        1        2        3        4        5        6        7        8
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##        9       10       11       12       13       14       15       16
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       17       18       19       20       21       22       23       24
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       25       26       27       28       29       30       31       32
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       33       34       35       36       37       38       39       40
```

```
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       41       42       43       44       45       46       47       48
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       49       50       51       52       53       54       55       56
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       57       58       59       60       61       62       63       64
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       65       66       67       68       69       70       71       72
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       73       74       75       76       77       78       79       80
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       81       82       83       84       85       86       87       88
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       89       90       91       92       93       94       95       96
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##       97       98
## "numeric" "numeric"

View(NA_All_Int
)



Population=c("GI","GI","GI","GI","GI","GI","GI","GI","GI","GI","GI","GI","GI",
             "GI","GI","GI","SA","SA","SA","SA","SA","SA","SA","SA","SA","SA",
             "SA","SA","SA","SA","SA","SA","WE","WE","WE","WE","WE","WE","WE",
             "WE","WE","WE","WE","WE","WE","WE",
             "WE","WE")
NA_All_Int<-cbind(NA_All_Int,Population)

Age=c("1","1","1","1","2","2","2","2","3","3","3","3","4","4","4","4","1","1",
      "1","1","2","2","2","2","3","3","3","3","4","4","4","4","1","1","1","1",
      "2","2","2","2","3","3","3","3","4","4","4","4")
NA_All_Int<-cbind(NA_All_Int,Age)
Data_all_Int<-NA_All_Int
BM_data_Int <- as.matrix(Data_all_Int[, 1:98])
Population <- Data_all_Int$Population
Age <- Data_all_Int$Age

Man_pop_Int <- manova(BM_data_Int ~ Population*Age, data = NA_All_Int)
summary.aov(Man_pop_Int)

##  Response 1 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  16.62   8.310  0.3252 0.7245
## Age           3 171.51  57.168  2.2373 0.1006
## Population:Age  6 160.61  26.768  1.0476 0.4115
```

```
## Residuals      36 919.89  25.553
##
##  Response 2 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2    98.1   49.06  0.1335 0.8754
## Age           3   705.9  235.31  0.6405 0.5939
## Population:Age  6  2428.5  404.75  1.1017 0.3804
## Residuals     36 13225.5  367.37
##
##  Response 3 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2     2.2    1.09  0.0025 0.9975
## Age           3  2366.4  788.81  1.8149 0.1619
## Population:Age  6  4599.7  766.62  1.7639 0.1345
## Residuals     36 15646.4  434.62
##
##  Response 4 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   331.8  165.90  0.4107 0.6662
## Age           3  2470.6  823.53  2.0390 0.1257
## Population:Age  6   995.8  165.97  0.4109 0.8669
## Residuals     36 14540.2  403.89
##
##  Response 5 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2   816.5  408.27  1.2135 0.30902
## Age           3  3178.7 1059.57  3.1493 0.03669 *
## Population:Age  6  1905.8  317.63  0.9441 0.47613
## Residuals     36 12112.0  336.45
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 6 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2   28.0  13.998  0.1429 0.86735
## Age           3  697.2 232.395  2.3720 0.08651 .
## Population:Age  6  293.9  48.982  0.5000 0.80413
## Residuals     36 3527.0  97.973
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 7 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   96.91  48.455  0.9151 0.4096
## Age           3   59.65  19.882  0.3755 0.7712
## Population:Age  6  223.00  37.167  0.7019 0.6499
```

```
## Residuals      36 1906.22  52.950
##
##  Response 8 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  130.67  65.335  1.4895 0.2390
## Age             3   69.96  23.318  0.5316 0.6635
## Population:Age  6  431.24  71.874  1.6386 0.1649
## Residuals      36 1579.07  43.863
##
##  Response 9 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   52.24  26.121  0.8243 0.4467
## Age             3   92.70  30.901  0.9751 0.4152
## Population:Age  6  440.17  73.362  2.3150 0.0542 .
## Residuals      36 1140.85  31.690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 10 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  51.03  25.517  1.3585 0.2699
## Age             3  18.45   6.148  0.3273 0.8056
## Population:Age  6 198.45  33.075  1.7609 0.1352
## Residuals      36 676.21  18.784
##
##  Response 11 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   86.84  43.419  1.4701 0.2434
## Age             3   44.11  14.703  0.4978 0.6861
## Population:Age  6  263.76  43.960  1.4884 0.2099
## Residuals      36 1063.27  29.535
##
##  Response 12 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  36.551 18.2754  2.3691 0.10800
## Age             3  29.640  9.8800  1.2808 0.29568
## Population:Age  6 121.111 20.1852  2.6166 0.03294 *
## Residuals      36 277.710  7.7142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 13 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  21.985 10.9926  1.7117 0.1949
## Age             3   3.286  1.0954  0.1706 0.9156
## Population:Age  6  28.753  4.7921  0.7462 0.6163
```

```
## Residuals      36 231.195  6.4221
##
##  Response 14 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2  26.270 13.1350  1.8838 0.1667
## Age           3   4.126  1.3754  0.1973 0.8976
## Population:Age  6  38.615  6.4358  0.9230 0.4901
## Residuals     36 251.018  6.9727
##
##  Response 15 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  261.6  130.82  0.6071 0.5504
## Age           3  888.4  296.14  1.3743 0.2662
## Population:Age  6 1488.1  248.02  1.1510 0.3537
## Residuals     36 7757.4  215.48
##
##  Response 16 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   270.0  135.01  0.3446 0.7108
## Age           3  1449.1  483.03  1.2329 0.3119
## Population:Age  6  1308.8  218.13  0.5568 0.7615
## Residuals     36 14103.8  391.77
##
##  Response 17 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2    64.1   32.06  0.0667 0.9356
## Age           3  2816.2  938.74  1.9524 0.1386
## Population:Age  6  1201.3  200.21  0.4164 0.8632
## Residuals     36 17309.0  480.81
##
##  Response 18 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   180.2   90.09  0.2599 0.7726
## Age           3  1893.5  631.17  1.8208 0.1608
## Population:Age  6  1039.7  173.28  0.4999 0.8042
## Residuals     36 12479.0  346.64
##
##  Response 19 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  175.2   87.59  0.3598 0.7003
## Age           3 1509.1  503.03  2.0662 0.1219
## Population:Age  6 1219.2  203.21  0.8347 0.5513
## Residuals     36 8764.6  243.46
##
##  Response 20 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
```

```
## Population      2   56.72  28.360  0.3643 0.6972
## Age             3  270.42  90.140  1.1580 0.3392
## Population:Age  6  190.73  31.789  0.4084 0.8686
## Residuals      36 2802.35  77.843
##
##  Response 21 :
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  55.57  27.785  1.0246 0.36918
## Age             3 187.09  62.364  2.2997 0.09380 .
## Population:Age  6 387.73  64.621  2.3829 0.04844 *
## Residuals      36 976.27  27.119
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 22 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  667.4  333.68  1.3300 0.2771
## Age             3  509.6  169.87  0.6771 0.5718
## Population:Age  6 1342.9  223.82  0.8922 0.5110
## Residuals      36 9031.6  250.88
##
##  Response 23 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  269.5  134.77  0.5630 0.5744
## Age             3 1128.1  376.02  1.5708 0.2133
## Population:Age  6 1493.5  248.92  1.0398 0.4161
## Residuals      36 8617.9  239.39
##
##  Response 24 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  215.3  107.63  0.5250 0.5960
## Age             3  882.0  294.00  1.4342 0.2488
## Population:Age  6 1404.6  234.10  1.1420 0.3585
## Residuals      36 7379.7  204.99
##
##  Response 25 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   17.61   8.807  0.1793 0.8366
## Age             3  239.70  79.901  1.6267 0.2003
## Population:Age  6  268.68  44.780  0.9117 0.4977
## Residuals      36 1768.29  49.119
##
##  Response 26 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2 105.30  52.650  1.9040 0.1637
## Age             3  25.92   8.640  0.3125 0.8162
```

```
## Population:Age  6 241.24  40.207  1.4540 0.2217
## Residuals      36 995.48  27.652
##
##  Response 27 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  23.47 11.7367  1.3170 0.2805
## Age             3   3.98  1.3275  0.1490 0.9297
## Population:Age  6  60.36 10.0598  1.1288 0.3655
## Residuals      36 320.83  8.9118
##
##  Response 28 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   18.46   9.230  0.1710 0.8435
## Age             3  163.97  54.657  1.0126 0.3984
## Population:Age  6  358.78  59.796  1.1078 0.3770
## Residuals      36 1943.15  53.976
##
##  Response 29 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   72.58  36.290  0.6443 0.5310
## Age             3  256.16  85.388  1.5159 0.2269
## Population:Age  6  202.65  33.775  0.5996 0.7287
## Residuals      36 2027.78  56.327
##
##  Response 30 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  371.0  185.49  1.6655 0.2033
## Age             3  339.5  113.16  1.0160 0.3969
## Population:Age  6  687.9  114.66  1.0295 0.4223
## Residuals      36 4009.4  111.37
##
##  Response 31 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2    21.6   10.79  0.0331 0.96746
## Age             3  2635.1  878.37  2.6956 0.06036 .
## Population:Age  6  1568.5  261.42  0.8023 0.57471
## Residuals      36 11730.8  325.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 32 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   60.0   29.98  0.1100 0.8962
## Age             3 1685.4  561.81  2.0608 0.1227
## Population:Age  6 1795.8  299.30  1.0979 0.3825
## Residuals      36 9814.2  272.62
```

```
##
##  Response 33 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   768.4  384.19  1.1279 0.3349
## Age           3  1188.4  396.12  1.1629 0.3373
## Population:Age  6  3243.6  540.59  1.5871 0.1792
## Residuals    36 12262.2  340.62
##
##  Response 34 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2  145.16  72.580  1.1064 0.3417
## Age           3  311.50 103.834  1.5829 0.2104
## Population:Age  6  731.09 121.848  1.8575 0.1154
## Residuals    36 2361.55  65.599
##
##  Response 35 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2  134.28  67.138  0.8414 0.4394
## Age           3  383.42 127.806  1.6016 0.2060
## Population:Age  6  573.16  95.527  1.1971 0.3301
## Residuals    36 2872.69  79.797
##
##  Response 36 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   270.8  135.40  0.4393 0.6479
## Age           3   809.2  269.73  0.8751 0.4630
## Population:Age  6  2461.1  410.18  1.3308 0.2690
## Residuals    36 11096.2  308.23
##
##  Response 37 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2   113.7   56.87  0.1918 0.8263
## Age           3  1077.7  359.24  1.2117 0.3195
## Population:Age  6  1493.2  248.87  0.8394 0.5479
## Residuals    36 10673.4  296.48
##
##  Response 38 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  710.1  355.05  2.9608 0.0645 .
## Age           3  637.2  212.40  1.7712 0.1701
## Population:Age  6  830.6  138.44  1.1544 0.3519
## Residuals    36 4317.0  119.92
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 39 :
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  165.82  82.909  1.6971 0.1975
## Age             3   62.41  20.803  0.4258 0.7357
## Population:Age  6   42.36   7.061  0.1445 0.9890
## Residuals      36 1758.69  48.852
##
##  Response 40 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   51.34  25.671  0.8622 0.4308
## Age             3   41.17  13.724  0.4610 0.7113
## Population:Age  6  117.76  19.627  0.6592 0.6827
## Residuals      36 1071.84  29.773
##
##  Response 41 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  234.78 117.388  1.6590 0.2045
## Age             3  287.20  95.734  1.3530 0.2727
## Population:Age  6  741.85 123.641  1.7474 0.1382
## Residuals      36 2547.28  70.758
##
##  Response 42 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  183.64  91.819  1.5719 0.2216
## Age             3   75.22  25.072  0.4292 0.7333
## Population:Age  6  369.78  61.630  1.0551 0.4071
## Residuals      36 2102.86  58.413
##
##  Response 43 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  292.9  146.44  0.6124 0.5476
## Age             3 1248.2  416.08  1.7399 0.1762
## Population:Age  6 1430.1  238.35  0.9967 0.4425
## Residuals      36 8609.2  239.14
##
##  Response 44 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  171.39  85.695  2.4508 0.1005
## Age             3  145.60  48.533  1.3880 0.2621
## Population:Age  6  367.81  61.301  1.7532 0.1369
## Residuals      36 1258.78  34.966
##
##  Response 45 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  294.69 147.346  3.8409 0.03076 *
## Age             3   93.43  31.143  0.8118 0.49573
## Population:Age  6  421.91  70.319  1.8330 0.12013
```

```
## Residuals      36 1381.04  38.362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 46 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2   28.8  14.398  0.1368 0.8726
## Age           3  238.7  79.551  0.7559 0.5262
## Population:Age  6  982.8 163.804  1.5565 0.1883
## Residuals     36 3788.5 105.237
##
##  Response 47 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population    2  571.79 285.893  3.7120 0.03422 *
## Age           3   67.27  22.423  0.2911 0.83150
## Population:Age  6  784.23 130.706  1.6971 0.15000
## Residuals     36 2772.65  77.018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 48 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2   24.7  12.372  0.1206 0.8868
## Age           3   20.5   6.827  0.0665 0.9773
## Population:Age  6  665.0 110.831  1.0802 0.3925
## Residuals     36 3693.7 102.602
##
##  Response 49 :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Population    2  149.9   74.93  0.5276 0.59451
## Age           3  353.0  117.65  0.8284 0.48698
## Population:Age  6 2675.1  445.85  3.1392 0.01404 *
## Residuals     36 5112.9  142.03
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 50 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  500.5  250.24  1.2067  0.311
## Age           3  545.7  181.90  0.8772  0.462
## Population:Age  6 2281.5  380.26  1.8336  0.120
## Residuals     36 7465.6  207.38
##
##  Response 51 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  497.6  248.78  1.5086 0.2349
```

```
## Age             3  339.5  113.18  0.6864 0.5663
## Population:Age  6  531.9   88.65  0.5376 0.7760
## Residuals      36 5936.6  164.91
##
##  Response 52 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   85.7  42.849  0.3765 0.6890
## Age             3  415.0 138.343  1.2154 0.3181
## Population:Age  6  642.4 107.074  0.9407 0.4783
## Residuals      36 4097.5 113.821
##
##  Response 53 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  127.09  63.546  0.8792 0.4238
## Age             3  238.40  79.468  1.0995 0.3619
## Population:Age  6  329.92  54.987  0.7608 0.6053
## Residuals      36 2601.89  72.275
##
##  Response 54 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  106.66  53.330  1.2130 0.30918
## Age             3   76.16  25.386  0.5774 0.63361
## Population:Age  6  590.75  98.458  2.2394 0.06142 .
## Residuals      36 1582.79  43.966
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 55 :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  103.69  51.844  1.2290 0.30456
## Age             3  131.48  43.828  1.0390 0.38701
## Population:Age  6  700.72 116.787  2.7686 0.02566 *
## Residuals      36 1518.60  42.183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 56 :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   78.22  39.108  1.3208 0.2795
## Age             3  119.83  39.943  1.3490 0.2739
## Population:Age  6  272.78  45.463  1.5354 0.1947
## Residuals      36 1065.93  29.609
##
##  Response 57 :
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  800.3  400.16  1.8543 0.17121
```

```
## Age            3 1605.0  535.01  2.4792 0.07676 .
## Population:Age  6 2389.2  398.20  1.8452 0.11775
## Residuals      36 7768.9  215.80
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 58 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  198.14  99.068  1.4819 0.24074
## Age            3  570.39 190.131  2.8440 0.05124 .
## Population:Age  6  945.47 157.578  2.3571 0.05055 .
## Residuals      36 2406.71  66.853
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 59 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  96.60  48.302  2.3783 0.1071
## Age            3  54.53  18.177  0.8950 0.4531
## Population:Age  6  97.11  16.184  0.7969 0.5787
## Residuals      36 731.16  20.310
##
##  Response 60 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   98.87  49.434  1.0955 0.3453
## Age            3   46.92  15.640  0.3466 0.7918
## Population:Age  6  106.31  17.719  0.3927 0.8788
## Residuals      36 1624.45  45.124
##
##  Response 61 :
##              Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  398.32 199.159  4.7598 0.01465 *
## Age            3  219.56  73.186  1.7491 0.17439
## Population:Age  6  400.27  66.711  1.5944 0.17715
## Residuals      36 1506.32  41.842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 62 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  368.5 184.233  1.4658 0.2443
## Age            3   80.9  26.967  0.2146 0.8857
## Population:Age  6  411.7  68.613  0.5459 0.7697
## Residuals      36 4524.6 125.684
##
##  Response 63 :
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   34.57  17.286  0.2777 0.7591
## Age             3  138.75  46.251  0.7431 0.5334
## Population:Age  6  199.26  33.210  0.5336 0.7791
## Residuals      36 2240.77  62.244
##
##  Response 64 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   3.85  1.9232  0.0919 0.9124
## Age             3  49.00 16.3349  0.7806 0.5126
## Population:Age  6 119.75 19.9585  0.9538 0.4698
## Residuals      36 753.34 20.9262
##
##  Response 65 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  18.18  9.0919  0.4184 0.6613
## Age             3  90.89 30.2956  1.3941 0.2603
## Population:Age  6 159.59 26.5991  1.2240 0.3169
## Residuals      36 782.30 21.7306
##
##  Response 66 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  19.19   9.595  0.4683 0.6299
## Age             3 105.43  35.144  1.7151 0.1812
## Population:Age  6 180.46  30.077  1.4678 0.2169
## Residuals      36 737.67  20.491
##
##  Response 67 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  15.73  7.8665  0.3888 0.6807
## Age             3  50.15 16.7170  0.8262 0.4881
## Population:Age  6 100.34 16.7236  0.8265 0.5571
## Residuals      36 728.44 20.2345
##
##  Response 68 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  25.30  12.650  0.6665 0.5197
## Age             3  39.75  13.250  0.6981 0.5593
## Population:Age  6  99.56  16.593  0.8742 0.5234
## Residuals      36 683.29  18.980
##
##  Response 69 :
##                Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  10.37  5.1841  0.2485 0.7813
## Age             3  41.63 13.8765  0.6653 0.5789
## Population:Age  6 138.48 23.0803  1.1065 0.3777
```

```
## Residuals      36 750.92 20.8588
##
##  Response 70 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2    1.29   0.645  0.0201 0.9801
## Age           3   39.81  13.269  0.4143 0.7438
## Population:Age  6  100.38  16.730  0.5223 0.7875
## Residuals     36 1153.02  32.028
##
##  Response 71 :
##              Df  Sum Sq Mean Sq F value   Pr(>F)
## Population    2  219.57  109.79  1.6355 0.208996
## Age           3 1201.37  400.46  5.9657 0.002071 **
## Population:Age  6  661.99  110.33  1.6436 0.163592
## Residuals     36 2416.56   67.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 72 :
##              Df  Sum Sq Mean Sq F value   Pr(>F)
## Population    2   36.46  18.231  0.4568 0.636947
## Age           3  623.87 207.958  5.2103 0.004314 **
## Population:Age  6  354.03  59.004  1.4783 0.213323
## Residuals     36 1436.87  39.913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 73 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  633.5  316.74  2.1403 0.1323
## Age           3  665.2  221.74  1.4984 0.2315
## Population:Age  6  731.3  121.89  0.8236 0.5592
## Residuals     36 5327.5  147.99
##
##  Response 74 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  340.0 169.992  1.7155 0.1943
## Age           3  574.5 191.492  1.9324 0.1418
## Population:Age  6  490.3  81.725  0.8247 0.5584
## Residuals     36 3567.4  99.094
##
##  Response 75 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  219.4 109.722  0.8306 0.4439
## Age           3   81.7  27.241  0.2062 0.8914
## Population:Age  6  477.3  79.558  0.6023 0.7266
```

```
## Residuals      36 4755.3 132.093
##
##  Response 76 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population    2  134.55  67.276  1.1163 0.3386
## Age           3   72.83  24.277  0.4028 0.7518
## Population:Age  6  109.73  18.289  0.3035 0.9310
## Residuals     36 2169.69  60.269
##
##  Response 77 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  16.93  8.4637  0.3779 0.6880
## Age           3  39.87 13.2915  0.5935 0.6233
## Population:Age  6 152.53 25.4221  1.1351 0.3621
## Residuals     36 806.26 22.3961
##
##  Response 78 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  19.24   9.620  0.3726 0.6916
## Age           3 107.27  35.757  1.3848 0.2631
## Population:Age  6 115.37  19.228  0.7447 0.6175
## Residuals     36 929.54  25.821
##
##  Response 79 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  24.40  12.202  0.4766 0.6248
## Age           3 109.31  36.435  1.4231 0.2520
## Population:Age  6 116.68  19.447  0.7596 0.6063
## Residuals     36 921.67  25.602
##
##  Response 80 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2  39.16  19.580  0.7726 0.4693
## Age           3  81.72  27.240  1.0748 0.3720
## Population:Age  6 134.17  22.362  0.8824 0.5177
## Residuals     36 912.36  25.343
##
##  Response 81 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population    2   5.83   2.916  0.1218 0.8857
## Age           3 103.87  34.623  1.4459 0.2456
## Population:Age  6 140.23  23.372  0.9760 0.4555
## Residuals     36 862.07  23.946
##
##  Response 82 :
##              Df Sum Sq Mean Sq F value Pr(>F)
```

```
## Population      2   1.07  0.5328  0.0238 0.9765
## Age             3  80.14 26.7143  1.1909 0.3269
## Population:Age  6 125.59 20.9308  0.9331 0.4834
## Residuals      36 807.53 22.4314
##
##  Response 83 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   21.51  10.755  0.2713 0.7640
## Age             3  132.32  44.107  1.1125 0.3568
## Population:Age  6  188.78  31.463  0.7936 0.5811
## Residuals      36 1427.30  39.647
##
##  Response 84 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   14.02   7.009  0.1380 0.8716
## Age             3  231.26  77.086  1.5176 0.2265
## Population:Age  6  295.17  49.195  0.9685 0.4603
## Residuals      36 1828.68  50.797
##
##  Response 85 :
##                 Df  Sum Sq Mean Sq F value   Pr(>F)
## Population      2   62.46  31.230  0.5596 0.576311
## Age             3  744.23 248.077  4.4455 0.009322 **
## Population:Age  6  240.44  40.074  0.7181 0.637531
## Residuals      36 2008.95  55.804
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 86 :
##                 Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   0.06  0.0321  0.0014 0.9987
## Age             3  61.42 20.4723  0.8617 0.4698
## Population:Age  6 128.48 21.4129  0.9013 0.5047
## Residuals      36 855.28 23.7577
##
##  Response 87 :
##                 Df  Sum Sq Mean Sq F value  Pr(>F)
## Population      2  212.49 106.247  2.0747 0.14035
## Age             3  478.43 159.476  3.1142 0.03812 *
## Population:Age  6  514.83  85.806  1.6756 0.15534
## Residuals      36 1843.55  51.210
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 88 :
##                 Df  Sum Sq Mean Sq F value Pr(>F)
```

```
## Population      2   75.32  37.662  0.5544 0.5792
## Age             3   56.75  18.917  0.2785 0.8406
## Population:Age  6  549.14  91.524  1.3473 0.2622
## Residuals      36 2445.49  67.930
##
##  Response 89 :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   92.4  46.182  0.4711 0.6281
## Age             3   12.2   4.072  0.0415 0.9885
## Population:Age  6  719.9 119.979  1.2240 0.3169
## Residuals      36 3528.7  98.020
##
##  Response 90 :
##             Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2  121.08  60.538  1.9785 0.1530
## Age             3  106.93  35.642  1.1649 0.3366
## Population:Age  6   46.85   7.808  0.2552 0.9539
## Residuals      36 1101.52  30.598
##
##  Response 91 :
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Population      2  54.49  27.243  1.1500 0.32799
## Age             3  72.05  24.016  1.0138 0.39791
## Population:Age  6 277.50  46.250  1.9524 0.09872 .
## Residuals      36 852.80  23.689
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 92 :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  14.68   7.341  0.3064 0.7380
## Age             3 143.39  47.796  1.9947 0.1322
## Population:Age  6 196.07  32.678  1.3638 0.2555
## Residuals      36 862.61  23.961
##
##  Response 93 :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  14.68   7.341  0.3064 0.7380
## Age             3 143.39  47.796  1.9947 0.1322
## Population:Age  6 196.07  32.678  1.3638 0.2555
## Residuals      36 862.61  23.961
##
##  Response 94 :
##             Df Sum Sq Mean Sq F value Pr(>F)
## Population      2  14.68   7.341  0.3064 0.7380
## Age             3 143.39  47.796  1.9947 0.1322
```

```
## Population:Age  6 196.07  32.678  1.3638 0.2555
## Residuals      36 862.61  23.961
##
##  Response 95 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   0.89   0.445  0.0176 0.9825
## Age             3 104.65  34.884  1.3806 0.2643
## Population:Age  6 147.75  24.624  0.9746 0.4564
## Residuals      36 909.58  25.266
##
##  Response 96 :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Population      2   0.89   0.445  0.0176 0.9825
## Age             3 104.65  34.884  1.3806 0.2643
## Population:Age  6 147.75  24.624  0.9746 0.4564
## Residuals      36 909.58  25.266
##
##  Response 97 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   56.31  28.153  0.7410 0.4837
## Age             3  148.24  49.415  1.3007 0.2891
## Population:Age  6  244.08  40.681  1.0708 0.3979
## Residuals      36 1367.68  37.991
##
##  Response 98 :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## Population      2   11.09   5.545  0.1397 0.8700
## Age             3  131.67  43.890  1.1061 0.3593
## Population:Age  6  235.83  39.306  0.9906 0.4463
## Residuals      36 1428.46  39.679

 df_AG1<-subset(Data_all_Int, Data_all_Int$Age=="1" )

Data_Test<-cbind(Data_all_Int[,1:98],Data_all)

 df_AG1<-subset(Data_Test, Data_Test$Age=="1" )


#Hypothesis 2
#population, age group not pooled
Perman_AG1 <- adonis2(df_AG1[,1:196] ~ df_AG1$Population, data = df_AG1) #here

## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##               in method "bray"
```

```
#you need a dataframe with the percentages and a column with the population info
Perman_AG1

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = df_AG1[, 1:196] ~ df_AG1$Population, data = df_AG1)
##                  Df SumOfSqs     R2      F Pr(>F)
## df_AG1$Population  2   0.6528 0.2014 1.1349  0.237
## Residual           9   2.5885 0.7986
## Total             11   3.2414 1.0000

Data_Test<-cbind(Data_all_Int[,1:98],Data_all)

df_AG2<-subset(Data_Test, Data_Test$Age=="2" )


Perman_AG2 <- adonis2(df_AG2[,1:196] ~ df_AG2$Population, data = df_AG2) #here

## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##                  in method "bray"

#you need a dataframe with the percentages and a column with the population info
Perman_AG2

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = df_AG2[, 1:196] ~ df_AG2$Population, data = df_AG2)
##                  Df SumOfSqs      R2      F Pr(>F)
## df_AG2$Population  2  0.56944 0.23478 1.3807  0.065 .
## Residual           9  1.85597 0.76522
## Total             11  2.42541 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

df_AG3<-subset(Data_Test, Data_Test$Age=="3" )


Perman_AG3 <- adonis2(df_AG3[,1:196] ~ df_AG3$Population, data = df_AG3) #here
```

```
## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##              in method "bray"

#you need a dataframe with the percentages and a column with the population info
Perman_AG3

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = df_AG3[, 1:196] ~ df_AG3$Population, data = df_AG3)
##                Df SumOfSqs      R2      F Pr(>F)
## df_AG3$Population  2  0.63098 0.20588 1.1667  0.217
## Residual           9  2.43378 0.79412
## Total             11  3.06475 1.00000

df_AG4<-subset(Data_Test, Data_Test$Age=="4" )


Perman_AG4 <- adonis2(df_AG4[,1:196] ~ df_AG4$Population, data = df_AG4) #here

## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##              in method "bray"

#you need a dataframe with the percentages and a column with the population info
Perman_AG4

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = df_AG4[, 1:196] ~ df_AG4$Population, data = df_AG4)
##                Df SumOfSqs      R2      F Pr(>F)
## df_AG4$Population  2   0.6492 0.18251 1.0047  0.465
## Residual           9   2.9077 0.81749
## Total             11   3.5568 1.00000

#Population, age pooled
Perman_All <- adonis2(Data_Test[,1:196] ~ Data_Test$Population, data = Data_Test
                ) #here you need a dataframe with the percentages and a

## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##              in method "bray"
```

```
#column with the population info
Perman_All

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = Data_Test[, 1:196] ~ Data_Test$Population, data = Data_Test)
##                    Df SumOfSqs      R2      F Pr(>F)
## Data_Test$Population  2   0.7629 0.05605 1.3359  0.044 *
## Residual             45  12.8487 0.94395
## Total                47  13.6116 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ALL_post_test<-pairwise.adonis(Data_Test[,1:196],factor= Data_Test$Population)

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                in method "bray"

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                in method "bray"

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                in method "bray"

ALL_post_test

##      pairs Df SumsOfSqs  F.Model        R2 p.value p.adjusted sig
## 1 GI vs SA  1 0.3910637 1.330242 0.04245872   0.124      0.372
## 2 GI vs WE  1 0.4311409 1.495024 0.04746857   0.042      0.126
## 3 SA vs WE  1 0.3221206 1.174703 0.03768130   0.217      0.651

#Hypothesis 1



WE_Test<-subset(Data_Test, Data_Test$Population=="WE")


Perman_WE <- adonis2(WE_Test[,1:196] ~ WE_Test$Age, data = WE_Test) #here you

## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##                in method "bray"
```

```
#need a dataframe with the percentages and a column with the population info
Perman_WE

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = WE_Test[, 1:196] ~ WE_Test$Age, data = WE_Test)
##             Df SumOfSqs      R2      F Pr(>F)
## WE_Test$Age  3   1.1275 0.27984 1.5543  0.003 **
## Residual    12   2.9017 0.72016
## Total       15   4.0293 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#posthoc test

WE_post_test<-pairwise.adonis(WE_Test[,1:196],factor= WE_Test$Age)

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                 in method "bray"

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                 in method "bray"

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                 in method "bray"

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                 in method "bray"

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                 in method "bray"

## Warning in vegdist(x[factors %in% c(co[1, elem], co[2, elem]), ], method = sim.method): res
ults may be meaningless because data have negative entries
##                 in method "bray"

WE_post_test

##    pairs Df SumsOfSqs  F.Model        R2 p.value p.adjusted sig
## 1 1 vs 2  1 0.2179492 1.245745 0.1719278   0.131      0.786
```

```
## 2 1 vs 3  1 0.2353277 1.043309 0.1481277   0.386      1.000
## 3 1 vs 4  1 0.5517062 1.743226 0.2251292   0.036      0.216
## 4 2 vs 3  1 0.1823277 1.090882 0.1538429   0.369      1.000
## 5 2 vs 4  1 0.5332572 2.066369 0.2561709   0.034      0.204
## 6 3 vs 4  1 0.5345180 1.731688 0.2239728   0.029      0.174
```

```
GI_Test<-subset(Data_Test, Data_Test$Population=="GI" )
```

```
Perman_GI <- adonis2(GI_Test[,1:196] ~ GI_Test$Age, data = GI_Test)
```

```
## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##                in method "bray"
```

```
#here you need a dataframe with the percentages and a column with the population
#info
Perman_GI
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = GI_Test[, 1:196] ~ GI_Test$Age, data = GI_Test)
##            Df SumOfSqs      R2      F Pr(>F)
## GI_Test$Age  3   1.0233 0.22139 1.1374  0.187
## Residual    12   3.5989 0.77861
## Total       15   4.6222 1.00000
```

```
SA_Test<-subset(Data_Test, Data_Test$Population=="SA" )
```

```
Perman_SA <- adonis2(SA_Test[,1:196] ~ SA_Test$Age, data = SA_Test)
```

```
## Warning in vegdist(as.matrix(lhs), method = method, ...): results may be meaningless because
data have negative entries
##                in method "bray"
```

```
#here you need a dataframe with the percentages and a column with the population
#info
Perman_SA
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = SA_Test[, 1:196] ~ SA_Test$Age, data = SA_Test)
```

```
##             Df SumOfSqs      R2      F Pr(>F)
## SA_Test$Age  3   0.9118 0.21725 1.1102  0.288
## Residual    12   3.2853 0.78275
## Total       15   4.1971 1.00000
```

REFERENCES

Adobe Inc. (2019). *Adobe Illustrator*. Retrieved from https://adobe.com/products/illustrator

Adobe Inc. (2019). *Adobe Photoshop*. Retrieved from https://www.adobe.com/products/photoshop.html

Allen, M. R., & Burr, D. B. (2019). Bone Growth, Modeling, and Remodeling. In Basic and Applied Bone Biology (pp. 85-100). https://doi.org/10.1016/b978-0-12-813259-3.00005-1

Anderson, M.J. (2001), A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, 26: 32-46. https://doi.org/10.1111/j.1442-9993.2001.01070.pp.x

Anderson PSL, Renaud S, & Rayfield EJ. (2014). Adaptive plasticity in the mouse mandible. *BMC EVOLUTIONARY BIOLOGY*, (1):85–85.

Aporta, N. B., Martinez-Maza, C., Gonzalez, P., & Bernal, V. (2014). Bone modeling patterns and morphometric craniofacial variation in individuals from two prehistoric human populations from Argentina. *Anatomical Record.* Hoboken, N.J.: 2007, 297(10), 1829-1838.

AlQahtani, S. J., Hector, M. P., & Liversidge, H. M. (2010). Brief communication: The London atlas of human tooth development and eruption. American Journal of Physical Anthropology, 142(3), 481–490.

Arbizu M.P. & S. Monteux, J. (2019). pairwiseAdonisR package version 0.4.

Baab, K. L., Copes, L. E., Ward, D. L., Wells, N. & Grine, F. E. Using modern human cortical bone distribution to test the systemic robusticity hypothesis. *J Hum Evol* **119**, 64-82, doi:10.1016/j.jhevol.2018.03.003 (2018).

Bastir, M., Rosas, A., & O'Higgins, P. (2006). Craniofacial levels and the morphological maturation of the human skull. *Journal of Anatomy*, 209(5), 637–654.

Bastir, M., O'Higgins, P., & Rosas, A. (2007). Facial ontogeny in Neanderthals and modern humans. *Proceedings of the Royal Society B: Biological Sciences*, *274*(1614), 1125-1132

Barak MM. (2020). Bone modeling or bone remodeling: That is the question. *American Journal of Physical Anthropology*, 72(2):153-155.

Barcikowski, R.S. (1981) *Journal of Educational Statistics.* 6(3): 267-285. https://www.jstor.org/stable/1164877

Bouvier, & Hylander, W. L. (1981). Effect of bone strain on cortical bone structure in macaques (Macaca mulatta). *Journal of Morphology (1931)*, 167(1), 1–12. https://doi.org/10.1002/jmor.1051670102

Brachetta Aporta, Martinez-Maza, C., Gonzalez, P. N., & Bernal, V. (2014). Bone Modeling Patterns and Morphometric Craniofacial Variation in Individuals From Two Prehistoric Human Populations From Argentina: Facial Bone Modeling and Morphometrics. *Anatomical Record (Hoboken, N.J. : 2007)*, *297*(10), 1829–1838. https://doi.org/10.1002/ar.22999

Brachetta-Aporta, Gonzalez, P. N., & Bernal, V. (2018). A quantitative approach for analysing bone modelling patterns from craniofacial surfaces in hominins. *Journal of Anatomy*, *232*(1), 3–14. https://doi.org/10.1111/joa.12716

Brachetta-Aporta, N., Gonzalez, P. N., & Bernal, V. (2019). Integrating data on bone modeling and morphological ontogenetic changes of the maxilla in modern humans. *Annals of Anatomy-Anatomischer Anzeiger*, *222*, 12-20.

Brachetta-Aporta, Gonzalez, P. N., & Bernal, V. (2019). Variation in facial bone growth remodeling in prehistoric populations from southern South America. *American Journal of Physical Anthropology*, *169*(3), 422–434. https://doi.org/10.1002/ajpa.23857

Brachetta-Aporta, N., & Toro-Ibacache, V. (2021). Differences in masticatory loads impact facial bone surface remodeling in an archaeological sample of South American individuals. *Journal of Archaeological Science: Reports*, 38, 103034.

Bromage. (1984). Interpretation of scanning electron microscopic images of abraded forming bone surfaces. American Journal of Physical Anthropology, 64(2), 161–178. https://doi.org/10.1002/ajpa.1330640210

Bromage T.G. 1985. Systematic inquiry in tests of negative/positive replica combinations for SEM. J*ournal of Microscopy (Oxford*), 137(2), 209-216.

Bromage. (1989). Ontogeny of the early hominid face. *Journal of Human Evolution*, 18(8), 751–773. https://doi.org/10.1016/0047-2484(89)90088-2

Butaric, L. N., Nicholas, C. L., Kravchuk, K. & Maddux, S. D. (2022) Ontogenetic variation in human nasal morphology. *Anat Rec (Hoboken)* 305, 1910-1937, doi:10.1002/ar.24760.

Cakirer, Hans, M. G., Graham, G., Aylor, J., Tishler, P. V., & Redline, S. (2001). The Relationship Between Craniofacial Morphology and Obstructive Sleep Apnea in Whites and in African-Americans. *American Journal of Respiratory and Critical Care Medicine*, 163(4), 947–950. https://doi.org/10.1164/ajrccm.163.4.2005136

Churchill, S.E. and Morris, A.G. (1998), Muscle marking morphology and labour intensity in prehistoric Khoisan foragers. Int. J. Osteoarchaeol., 8: 390-411. https://doi.org/10.1002/(SICI)1099-1212(1998090)8:5<390::AID-OA435>3.0.CO;2-N

Coquerelle, M., Prados-Frutos, J. C., Benazzi, S., Bookstein, F. L., Senck, S., Mitteroecker, P., & Weber, G. W. (2013a). Infant growth patterns of the mandible in modern humans: a closer exploration of the developmental interactions between the symphyseal bone, the teeth, and the suprahyoid and tongue muscle insertion sites. J Anat, 222(2), 178-192.

Coquerelle, Bookstein, F. L., Braga, J., Halazonetis, D. J., Weber, G. W., & Mitteroecker, P. (2011). Sexual dimorphism of the human mandible and its association with dental development. American Journal of Physical Anthropology, 145(2), 192–202. https://doi.org/10.1002/ajpa.21485

Coquerelle, Prados-Frutos, J. C., Rojo, R., Mitteroecker, P., & Bastir, M. (2013b). Short Faces, Big Tongues: Developmental Origin of the Human Chin. PloS One, 8(11), e81287–e81287. https://doi.org/10.1371/journal.pone.0081287

Corruccini, & Beecher, R. (1982). Occlusal variation related to soft diet in a nonhuman primate. *Science*, 218(4567), 74–76. https://doi.org/10.1126/science.7123221

Dennison, Healey, D., & Herbison, P. (2007). The Polynesian chin and mandible. Anthropologischer Anzeiger, 65(3), 241–252. https://doi.org/10.1127/anthranz/65/2007/241

DiGangi, & Moore, M. K. (2013). Research methods in human skeletal biology. Academic Press.

Enlow, D. H. (1962). A study of the post-Natal growth and remodeling of bone. American Journal of Anatomy, 110(2), 79–101.

Enlow, D.H. (1966). A comparative study of facial growth in Homo and Macaca: Am. J. Phys. xAnthrop. 24: 293–303. https://doi.org/10.1016/0002-9416(67)90216-3

Enlow, DH, & Bang S. (1965). Growth and remodeling of the human maxilla. Am J Orthod, 51, 446-464.

Enlow, & Moyers, R. E. (1971). Growth and architecture of the face. The Journal of the American Dental Association (1939), 82(4), 763–774.

Enlow PH, Hans MG, with 3 contributors. Essentials of facial growth: Philadelphia, PA, Saunders, 1996, 303 pages, illustrated, paperback. (1997). Journal of Oral and Maxillofacial Surgery, 55(7), 786–786. https://doi.org/10.1016/S0278-2391(97)90621-2

Evteev, Movsesian, A. A., & Grosheva, A. N. (2017). The association between mid-facial morphology and climate in northeast Europe differs from that in north Asia: Implications for understanding the morphology of Late Pleistocene Homo sapiens. Journal of Human Evolution, 107, 36–48. https://doi.org/10.1016/j.jhevol.2017.02.008

Feise, R.J. (2002). Do multiple outcome measures require p-value adjustment?. *BMC Med Res Methodol* 2:8. https://doi.org/10.1186/1471-2288-2-8

Fisher. (1954). *Statistical methods for research workers*.

Freidline, Gunz, P., & Hublin, J.-J. (2015). Ontogenetic and static allometry in the human face: Contrasting Khoisan and Inuit. American Journal of Physical Anthropology, 158(1), 116–131. https://doi.org/10.1002/ajpa.22759

Freidline, S. E., Martinez-Maza, C., Gunz, P., & Hublin, J. J. (2017). Exploring modern human facial growth at the micro-and macroscopic levels. In *Building bones: Bone formation and development in anthropology*, 104-127.

Frost, H. M. (1990). Skeletal structural adaptations to mechanical usage (SATMU): 1. Redefining Wolff's law: the bone modeling problem. *The Anatomical Record*, 226(4), 403-413.

Gamer M, Lemon J. (2019). _irr: Various Coefficients of Interrater Reliability and Agreement_. R package version 0.84.1, <https://CRAN.R-project.org/package=irr>.

German, & Crompton, A. . (1996). Ontogeny of Suckling Mechanisms in Opossums (Didelphis virginiana). *Brain, Behavior and Evolution*, 48(3), 157–164. https://doi.org/10.1159/000113194

Gomes, A. S., & Lima, E. M. (2006). Mandibular Growth during Adolescence. *The Angle Orthodontist*, *76*(5), 786-790. https://doi.org/10.1043/00033219(2006)076[0786:Mgda]2.0.Co;2

Gonzalez, R., & Griffin, D. (n.d.). *Lecture Notes #2*. Retrieved March 21, 2023, from http://www-personal.umich.edu/~gonzo/course/new.html#CITEGriffin&Gonzalez95exch

Hans M.G., Enlow, D. H., & Noachtar, R. (1995). Age-related differences in mandibular ramus growth: a histologic study. *The Angle Orthodontist*, 65(5), 335–340.

Hadjidakis, D. J., & Androulakis, II. (2006). Bone remodeling. Ann N Y Acad Sci, 1092, 385-396.

Hallgrímsson, B., Willmore, K., & Hall, B. K. (2002). Canalization, developmental stability, and morphological integration in primate limbs. *American Journal of Physical Anthropology, Suppl* 35, 131–158.

Heagerty P.J., DeLong E.R.(2022) for the NIH Health Care Systems Research Collaboratory Biostatistics and Study Design Core. Analysis Plan: Intraclass Correlation. In: *Rethinking Clinical Trials: A Living Textbook of Pragmatic Clinical Trials*. Bethesda, MD: NIH Pragmatic Trials Collaboratory. Available at: https://rethinkingclinicaltrials.org/chapters/design/analysis-plan-top/intraclasscorrelation/.

Helm, J. W., & German, R. Z. (1996). The epigenetic impact of weaning on craniofacial morphology during growth. *Journal of Experimental Zoology*, 276(4), 243-253.

Holton, N. E., Bonner, L. L., Scott, J. E., Marshall, S. D., Franciscus, R. G., & Southard, T. E. (2015). The ontogeny of the chin: an analysis of allometric and biomechanical scaling. *J Anat*, 226(6), 549-559. https://doi.org/10.1111/joa.12307

Hoover, & Williams, F. L. (2016). Variation in regional diet and mandibular morphology in prehistoric Japanese hunter–gatherer–fishers. *Quaternary International*, 405, 101–109.

Hubbe, M., Hanihara, T., & Harvati, K. (2009). Climate signatures in the morphological differentiation of worldwide modern human populations. *Anat Rec (Hoboken)*, 292(11), 1720-1733. https://doi.org/10.1002/ar.20976

Hylander. (1977). The adaptive significance of eskimo craniofacial morphology. Proquest dissertations publishing.

Iannotti, Gyimah, E. A., Reid, M., Chapnick, M., Cartmill, M. K., Lutter, C. K., Hilton, C., Gildner, T. E., & Quinn, E. A. (2022). Child dietary patterns in Homo sapiens evolution: A systematic review. *Evolution, Medicine, and Public Health*, *10*(1), 371–390.

Ichim, Swain, M., & Kieser, J. A. (2006). Mandibular Biomechanics and Development of the Human Chin. Journal of Dental Research, 85(7), 638–642. https://doi.org/10.1177/154405910608500711

Işeri, H., & Solow, B. (2000). Change in the width of the mandibular body from 6 to 23 years of age: an implant study. *European journal of orthodontics*, 22*(*3), 229-238.

Koo, T. K., & Li, M. Y. (2016). A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. *Journal of chiropractic medicine*, 15(2), 155–163. https://doi.org/10.1016/j.jcm.2016.02.012

Kranioti, Rosas, A., GarcÍA-Vargas, S., Estalrrich, A., Bastir, M., & PeÑA-MeliÁN, Án. (2009). Remodeling Patterns of Occipital Growth: A Preliminary Report. Anatomical Record (Hoboken, N.J. : 2007), 292(11), 1764–1770. https://doi.org/10.1002/ar.20997

Kurihara, Enlow, D. H., & Rangel, R. D. (1980). Remodeling reversals in anterior parts of the human mandible and maxilla. *The Angle Orthodontist*, 50(2), 98–106.

Lacruz RS, de Castro, J. M. B., Martinón-Torres, M., O'Higgins, P., Paine, M. L., Carbonell, E., Arsuaga, J. L., & Bromage, T. G. (2013). Facial Morphogenesis of the Earliest Europeans. *PLoS ONE*, 8(6), e65199–e65199.

Lacruz, R. S., Bromage, T. G., O'Higgins, P., Arsuaga, J. L., Stringer, C., Godinho, R. M., Warshaw, J., Martínez, I., Gracia-Tellez, A., de Castro, J. M. B., & Carbonell, E. (2015).

Ontogeny of the maxilla in Neanderthals and their ancestors. *Nature Communications*, 6(1), 1-6.

Liu, Y. P., Behrents, R. G., & Buschang, P. H. (2010). Mandibular growth, remodeling, and maturation during infancy and early childhood. Angle Orthod, 80(1), 97-105.

Martínez-Maza, C., Rosas, A., & García-Vargas, S. (2006). Bone paleohistology and human evolution. *Journal of Anthropological Sciences*, 84, 33-52.

Martinez-Maza, Rosas, A., & Nieto-Diaz, M. (2010). Brief communication: Identification of bone formation and resorption surfaces by reflected light microscopy. American Journal of Physical Anthropology, 143(2), 313–320. https://doi.org/10.1002/ajpa.21352

Martinez-Maza C, Rosas A, García-Vargas S, Estalrrich A, & de la Rasilla M. (2011). Bone remodelling in Neanderthal mandibles from the El Sidrón site (Asturias, Spain). *Biology Letters (2005)*, 7(4):593–596.

Martinez-Maza, Rosas, A., & Nieto-Díaz, M. (2013). Postnatal changes in the growth dynamics of the human face revealed from bone modelling patterns. Journal of Anatomy, 223(3), 228–241. https://doi.org/10.1111/joa.12075

Martinez-Maza, Freidline, S. E., Strauss, A., & Nieto-Diaz, M. (2015). Bone Growth Dynamics of the Facial Skeleton and Mandible in Gorilla gorilla and Pan troglodytes. Evolutionary Biology, 43(1), 60–80. https://doi.org/10.1007/s11692-015-9350-7

McCollum. (1999). The Robust Australopithecine Face: A Morphogenetic Perspective. Science, 284(5412), 301–305. https://doi.org/10.1126/science.284.5412.301

McCollum. (2008). Nasomaxillary remodeling and facial form in robust Australopithecus: a reassessment. Journal of Human Evolution, 54(1), 2–14. https://doi.org/10.1016/j.jhevol.2007.05.013

Milton. (2000). Hunter-gatherer diets—a different perspective. *The American Journal of Clinical Nutrition*, *71*(3), 665–667. https://doi.org/10.1093/ajcn/71.3.665

Moisik, & Dediu, D. (2017). Anatomical biasing and clicks: Evidence from biomechanical modeling. Journal of Language Evolution, 2(1), 37–51. https://doi.org/10.1093/jole/lzx004

Molnar. (2008). Dental wear and oral pathology: Possible evidence and consequences of habitual use of teeth in a Swedish Neolithic sample. American Journal of Physical Anthropology, 136(4), 423–431. https://doi.org/10.1002/ajpa.20824

Noback, & Harvati, K. (2015). The contribution of subsistence to global human cranial variation. *Journal of Human Evolution*, 80, 34–50.

Norton, Bieler, G. S., Ennett, S. T., & Zarkin, G. A. (1996). Analysis of Prevention Program Effectiveness With Clustered Data Using Generalized Estimating Equations. *Journal of Consulting and Clinical Psychology*, 64(5), 919–926.

Oksanen J., Simpson G., Blanchet F, Kindt R., Legendre P., Minchin P., O'Hara R., Solymos P., Stevens M., Szoecs E., Wagner H., Barbour M., Bedward M., Bolker B., Borcard D., Carvalho G., Chirico M., De Caceres M., Durand S., Evangelista, H., FitzJohn R., Friendly M., Furneaux B., Hannigan G., Hill M, Lahti L., McGlinn D., Ouellette M., Ribeiro Cunha E., Smith T., Stier A., Ter Braak C., Weedon J.(2022). _vegan: Community Ecology Package_. R package version 2.6-4, <https://CRAN.R-project.org/package=vegan>.

Pampush J.D., & Daegling, D. J. (2016). The enduring puzzle of the human chin. Evolutionary Anthropology, 25(1), 20–35. https://doi.org/10.1002/evan.21471

Pedersen, P.O. (1947). Dental Investigations of Greenland Eskimos.[Summary.]. *Proceedings of the Royal Society of Medicine*, 40(12), 726–732. https://doi.org/10.1177/003591574704001216

Polanski, J. M. (2011). Morphological Integration of the Modern Human Mandible during Ontogeny. *Int J Evol Biol*, 2011, 545879. https://doi.org/10.4061/2011/54587

Radlanski, R. J., & Klarkowski, M. C. (2001). Bone remodeling of the human mandible during prenatal development. *Journal of Orofacial Orthopedics/Fortschritte der Kieferorthopädie*, 62(3), 191-201.

Rosas A & Martinez-Maza C. (2010). Bone remodeling of the Homo heidelbergensis mandible; the Atapuerca-SH sample. *Journal of Human Evolution*, 58(2), 127–137.

Rosas, Bastir, M., & Alarcón, J. A. (2019). Tempo and mode in the Neandertal evolutionary lineage: A structuralist approach to mandible variation. Quaternary Science Reviews, 217, 62–75. https://doi.org/10.1016/j.quascirev.2019.02.025

Sardi, Novellino, P. S., & Pucciarelli, H. M. (2006). Craniofacial morphology in the Argentine center-west: Consequences of the transition to food production. *American Journal of Physical Anthropology*, 130(3), 333–343. https://doi.org/10.1002/ajpa.20379

Schaefer, Black, S. M., & Scheuer, L. (2009). *Juvenile osteology: a laboratory and field manual*. Academic.

Schuh, Kupczik, K., Gunz, P., Hublin, J., & Freidline, S. E. (2019). Ontogeny of the human maxilla: a study of intra-population variability combining surface bone histology and

geometric morphometrics. *Journal of Anatomy*, 235(2), 233–245. https://doi.org/10.1111/joa.13002

Schuh A, Gunz P, Villa C, Kupczik K, Hublin J, & Freidline SE. (2020). Intraspecific variability in human maxillary bone modeling patterns during ontogeny. *American Journal of Physical Anthropology*, 173(4):655–670.

Schuh, Gunz, P., Kupczik, K., Hublin, J.-J., & Freidline, S. E. (2021). Quantifying maxillary development in chimpanzees and humans: An analysis of prognathism and orthognathism at the morphological and microscopic scales. Journal of Human Evolution, 157, 103031–103031. https://doi.org/10.1016/j.jhevol.2021.103031

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH image to ImageJ: 25 years of image analysis. Nature Methods, 9(7), 671–675.

Scott, G.R. and Winn, J.R. (2011), Dental chipping: Contrasting patterns of microtrauma in inuit and European populations. Int. J. Osteoarchaeol., 21: 723-731. https://doi.org/10.1002/oa.1184

Sellen, D.W., Smay, D.B. (2001). Relationship between subsistence and age at weaning in "preindustrial" societies. *Hum Nat* 12, 47–87. https://doi.org/10.1007/s12110-001-1013-y

Stansfield, E., Mitteroecker, P., Vasilyev, S. Y., Vasilyev, S. & Butaric, L. N. (2021). Respiratory adaptation to climate in modern humans and Upper Palaeolithic individuals from Sungir and Mladec. *Sci Rep* **11**, 7997, doi:10.1038/s41598-021-86830-x.

Stringer, C. (2016). The origin and evolution of *Homo sapiens*. *Philosophical Transactions of the Royal Society B: Biological Sciences* **371**, 20150237, doi: doi:10.1098/rstb.2015.0237.

Terhune, C. E., Ritzman, T. B., & Robinson, C. A. (2018). Mandibular ramus shape variation and ontogeny in Homo sapiens and Homo neanderthalensis. *J Hum Evol*, *121*, 55-71. https://doi.org/10.1016/j.jhevol.2018.03.009

Viðarsdóttir, U. S., O'Higgins, P., & Stringer, C. (2002). A geometric morphometric study of regional differences in the ontogeny of the modern human facial skeleton. *Journal of anatomy*, 201(3), 211-229.

Von Cramon-Taubadel, N. (2011). Global human mandibular variation reflects differences in agricultural and hunter-gatherer subsistence strategies. Proceedings of the National Academy of Sciences - PNAS, 108(49), 19546–19551. https://doi.org/10.1073/pnas.1113050108

Waddington, C. H. (1942). Canalization of the development and the inheritance of acquired characters. *Nature*, 150, 563–565.

Wealthall. (2002). Surface Remodelling of the Facial Skeleton in Juvenile Macaca mulatta: Implications for Sexual Dimorphism. *Folia Primatologica*, 73(1), 49–53.

White, Black, M. T., & Folkens, P. A. (2012). Human osteology (3rd ed.). Academic Press.

Wilson, & Humphrey, L. T. (2017). Voyaging into the third dimension: A perspective on virtual methods and their application to studies of juvenile sex estimation and the ontogeny of sexual dimorphism. Forensic Science International, 278, 32–46. https://doi.org/10.1016/j.forsciint.2017.06.016

Wroe, Parr, W. C. H., Ledogar, J. A., Bourke, J., Evans, S. P., Fiorenza, L., Benazzi, S., Hublin, J.-J., Stringer, C., Kullmer, O., Curry, M., Rae, T. C., & Yokley, T. R. (2018). Computer simulations show that Neanderthal facial morphology represents adaptation to cold and high energy demands, but not heavy biting. Proceedings of the Royal Society. B, Biological Sciences, 285(1876), 20180085–20180085. https://doi.org/10.1098/rspb.2018.0085

Yamada, & Kimmel, D. B. (1991). The effect of dietary consistency on bone mass and turnover in the growing rat mandible. *Archives of Oral Biology*, 36(2), 129–138.

Zeger, S. L., Liang, K.-Y., & Albert, P. S. (1988). Models for Longitudinal Data: A Generalized Estimating Equation Approach. *Biometrics*, 44(4), 1049–1060. https://doi.org/10.2307/2531734