Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations, Spring 1920 to Summer 2023

Graduate Studies

5-2015

Analytical Comparison of Bovine Parentage Single Nucleotide Polymorphisms

Joshua Jeffrey Dallin
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

 Part of the Life Sciences Commons, and the Physical Sciences and Mathematics Commons

Recommended Citation

Dallin, Joshua Jeffrey, "Analytical Comparison of Bovine Parentage Single Nucleotide Polymorphisms" (2015). *All Graduate Theses and Dissertations, Spring 1920 to Summer 2023*. 4450.
https://digitalcommons.usu.edu/etd/4450

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations, Spring 1920 to Summer 2023 by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.

UtahStateUniversity
MERRILL-CAZIER LIBRARY

ANALYTICAL COMPARISON OF BOVINE PARENTAGE SINGLE NUCLEOTIDE POLYMORPHISMS

by

Joshua Jeffrey Dallin

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Applied Science, Technology, and Education
(Agricultural Systems Technology)

Approved:

Dr. Rhonda Miller Major Professor	Dr. Lee F. Rickords Committee Member
Dr. Rebecca Lawver Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2015

Copyright 2015 Joshua Jeffrey Dallin

All Rights Reserved

ABSTRACT

Analytical Comparison of Bovine Parentage

Single Nucleotide Polymorphisms

by

Joshua Jeffrey Dallin, Master of Science

Utah State University, 2015

Major Professor: Dr. Rhonda Miller
Department: Applied Science, Technology, and Education

Often on cow/calf operations and dairy farms, where multiple bulls are exposed to cows either by live cover or artificial insemination, error can be present in parentage record keeping for breed registries or production use. Research has evolved to the integration of using single nucleotide polymorphisms (SNPs) to answer questions where cases of unknown parentage may exist. With the evolution of the research, differentiated panels have been created specifically for parentage analysis. Our objective was to complete an analytical comparison between two specific panels, a proven 88 parentage SNP panel and a recently developed 25 SNP panel. A smaller panel would be beneficial in a parentage test as the smaller panel would reduce time and costs associated with the parentage analysis. In this study, 3,678 cattle samples were collected and prepared from offspring, sires, and dams. Parentage calling software was used to identify the parentage assignments of the samples. Through these procedures and comparisons, it was

determined that the smaller 25 SNP panel did not have the magnitude or strength necessary to be able to correctly identify cattle in the same manner as the 88 SNP panel.

(116 pages)

PUBLIC ABSTRACT

Analytical Comparison of Bovine Parentage

Single Nucleotide Polymorphisms

by

Joshua Jeffrey Dallin, Master of Science

Utah State University, 2015

Often on cattle operations and dairy farms, where multiple bulls are exposed to cows either by live cover or artificial insemination, error can be present in parentage record keeping for breed registries or production use. Research has evolved to the integration of using single nucleotide polymorphisms (SNPs) to answer questions where cases of unknown parentage may exist. With the evolution of the research, differentiated panels have been created specifically for parentage analysis. Our objective was to complete an analytical comparison between two specific panels, a proven 88 parentage SNP panel and a recently developed 25 SNP panel. A smaller panel would be beneficial in a parentage test as the smaller panel would reduce time and costs associated with the parentage analysis. In this study, nearly 4,000 cattle samples were collected and prepared from offspring, sires, and dams. Parentage calling software was used to identify the parentage assignments of the samples. Through these procedures and comparisons, it was determined that the smaller 25 SNP panel did not have the magnitude or strength necessary to be able to correctly identify cattle in the same manner as the 88 SNP panel.

ACKNOWLEDGMENTS

I would first like to start out by thanking Dr. Miller, Dr. Rickords, and Dr. Lawver for their unwavering support, guidance, and patience. I would especially like to thank Dr. Rickords for his kindness and mentoring that made this whole project possible from the very beginning. Secondly, I would like to thank my teammates and close friends, Marcy Labrum and Kim Blanchard, for the fun stories, great times, and especially the amazing help from both near and far. Al Dustin made everything possible; I appreciate Al's willingness to let me tag along and learn from his goodness and expertise. Dr. Allen Young, Brett Bowman, and Dr. Ben Sessions are to be thanked for helping connect us to important samples both on campus and off. I would like to thank all of the cattle owners for opening up their herds to research. I also appreciate the kindness that was always shown to us when visiting their operations. I would like to thank my 4-H colleagues and friends, Kevin, Craig, Jim, JP, Dave, Stacey, Kelsey, Lorry, Whitney, and Cassie, for being so supportive and encouraging of this project. I would like to acknowledge how grateful I am to my very supportive parents and siblings. Daxtin and Knox, you have been my motivation! Lastly, I would like to acknowledge my sincere gratitude and love for my incredible wife, Brooke. Whether it was cutting thousands of tail hairs on "date night," putting up with long hours spent in the lab, or dealing with me inputting data into the wee hours of the morning, you have been nothing but the best champion of support. From this project I have made some everlasting relationships with the amazing people mentioned above and many others that were met along the way. I consider myself one of the luckiest for this outstanding opportunity.

Joshua Jeffrey Dallin

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

REVIEW OF LITERATURE

Restriction Endonucleases and Genetic Diversity

For years, questions of paternity presented a significant challenge to scientists and potential parents alike. During the first half of the twentieth century, thanks to a discovery by Karl Landsteiner, researchers often turned to people's ABO phenotypes when such issues arose; however, ABO blood group information could only be used to exclude potential fathers, rather than confirm the presence of a parental relationship (Adams, 2008).

In 1977, A. J. Jeffreys had the idea, "Why not try to marry together genomics with the classic discipline of human genetics and try to detect heritable variation directly in human Deoxyribonucleic Acid (DNA)?" (Jeffreys, 2005). This idea was the beginning stages of research that would revolutionize the world of genetics as we know it today.

The experiment began with rabbit liver DNA that was cleaved with restriction endonucleases. The fragments were denatured, separated by electrophoresis in an agarose gel, and transferred by blotting onto nitrocellulose filters (Jeffreys & Flavell, 1977).

The nitrocellulose filters were then hybridized with a DNA copy of rabbit β-globin messenger Ribonucleic Acid (RNA) from a plasmid. From this, a minimal amount of rabbit DNA fragments were detected. These fragments were found to form well matched hybrids with the PβG1plasmid (Maniatis, Kee, Efstratiadis, & Kafatos, 1976).

From these fragments, two β-globin DNA fragments were produced using endonuclease Eco R1 which cleaves the rabbit β-globin gene. Using double digests, these fragments were ordered into a single map of restriction endonuclease cleavage sites around the β-globin gene in the rabbit genome (Jeffreys & Flavell, 1977).

These breakthroughs led to the first official physical map of the mammalian gene (Jeffreys, 2005). From this, small specific areas of interest could be isolated and genetically evaluated. This opened the door to further research and discovery (Jeffreys & Flavell, 1977).

The next step, as is the case with most mammalian studies, was to attempt to move into the human realm. The quest was to begin to locate variation within human DNA. To start, the genome of the short arm of chromosome 11 was analyzed and was found to have at least four structural genes coding for the β-related globin and δ-related globin polypeptides (Deisseroth et al., 1978). These polypeptides were similar to those that were used from the rabbit genome (Jeffreys & Flavell, 1977).

This time, instead of one endonuclease, eight endonucleases were used to screen DNA prepared from sixty human individuals. The goal was to look for unusual patterns of globin DNA fragments that came from DNA sequence variants which had either destroyed or created new restriction endonuclease cleavage sites surrounding the β-related globin genes (Jeffreys, 1979).

The first sign of a variant restriction enzyme pattern was found in a healthy Caucasian woman who showed an unusual endonuclease Pst 1 cleavage pattern which contains the δ-globin gene. This also showed the appearance of two new DNA fragments. Interestingly enough, the combined length of the two fragments closely approximated the

length of the normal δ-globin gene fragment. It appeared that the woman was heterozygous for the presence/absence of a Pst 1 cleavage site within the δ-globin gene (Jeffreys, 1979).

A later experiment was carried out on the woman's mother and father. The father had a normal Pst 1 pattern, whereas the mother was also heterozygous for the new Pst 1 cleavage site. This was found to be a trait that was inherited in a normal Mendelian fashion, and was not due to a mutation (Jeffreys, 1979).

Perhaps the most important and fascinating discovery of this experiment was polymorphisms that were found in what was labeled as the Hind III site. This discovery during a screening for restriction enzyme cleavage site variants in and around the β-related globin genes enabled scientists to estimate the overall degree of genetic diversity in the human genome. The discovery of this analysis allowed for application to more extensive screenings with wider ranges of restriction endonucleases which would reveal rarer restriction enzyme cleavage site variants and more precisely define numbers of different DNA variants and their gene frequencies in humans (Jeffreys, 1979).

The length of polymorphisms can be visualized by treating DNA samples with restriction enzymes. In this process, the restriction enzymes cleave DNA. In turn, fragments that contain a repeating segment are produced (Hartl & Lewontin, 1994). In order to separate the DNA fragments by size, a molecular sieving technique of gel electrophoresis is utilized (Renkin, 1954). After transferring the fragments from a gel to a nylon membrane, the membranes were treated with a radioactive probe. The probe bound a selected repetitive nucleotide sequence on the membranes. Nylon sheets were then placed against an X-ray sensitive film. This revealed positions of DNA fragments

carrying the radioactive markers in a series of lines that closely resemble bar codes (Jones, 2004). These lines proved to have two important qualities: the pattern varied from person to person, which enabled the characterization of an individual's DNA; and the patterns were inherited, which provided information about family relationships (Jones, 2004).

Restriction Fragment Length Polymorphism

This amplification process led to further research, finding that insertion or deletion of DNA amid two recognition sites for an enzyme often modifies the size of the restriction fragment shaped by digestion. Differences in the sizes of fragments resulting from the digestion of DNA have been labeled restriction fragment length polymorphisms (RFLPs) (Gusella, 1986).

RFLPs quickly became a method used to follow a particular sequence of DNA as it is passed on to other cells. This research became key in accomplishing many different objectives. Paternity cases were among the top uses of the new technology (Campbell, 2001).

An RFLP is a sequence of DNA that has a restriction site on each end and carries a target sequence in between. The target sequence can be any segment of the DNA that binds to a probe. This binding then forms complementary base pairs. The probe is created by taking single-stranded DNA and tagging it with enzymes so that it can more easily be visualized. Probes fluoresce when they anneal to their targets, permitting detection of the binding site (Campbell, 2001).

Until the characterization of RFLPs, the genetic markers available were comparatively few in number and scattered randomly throughout the genome. DNA polymorphism took a leading role in genetic research as investigators realized the potential power and relative ease of the technology provided by RFLPs. Within relatively few years, this tool had a key impact on several important genetic illnesses (Gusella, 1986).

Deoxyribonucleic Acid Fingerprinting

As research progressed and geneticists became more familiar with the patterns that were expressed from the DNA in RFLPs, it was discovered that repetitive patterns of DNA, known as variable number tandem repeats (VNTRs) (Moreau-Horwin, 2015) or minisatellites (Crawford & Beaty, 2013), were present in all humans but that they varied in length for each individual. This variation could be used to establish the identity of a person. The newly found technique soon became coined as genetic fingerprinting (Jeffreys, Wilson, Thein, Weatherall, & Ponder, 1986). This would be catapulted into the world of forensic science when two murders were committed not far from the University of Leicester. For the first time, genetic fingerprinting was used to acquit a suspect and sentence the guilty person (Moreau-Horwin, 2015).

It was not long until this discovery was being utilized in other species of plants and animals. Certain genetic diseases, immunities, and other findings were quickly being researched to determine their cause by using DNA fingerprinting methods (Luangtongkum et al., 2015).

Deoxyribonucleic Acid Profiling

Genetic fingerprinting led to a more detailed understanding of DNA. It explained what had been described by James Watson and Francis Crick in their groundbreaking article “Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid,” which was published in 1953. The article described the structure of a DNA molecule (Watson & Crick, 1953).

Although the structure of a DNA molecule was found to be correct and reproducible, research soon proved that the original method of analysis required the use of large amounts of high quality DNA, which could be difficult to obtain (East Midlands Forensic Pathology Unit, n.d.). Interestingly enough, as the use of DNA fingerprinting was being developed, Dr. Kary Mullis was working with a team to create a way to solve the quantity need in the form of amplification.

Polymerase Chain Reaction

Dr. Mullis was successful in adding a thermostable DNA polymerase solution to an in vitro DNA amplification procedure. This became coined as the polymerase chain reaction (PCR). The isolated enzyme greatly streamlined the procedure and enabled the amplification reaction to be achieved at higher temperatures. It significantly advanced the specificity, yield, sensitivity, and length of the products that could be amplified (Saiki et al., 1988).

The research was so successful that the team was able to report that single-copy genomic sequences were amplified by a factor of more than 10 million. These sequences

came back with very high specificity, and DNA segments up to 2,000 base pairs were promptly amplified. In addition to this success, the technique was used to amplify and detect a target DNA molecule existing only once in a sample of 10^5 cells (Saiki et al., 1988).

PCR became known as the "ingenious new tool for molecular biology" (Innis & Gelfand, 1990). The process became fine-tuned and created ways that were delicate enough that a single DNA molecule was able to be amplified, and single-copy genes were regularly extracted out of complex mixtures of genomic sequences and viewed as distinct bands on agarose gels (Innis & Gelfand, 1990).

Amplified Fragment Length Polymorphism

Science continued to dive even deeper into identification and parentage testing utilizing DNA by looking for variances in fragment lengths due to the occurrence or nonappearance of a restriction enzyme site, or due to an insertion or deletion that transpires between two restriction enzyme sites (Chial, n.d.).

Amplified Fragment Length Polymorphism (AFLP) was first described by researcher P. Vos and colleagues in 1995. The AFLP technique was based on selective PCR amplification of restriction fragments from a total digest of genomic DNA. The technique was described in three steps: (i) a restriction of the DNA and ligation of oligonucleotide adapters, (ii) the selective amplification of sets of restriction fragments, and (iii) a gel analysis of the amplified fragments (Vos et al., 1995). The results from this breakthrough study demonstrated that the AFLP technique was an effective way to amplify large numbers of fragments concurrently. It also proved that the technique would

generate fingerprints of any DNA regardless of the source or complexity, allowing for the design of high-density genetic maps of genomes or genome segments (Vos et al., 1995).

Short Tandem Repeat

PCR and research of RFLPs opened the door to genetic maps and areas of DNA sequences that seemed unconceivable just a few years previously. The study of DNA revealed that genomes were full of repeating DNA sequences. Specifically, in 1991, it was discovered that there were definite short repeat units that usually contained 2–6 base pairs of length ("STR - Forensic DNA Testing System," n.d.). These repeated sections are termed short tandem repeats (STRs), also known as microsatellites (Calafell, Shuster, Speed, Kidd, & Kidd, 1998). Microsatellites are among the most variable forms of DNA sequence in the genome (Ellegren, 2004). It was found that the STRs presented several properties that made them chiefly useful in population genetics: the huge number of STR loci available and the relative practicality with which they could be typed allowed for simpler gathering of genetic data for a large number of loci within population samples (Calafell et al., 1998). STRs also proved to be useful because they allowed easier identification without problems of variance amplification. Due to their smaller size, STR alleles were found to be easily divided from other chromosomal locations to ensure closely linked loci were not chosen. They were also found to have lower mutation rates, which makes data more constant and predictable ("STR - Forensic DNA Testing System," n.d.).

With all of the benefits that STR has to offer it is still very popular today as a method for genetic profiling using 4 to 5 nucleotide repeats, which allows scientists to

obtain relatively precise and accurate information. It also serves as a means for somewhat "mistake-proof" information to be gathered even in poor conditions ("STR Analysis - Short Tandem Repeat," n.d.).

Single Nucleotide Polymorphism

Research progressed in the study of genotyping and parentage analysis. The genomic world seemed to be small with only a few base pairs comprising STR microsatellites, but science was ready to go even smaller. Along came the discovery of the single nucleotide polymorphism (SNP).

SNPs are the most abundant form of genetic variation and a resource for mapping multifaceted genetic traits (Collins, Guyer, & Chakravarti, 1997). They have emerged as the genetic marker of choice for mapping ailment loci and candidate gene association studies, due to their high density and relatively level distribution in genomes (Wang et al., 2009).

By taking advantage of genomic sequencing, scientists are able to use uncomplicated yet more accurate methods for sequence configuration: fragment clustering, paralogue identification, and multiple arrangements. With SNPs, rigorous treatment of base quality permits completely automatic evaluation of the full span of all sequences, without restrictions on alignment depth (Marth et al., 1999).

Some researchers argue that SNPs have lower variation compared with microsatellites (Brumfield, Beerli, Nickerson, & Edwards, 2003), but others believe that SNPs should make the comparison of genomic diversities and histories of different species more direct than has been possible with microsatellites (Brumfield et al., 2003).

Roughly 90% of genetic difference in genomes is in the form of SNPs (Collins, Brooks, & Chakravarti, 1998), the result of point mutations that yield single base-pair differences among chromosome sequences. Large databases of overlapping SNP sequences and the development of large-scale SNP identification technologies has allowed for the enhancement of classification and use of markers (Brookes, 1999; Picoult-Newberg et al., 1999; Sachidanandam et al., 2001).

Through the development of identification and screening tools (Sachidanandam et al., 2001), SNPs have provided major benefits in parentage analysis. For example, with SNPs only a small quantity of DNA is needed (Chakraborty, Stivers, Su, Zhong, & Budowle, 1999). It has also been found that creating profiles for SNPs is much faster than STR profiles (Chakraborty et al., 1999). One of the most significant differences for SNPs is their lower cost for genotyping than other forms (Baruch & Weller, 2008; Heaton, 2008). Today, SNPs have been labeled as the top standard for genotyping in determining parentage (Heaton, 2008).

There are many systems and technologies that are currently available for detecting SNP markers and for genotyping. TaqMan®, utilized at the School of Aquatic and Fishery Sciences at the University of Washington, Seattle, USA, is one of the simpler systems, requiring only a thermal cycler, which results in cost savings (Perkel, 2008).

To look at a handful of SNPs, PCR-based TaqMan® is utilized to help run the American and US–Canadian treaty fisheries. TaqMan® probes are constructed to hybridize to a precise SNP allele, with a different 5′ fluorophore color for each allele. As a specific color or both colors light up throughout amplification, the genotype at a certain SNP can be easily ascertained (Perkel, 2008).

TaqMan® is performed as a singleplex reaction. This means that it requires just one tube and one SNP. It can be multiplexed, if desired, to 3 or 4 SNPs per reaction with other fluorescent colors. It is problematic to multiplex any further than this because more progressive chemistries would be required.

Not only is the process quick and effective for locating the desired alleles, it is reported that advancements have allowed spending on TaqMan® reagents to decrease by about 98% (Perkel, 2008).

Another method for SNP detection, in place of PCR, is the use of a chip format. Affymetrix® (Santa Clara, CA, USA) and Illumina® (San Diego, CA, USA) are two of the main companies in the United States that compete in the development of DNA microarray SNP testing (Perkel, 2008). Chips are solid supports, usually of glass or silicon, upon which DNA is fixed in an organized grid fashion. Each spot of DNA, called a probe, represents a single gene (Tiwari, 2013).

Although new technology continues to be developed, many challenges still exist with microarray chips. One of the biggest problems is cost-effectiveness. In the development phase, a large number of individuals are usually required to be genotyped on a small subgroup of candidates' SNPs. Many SNP genotyping methods have been established on a variety of platforms in the past decade (Chen & Sullivan, 2003; Kwok, 2001), such as the GeneChip® array and the BeadArray™ (Hardenbol et al., 2005). These methods have offered ultra-multiplex and high-throughput genotyping but are not cost-effective for genotyping a large number of samples for a small number of SNPs (Wang et al., 2009). This especially would not be financially effective for cattle producers seeking to determine parentage.

High-Throughput Single Nucleotide Polymorphism Genotyping

In an effort to settle issues of cost effectiveness, high-throughput SNP genotyping assays were created. For this project two assays were utilized: One assay developed by KBioscience® is called the Kompetitive Allele-Specific PCR genotyping system (KASP) (Robinson & Holme, 2011). The other, Fluidigm®'s SNPtype™ assays, was developed by the Fluidigm® Corporation. Both of these assays function in very similar ways and have only a few minor differences. These assays are made up of three main components: The first is a primer mix that contains two allele specific forward primers that both contain their own unique tail sequence. The second component is a common reverse primer. The third component of the assay is a PCR master mix. The master mix is where the KASP and Fluidigm® SNPtype™ assays slightly differ. For the KASP assay two proprietary oligos are added to the master mix. One is a labeled with FAM (with a corresponding quencher) and the other is labeled with VIC (with a corresponding quencher). In the case of the Fluidigm® SNPtype™ assay, one of the oligos is labeled with FAM and the other is labeled with HEX. These are called Fluorescence Resonance Energy Transfer (FRET) cassettes. In order to make all of this work properly, the FAM/VIC oligos and the FAM/HEX oligos are designed with identical sequences to those of the two allele specific forward primers. All of these components from the assay are then mixed with the extracted DNA of the desired specimen. These components are then loaded onto a chip and undergo PCR. After denaturation of the DNA occurs, one of the forward primers anneals with the template DNA. Annealing depends on specific nucleotides that are located in specific SNP locations. The other forward primer may

bind, but it will not elongate because of mismatching. At this point another round of PCR is given to the assay mix. During this reaction the reverse/forward primers bind to opposite strands. Both strands then elongate at the same time. This creates two double stranded molecules from an original one double stranded molecule. Because of the reverse primer's properties, the complement of the forward primer strand is generated. This complement is very important because it is what allows detection to occur. Through additional PCR cycles, the forward primer and complement strands continue to be amplified. After many PCR cycles, primers come off and on their specific target strands during the denaturation and annealing process. At this point, the fluorophore containing sequence has removed itself from its corresponding quencher (as was mentioned above). This removal is what causes the fluorophore attached to the sequence to fluoresce and allows detection to occur. In this type of fluorescence, if only one fluorophore fluoresces (FAM, HEX, or VIC), it indicates that the DNA sample in question contains homozygous alleles. In heterozygosity, two signals would fluoresce (LGC Genomics, n.d.).

Technology has developed that allows the tested SNPs to be run against numerous samples. Fluidigm® has developed an HP 96.96 Genotyping™ Integrated Fluid Circuit (IFC) system (Fluidigm Corporation, 2014). It allows 9,216 reactions of 96 assays and 96 samples to be run when using a 96x96 array. The IFC is a grid of fluid lines that are known as NanoFlex™ valves and chambers. The valves are made of a material which refracts under pressure to create a tight seal. This seal regulates the flow of liquids in the system. Individual samples are pipetted by hand into the sample inlets, and genotyping assays are pipetted into the assay inlets. The seal allows for individual samples to mix with individual assays in a controlled environment. Once samples have been loaded, PCR

is accomplished by placing the plate into a thermocycler (Wang et al., 2009). Once the oligos have fluoresced, endpoint fluorescent image data is then acquired using the BioMark™ System for Genetic Analysis. Data is then examined using the Fluidigm® SNP Genotyping Analysis software to obtain genotype calls (Wang et al., 2009).

Cattle SNP Analysis

A study was accomplished by the United States Department of Agriculture to determine the validity of bovine SNPs. There are currently 121 SNPs that have been approved ("USDA-MARC SNPs for parentage, traceback, and animal ID," 2008) (Table 1). These SNPs were not easy to locate, but it was accomplished by accessing and searching GenBank ("GenBank Home," 2015). In 2012, the International Society for Animal Genetics (ISAG) established a requirement that a minimum of 95 SNPs out of the 121 SNPs must be utilized for validation in parentage testing (Morrin & Boscher, 2012). The ISAG provided a core bank of 100 SNPs to be utilized (Table 2).

Correlated with the high number of required SNPs is the cost to have them processed. Generally, the "average" cattle rancher does not have the funding to be able to afford the 95 SNP panels. In an effort to research the validity of a smaller SNP panel, this project focused on analytically comparing two SNP panels that were less than the minimum 95 SNPs. The first is currently manufactured by Fluidigm®. The company sells a set of allele-specific primers (called SNPtype™ assays) for an 88 parentage SNP panel (reduced from the USDA's original 121) (Blanchard, 2013). This is a proven panel that has been deemed valid for correct parentage calls in the beef species.

The second panel is a 25 SNP array that was developed in 2009 by Karniol et al. (Table 3). SNaPshot®, a primer extension based method, was utilized to multiplex 38 markers to test relevance and potential use in a cattle parentage SNP panel. 25 out of the 38 tested markers were found to have an exclusion power of 99.9% (Karniol et al., 2009) when the genotypes of both believed parents were known. It is important to note that both dam and sire were used in this study. In an effort to reduce overall costs, in our analysis we tested both panels' ability to handle only one believed parent.

Table 1. *GenBank® Accession Numbers for the 121 USDA-MARC SNPs for Parentage, Traceback, and Animal ID*

DQ381153	AY842475	AY850194	EF034081	AY914316	EF089234
DQ451555	DQ485413	DQ837644	AY853302	DQ888311	DQ995977
DQ404150	DQ647188	DQ674265	AY853303	EF026084	DQ990833
DQ404149	DQ470475	DQ846689	DQ846690	DQ888312	AY939849
AY761135	DQ500958	DQ786765	DQ984826	AY916666	EF034083
DQ404151	DQ647189	DQ786766	DQ846691	EF164803	AY941204
DQ404152	DQ468384	DQ786760	DQ846692	DQ984828	AY942198
AY776154	DQ846688	DQ786761	EF042090	DQ888313	DQ990834
AY841151	AY844963	DQ984827	DQ866817	DQ990835	AY943841
DQ422949	DQ647190	DQ786762	DQ866818	AY919868	EF034086
DQ786757	DQ789028	DQ984825	DQ846693	DQ995976	EF150946
DQ422950	AY849380	DQ837646	AY857620	EF093511	EF093512
DQ647187	AY849381	DQ837645	DQ846694	EF026085	EF034084
AY842472	DQ888309	AY851162	DQ846695	DQ916059	EF141102
AY842474	DQ786758	DQ837643	DQ888310	DQ990832	EF034085
AY842473	DQ650635	AY851163	DQ381152	EF093509	EF034087
DQ435443	DQ916057	DQ786763	AY858890	EF093510	EF026086
DQ489377	DQ786759	DQ786764	AY860426	EF034082	EF042091
DQ839235	DQ916058	DQ832700	EF028073	AY929334	AY856094
DQ647186	DQ650636	EF026087	AY863214	AY937242	EF034080
					DQ404153

Table 2. *GenBank® Accession Numbers for the SNPs Included in the ISAG Reference Core Panel (Blanchard, 2013; Morrin & Boscher, 2012)*

AY761135	AY929334	DQ647189	DQ846691	EF026084
AY776154	AY937242	DQ647190	DQ846692	EF026086
AY842472	AY939849	DQ650635	DQ846693	EF026087
AY842473	AY941204	DQ650636	DQ866817	EF028073
AY842474	AY943841	DQ674265	DQ866818	EF034080
AY842475	DQ381152	DQ786757	DQ888309	EF034081
AY844963	DQ381153	DQ786758	DQ888310	EF034082
AY849381	DQ404149	DQ786759	DQ888311	EF034083
AY850194	DQ404150	DQ786761	DQ888313	EF034084
AY851162	DQ404151	DQ786762	DQ916057	EF034085
AY851163	DQ404152	DQ786763	DQ916058	EF034086
AY853302	DQ404153	DQ786764	DQ916059	EF034087
AY853303	DQ435443	DQ786766	DQ984825	EF042090
AY856094	DQ451555	DQ789028	DQ984826	EF042091
AY858890	DQ468384	DQ837643	DQ984827	EF093509
AY860426	DQ470475	DQ837644	DQ990832	EF093511
AY863214	DQ489377	DQ837645	DQ990833	EF093512
AY914316	DQ500958	DQ839235	DQ990834	EF141102
AY916666	DQ647186	DQ846688	DQ995976	EF150946
AY919868	DQ647187	DQ846690	DQ995977	EF164803

Table 3. *GenBank® Accession Numbers for the 25 SNP Panel by Karniol et al. (Karniol et al., 2009)*

SNP no.[1]	GenBank accession no.	Allele 1	Allele 2	Hol. (*N* = 107)	Sim. (*N* = 52)	Lim. (*N* = 40)	Char. (*N* = 31)	Ang. (*N* = 39)	Tux. (*N* = 20)
2	AF440368	G	C	0.61	0.72	0.39	0.79	0.58	0.45
5	AF440372	C	T	0.44	0.56	0.78	0.69	0.27	0.45
6	AF440381	C	T	0.67	0.81	0.89	0.90	0.37	0.92
7	AF440377	C	T	0.38	0.15	0.10	0.08	0.03	0.16
8	AF440369	G	A	0.67	0.79	0.91	0.73	1.00	0.63
9	AF440365	C	T	0.46	0.59	0.24	0.45	0.19	0.17
11	AJ505155	C	T	0.53	0.25	0.14	0.34	0.25	0.10
13	AJ505160	G	A	0.78	0.14	0.9	0.89	0.84	0.68
14	AJ505159	C	T	0.51	0.40	0.23	0.11	0.15	0.37
15	AJ496639	G	A	0.49	0.61	0.8	0.58	0.54	0.71
16	AJ496641	C	T	0.72	0.72	0.79	0.53	0.78	0.63
17	AJ496635	C	T	0.14	0.38	0.41	0.42	0.29	0.50
18	AJ496636	G	A	0.50	0.74	0.34	0.45	0.6	0.87
19	AJ496762	G	A	0.46	0.44	0.46	0.38	0.38	0.50
20	AJ496763	C	T	0.35	0.42	0.63	0.61	0.25	0.66
21	AJ496765	G	A	0.19	0.42	0.39	0.34	0.49	0.34
25	AJ496773	C	T	0.42	0.69	0.78	0.61	0.49	0.79
26	AJ496774	G	T	0.61	0.65	0.63	0.53	0.47	0.53
27	AJ506786	C	T	0.51	0.37	0.42	0.59	0.4	0.60
28	AJ496776	G	A	0.41	0.28	0.39	0.30	0.65	0.24
30	AJ496782	C	T	0.45	0.16	0.12	0.26	0.33	0.37
31	AJ496785	C	T	0.56	0.66	0.66	0.72	0.47	0.37
32	AJ496786	G	A	0.75	0.76	0.79	0.76	0.54	0.92
35	AF440378	G	C	0.24	0.1	0.23	0.11	0.06	0.26
38	AJ505157	G	T	0.58	0.62	0.75	0.59	0.38	0.60
Exclusion probability		One-parent[2]		0.932	0.892	0.844	0.892	0.893	0.892
		Two-parent[3]		0.9995	0.9992	0.9985	0.9990	0.9988	0.9990
ID power[4]				2.3×10^{-10}	1.5×10^{-9}	1.1×10^{-8}	2.5×10^{-9}	3.9×10^{-9}	2.4×10^{-9}
M_{max}[5]				43.7×10^{6}	6.8×10^{6}	0.9×10^{6}	4.0×10^{6}	2.6×10^{6}	4.1×10^{6}

Hol., Holstein; Sim., Simmental; Lim., Limousin; Char., Charolais; Ang., Angus; Tux., Tux Cattle.

[1]SNP numbers follow Werner *et al.* (2004).

[2]The probability of detecting a falsely reported parent with offspring, where the other parent is missing (equation 2a, Jamieson & Taylor 1997).

[3]The probability of exclusion, where both parents are falsely recorded (equation 3a, Jamieson & Taylor 1997).

[4]The probability that two individuals would have identical genotypes by chance (equation 1, Weller *et al.* 2006).

[5]M_{max} is the maximum number of individuals that can be resolved (equation 3, Weller *et al.* 2006).

Parentage Determination Calls and Genotyping

Fluidigm® SNP Genotyping Analysis software is used to obtain genotype calls after PCR has been accomplished and fluorescent image data obtained using the BioMark™ HD machine. For the 88 SNP panel, the software calculates the fluorescent signals from the FAM and HEX fluorophores and generates a scatter plot with the FAM relative intensity (to ROX background) on the X-axis and HEX relative intensity on the Y-axis. These data are also displayed, respectively, by FAM and VIC fluorophores that were created specifically by the KASPar™ SNP Genotyping System for the 25 SNP panel (KBiosciences, 2015). The color codes are arranged using a *k*-means clustering algorithm based on nearest-centroid sorting (Wang et al., 2009). The genotype call data are then exported as an Excel file. The file is then uploaded to Excel and run through macros for the 88 SNP panel and modified macros for the 25 SNP panel, both created in house by Kim Blanchard. These allow the information to be placed into the correct orientation for use in parentage software programs (CERVUS 3.0 and SireMatch 2.0) that then make parentage determination calls (Blanchard, 2013).

The Exclusion Method in Parentage Determination

The simplest and most straightforward process of parentage determination is called the exclusion method. In this process, potential parents are matched up adjacent to each offspring and their genotypes are compared. As they are being matched, if at least one allele in comparison does not match up at any given locus, it is then noted as a conflict. For example, if the genotype of the assumed sire is C:C and the calf is T:T, the

offspring would not have received an allele at that locus from the specific C:C sire. This difference would result in an exclusion being noted for that sire/calf combination. On the other hand, if the assumed sire genotype was C:C and the calf was reported as A:C, an exclusion would not be recorded because the offspring could possibly have received the C allele from this presumed sire. In these tests, there is always the possibility of mutations, call errors, genotyping errors, or unknown null alleles. Because of this, most researchers and software packages have allowed for at least one conflict call to be made before completely excluding a candidate parent. After all candidate parents have been evaluated in comparison to a calf, the candidate with zero exclusions is called as the parent. If no candidate parent/offspring pairing had zero conflicts, a step down process occurs and the candidate with one recorded exclusion is assigned parentage to that calf. If no sire or multiple sires meet these criteria, parentage is marked “unknown” for that calf (Baruch & Weller, 2008; Blanchard, 2013; Dodds, Tate, & McEwan, 1996; Langlois, 2005; Van Eenennaam et al., 2007).

Exclusion probability is defined as the probability that a randomly chosen individual other than a true parent of a randomly chosen offspring in the group be confirmed as a true parent, assuming a population is in Hardy–Weinberg equilibrium (Van Eenennaam et al., 2007). Exclusion probability is assessed from population frequencies of marker alleles (Visscher, Woolliams, Smith, & Williams, 2002). Jamieson and Taylor (Jamieson & Taylor, 1997) and others (Dodds et al., 1996; Marshall, Slate, Kruuk, & Pemberton, 1998) have devised equations for parentage exclusion probabilities based on three common scenarios that can arise in parentage designations: (in our case with sires) when genotypes of an assumed sire, a known sire, and a calf are present; when

only a calf and one sire genotype are present; and when the genotypes of two potential sires and a calf are present (Baruch & Weller, 2008). These methods are commonly used and cited (Baruch & Weller, 2008; Van Eenennaam et al., 2007). They can also be found in publicly available software packages, including a specific product used for this project, CERVUS 3.0 (Blanchard, 2013; Kalinowski, Taper, & Marshall, 2007; Marshall et al., 1998).

Marker panels with higher exclusion probabilities are more successful at resolving parentage exclusions than those that have lower exclusion probabilities. It has been found that when increasing the number of marker loci in a panel, the exclusion probability also increases (Van Eenennaam et al., 2007). This is due to the higher number of loci available. The unfortunate reality is that more markers usually equates to a higher cost to run the analysis. This generally forces people to find a healthy balance between cost and accuracy. If each marker's individual exclusion probability is high, fewer markers are needed in a screening panel than if there is a low number of exclusion probability markers involved. With SNPs, higher minor allele frequencies (MAF) (close to 0.5) generate greater exclusion probabilities and, consequently, allow for a reduction in the number of markers (Baruch & Weller, 2008). Attempts have been made to successfully reduce panel size and thus genotyping costs by excluding from panels the markers that had the lowest probabilities of exclusion (Blanchard, 2013; Gomez-Raya et al., 2008).

The exclusion method is generally easy to perform; however, it does present some downsides. As mentioned previously, due to the possibility of mutations, call errors, genotyping errors, or unknown null alleles there are often multiple candidate parents that could remain non-excluded. Because of this the test does not indicate which of the non-

excluded candidates is the true parent (Marshall, 2007; Marshall et al., 1998). Many argue that the method of exclusion is very limited because it utilizes only a portion of the information available in the data. It also does not adequately adjust to account for the genotyping errors mentioned above (Anderson & Garza, 2006). It has also been discussed that potential genotyping errors can lead to false exclusions and ultimately wrong decisions if exclusion is the only method used (Blanchard, 2013; Hill, Salisbury, & Webb, 2008; Langlois, 2005).

Determining Parentage Using the Likelihood Method Using CERVUS 3.0

Another option for determination of parentage is the use of likelihood statistics. This essentially uses the opposite approach of the exclusion method. CERVUS 3.0, a software program created by Marshall et al. (Kalinowski et al., 2007; Marshall, 2007; Marshall et al., 1998), uses likelihood calculations to make parentage calls.

Likelihood analysis is defined as the percentage of likelihood that a sire is the parent to a calf. Edwards further noted that the variable quantity in a likelihood statement is the hypothesis, and the outcome is what is actually observed. This differs from a probability statement where the hypothesis is assumed and fixed, and there are a variety of possible outcomes (Blanchard, 2013; Edwards, 1999).

The informative nature of likelihoods comes when sires are compared relative to another; this is labeled as the likelihood ratio (Edwards, 1999). For example, in mathematical form, the likelihood L of hypothesis H given data D is shown as $L(H \mid D)$. The likelihood ratio, or evaluation of one hypothesis (e.g., H_1) relative to another (e.g., H_2), can be written as

$$L(H_1,H_2 \mid D) = \frac{P(D \mid H_1)}{P(D \mid H_2)}$$

where $P(D \mid H_1)$ is the probability that under hypothesis H_1, data D can be achieved (Blanchard, 2013; Marshall, 2007). In the case of paternity determination for our study (with known dam), the genotypes at a specific locus of the offspring, dam, and assumed sire are the data *D*. The hypothesis that the assumed sire is the true sire (H_1) is tested against the hypothesis that the assumed sire is an individual chosen at random from the group and is unrelated (H_2) (Marshall et al., 1998). An individual whose likelihood ratio is highest compared to the likelihood ratios of other candidate sires is much more likely to be the true parent than not the true parent (Marshall, 2007; Marshall et al., 1998). This is where the term "likelihood" comes into play as the highest percentage parent is selected.

A benefit to the use of the likelihood method (using CERVUS 3.0) for each locus is that two sources of information skipped by the exclusion method are collected about the believed parent: the frequency of the offspring allele(s) that potentially came from the candidate parent(s) and whether the candidate parent(s) is homozygous or heterozygous. With this information, the parent that is most likely to be the true parent can be determined. For example, in cattle a paternity analysis case is presented where the sires's identity is unknown and the calf's genotype at a given locus is AB, with A being a common allele having a frequency of 0.5 and B with a frequency of 0.1 (a rare allele). Any sire that carries either allele (A or B) would be considered a potential sire. If three candidate sires were tested: sire 1, with a genotype of AC; sire 2, with a genotype of AA; and sire 3, with a genotype of BB (Blanchard, 2013) then none of these sires could be excluded and paternity would be left uncertain because of common alleles between them

and the offspring. However, utilizing the most likely approach, the heterozygote, sire 1, shared only one common allele with the offspring; the likelihood ratio of sire 1 calculates to 0.5, making him the least likely true parent. Sire 2 carries two copies of the allele and thus his likelihood ratio becomes two times greater than sire 1 at 1.0. Sire 3 happens to share the rare allele and carries two copies of it as a homozygote. His likelihood ratio calculates to 5.0; this makes his likelihood ratio five times greater than that of sire 2. From the findings at this single locus, sire 3 would be considered the most likely true sire. As in this case, sires with common genotypes have a higher probability of matching the offspring by chance compared to sires with rarer genotypes and lower probabilities. The likelihood method is able to differentiate between the groups, and if the probability of the match occurring by chance is low, it is rational to conclude that the match was called because the sire in question is the true father (Blanchard, 2013; Marshall, 2007). This type of analysis aids in panels of lower SNP counts where many common alleles are shared.

LOD Scoring

To further help in parentage determinations, likelihood ratios are obtained from numerous unlinked marker loci which can be multiplied together and the natural logarithm ($\log_e$) taken. This method used by the software is known as the LOD score. A score of 0 suggests that the candidate sire is just as likely to be the true sire of the calf as any other chosen animal. If the LOD score is positive, this indicates that the candidate sire is more likely than a randomly chosen bull to be the true sire of the calf. In almost all cases, the true parent has a positive score. In the case of a negative LOD score, the

suspected parent and the offspring share an uncommon set of alleles. They may also have a set of alleles that mismatch at one or more loci. Negative LOD scores generally mean the candidate parent is less likely to be the true parent (Blanchard, 2013; Marshall, 2007; Marshall et al., 1998).

Delta (Δ)

Another measurement used by CERVUS 3.0, delta (Δ) is defined as the difference in LOD scores between the most likely parent and the animal that is called as the next-most-likely parent. This method is used for discernment between non-excluded parents or parents also known as individuals who both have a LOD score greater than zero (Blanchard, 2013; Marshall et al., 1998). This becomes very important in closely related herds for exact determination of parentage.

Error Rates

An important element that CERVUS 3.0 utilizes in its likelihood equations is error rate. This is used to account for typing errors as well as mutations and null alleles that cannot be handled using the exclusion method. In an error, substitution of a true genotype at a specific locus under Hardy–Weinberg assumptions takes place. This method provides a way to use data that may not be perfect, without excluding potential sires that may mismatch at one or more loci. In reality, these sires may be true parents. Error rates help adjust for anything that may have occurred from previously described instances, such as possibility of mutations, call errors, genotyping errors, or unknown null alleles. For this study, the software only allows room for the possibility of non-parents

being mismatched at only one or two loci and falsely being assigned parentage. Because of the high probability for error in data entry, best results are obtained when typing errors are allowed (i.e., the error rate is set at a value higher than zero) (Blanchard, 2013; Marshall, 2007; Marshall et al., 1998).

Simulations

Simulations are used by CERVUS 3.0 to analyze the statistical significance of Δ values. Before each parentage analysis is run, a simulation is performed which follows the steps of parentage analysis using allele frequencies of loci typed in the study population. Dam and sire genotypes are generated assuming Hardy–Weinberg equilibrium from the observed allele frequencies in the study group. Sampling from the group in a Mendelian fashion creates offspring genotypes. Genotypes are then created for a number of unrelated candidate parents. Next, genotypes are altered to make the simulated data more realistic according to parameters which can include numbers of candidates, proportion of candidates sampled, proportion of loci typed, and error rate. Each assumed parent is then tested as the true parent and LOD scores are recorded. Once all parents have been "simulated," the most likely and next-most-likely parents are labeled and the Δ score is determined. This simulation is carried out for a large number of offspring in order to create proper distributions of Δ (Blanchard, 2013).

Finally, essential values of Δ are determined so that the significance of the Δ scores found in an actual parentage analysis run on the population can be tested. CERVUS 3.0 makes a comparison of these distributed Δ scores for true parents who were the most likely parents against the distribution of Δ for non-parents who were deemed the

most likely parents. From this data, confidence levels, or levels of tolerance of false positives, are produced. In the case of a 95% confident paternity call, for example, the program identifies the value of Δ for which 19 out of 20 Δ values exceeding the value come from the distribution of Δ scores for the most-likely candidates that were true parents. Therefore, only 1 out of 20 Δ values that falls higher than that Δ value resulted from the distribution for most-likely candidates that were non-parents. Any candidate that has a Δ score that is higher than this critical value is then assigned parentage with 95% confidence. Confidence levels can be set by the user depending on the accuracy needed for parentage assignment (Blanchard, 2013; Marshall, 2007; Marshall et al., 1998). This allows for flexibility in calls and higher accuracy when needed.

Arguments against Likelihood

Arguments against the use of the likelihood method for parentage determination come when close relatives (full siblings of the offspring) can be considered as candidate parents. This can be very common in beef production practices when animals are kept as replacements in the same herd. Interestingly enough, when using likelihood ratios, it has been found that when no paternal information is known and full siblings of the offspring are being tested for maternity, on average, non-excluded full siblings have a higher likelihood of maternity than that of their true mother. We have found in our experience that close relatives can potentially lead to overestimation of confidence levels. It is important to be cautious when full siblings of the offspring, as well as half-siblings, can be considered as candidate parents (Blanchard, 2013). In an effort to prevent these incorrect calls, CERVUS 3.0 has a setting that can simulate relatives of the

offspring/dam/sire trio amidst the pool of candidates in order to generate more accurate critical Δ scores for determining confidence levels (Blanchard, 2013; Marshall, 2007; Marshall et al., 1998).

SireMatch

SireMatch 2.0 (Pollak, 2006) is a second software program that is used in this analysis. SireMatch 2.0 also uses the likelihood-based approach to calculate its probabilities for assumed true sires based on the genotypes of the calf, dam, and possible sires (Van Eenennaam et al., 2007). One of the chief problems with SireMatch 2.0 is that it is not as extensive of a program as CERVUS 3.0. Some of the main problems with the software are that it does not perform simulations or account for relatives or error rates. As mentioned above, these are very critical when dealing with multiple animals that could be related. Because of this, in this study, we have chosen to use it as a secondary check for results obtained primarily from CERVUS 3.0 (Blanchard, 2013).

Objectives

The objective of this study is to analytically compare the efficacy of SNP markers from the proven Fluidigim® 88-marker parentage panel (Fluidigm Corporation, 2011a) to the 25-marker parentage panel (Karniol et al., 2009) derived by Karniol et al. The comparison will evaluate if the smaller 25-marker parentage panel (Karniol et al., 2009) will be statistically relevant and reliable for determining parentage in United States cattle breed populations. If this relevance exists, the comparison will showcase a marker panel that can be used at a more economical cost to cattle producers.

Justification

In cattle ranching, production is the single greatest factor for success. Production of offspring correlates with the livelihood and income of cattle producers. Many methods have been created to measure estimated progeny differences (EPDs; a measure of genetic merit) for various traits, but in the cattle industry there are few EPDs specifically identified for selection based on reproduction (Drake & Weber, 2012). This is especially difficult from a production standpoint for cattle producers when they select which bulls to purchase and utilize in their herds. Utilizing parentage determination as a method for producers to receive important and accurate feedback on the reality of their operations helps with cost savings and efficacy. For example, on average in multi-sire breeding pastures, 7% of bulls are not producing any offspring (Drake & Weber, 2012). Without genetic parentage testing, a producer has no way of determining which bulls are siring calves.

Parentage determination also helps ensure accurate pedigree recording, which is the key to increasing genetic gain. In turn, genetic gain (which can be reflected in meat, milk, and performance) is highly linked to commercial and economic gain (University of Queensland Animal Genetics Lab, 2013). Being able to utilize SNP parentage testing would allow for better cattle production management and cost savings over time, pending that a SNP panel could be smaller and therefore more affordable.

CHAPTER 2

MATERIALS AND METHODS

Materials

In total, 3,678 bovine samples (hair, semen, or blood) were collected from three dairy farms and 6 cow/calf beef operations in the Northern Utah, Northern/Southern Idaho, and Southwestern Wyoming areas. The breeds that were represented included Holstein, Brown Swiss, Jersey, Hereford, Irish Black, Black Angus, Red Angus, Limousin, Hereford X Angus, Charolais X Angus, and Maine-Anjou. A small quantity of these samples came from known offspring/dam/sire trios and from offspring/sire or offspring/dam duos to utilize as controls to check software and other records to ensure that correct calls were being made. A significant quantity of the other collected samples were from unknown groupings of parents and offspring. For this study, 250 offspring/dam/sire trios were hand selected from the pool of 3,678 bovine samples based on 100% accuracy calls from both the SireMatch 2.0 (Pollak, 2006) and CERVUS 3.0 (Kalinowski et al., 2007; Marshall, 2007; Marshall et al., 1998) parentage software.

DNA from a minimum of 10 hair root follicles were extracted using a modified protocol derived from the Agencourt® DNAdvance™ Genomic DNA Isolation Kit ("Agencourt DNAdvance Genomic DNA Isolation Kit," 2007). The modifications were as follows: The use of 1 M Dithiothreitol (DTT) (which is used for DNA extraction on rodent tails and which was not needed for DNA extraction of the hair root follicles) was excluded. Hair follicles were cut to size and carefully placed follicle down in a 0.6 mL tube. A Lysis Master Mix was made by combining 94 μL of Lysis Buffer with 3.5 μL of

Proteinase K (40 mg/ml) for each sample. 98 μL of the prepared Lysis Master Mix was placed into each 0.6 mL tube. Tubes were placed in a shaking incubator for 15 minutes at 55°C. Upon removal from the incubator, tubes were quick-spun to collect any condensation. The lysate was then transferred into a 96-well plate. 50 μL of Bind1 buffer was then added to each well and mixed by pipetting until well mixed.

Bind2 buffer was vortexed until beads were resuspended. 85 μL was then added to each well and mixed by pipetting until well mixed. Samples were incubated at room temperature for 1 minute. The sample plate was then placed on a 96-well magnet for 4 minutes to allow for proper separation. The supernatant was aspirated and discarded while the plate was situated on the magnet.

After the supernatant was removed the plate was taken off the magnet, and three ethanol washes were performed as follows: 170 μL of 70% ethanol was pipetted into each well. Magnetic beads were resuspended and the plate was placed back on the magnet for 1 minute, or until the solution cleared. The ethanol was aspirated and discarded while the plate remained situated on the magnet.

After the third wash, with the plate still on the magnet, as much of the final ethanol wash as possible was removed. The plate was then removed from the magnet, and 30 μL of Elution Buffer was added and mixed by pipetting until well mixed. The plate was incubated at 55°C for 15 minutes then placed back on the magnet for 5 minutes, or until the solution cleared. 25 μL of extracted DNA was transferred to a clean 96-well plate and stored at −20°C for future use.

Sample concentration and purity were determined by spectrophotometer (NanoDrop® 2000, Thermo Scientific™, Wilmington, DE, USA). Samples were required

to have a minimum concentration of 10 ng/μL and a 260/280 range of 1.7–2.1. Samples were re-extracted if these conditions were not met.

Semen samples were processed as described by Sambrook and Russell (Sambrook & Russell, 2001) and Dayem et al. (Abdel Dayem, Mahmoud, Nawito, Ayoub, & Darwish, 2009) as follows: Frozen semen straws were thawed in a 37°C water bath for approximately 1 minute. Thawed semen was then transferred into a 1 mL microfuge tube and centrifuged at 5000 rpm for 5 minutes. The supernatant was discarded and 0.5 mL lysis buffer (Sambrook SNET: 20 mM Tris-HCl [pH 8], 5 mM EDTA [pH 8], 400 mM NaCl, 1% SDS, 400 μg/mL proteinase K) added, followed by an overnight incubation at 55°C with slow shaking. The sample was then extracted by adding an equal volume of phenol:chloroform:isoamyl (25:24:1), followed by vortexing for 10 seconds, then centrifuging at max speed for 2 minutes. The aqueous layer was moved to a new tube and the extraction step was repeated without phenol. The DNA was precipitated in 0.5 mL isopropanol. After washing with 70% ethanol, the samples were resuspended in 30 μL TE buffer or water, specked on the NanoDrop™, and stored as described above for other samples (Blanchard, 2013).

Blood was collected on standard FTA™ (Fast Technology for Analyses [of nucleic acids]) cards (Whatman® Inc., GE Healthcare, Florham Park, NJ, USA). The DNA was extracted following the protocol for hair follicles described above with minor modifications as follows: A 96-well plate was prepared with each well containing a blood sample from an individual animal. To obtain these samples each FTA™ card was punched with a 6-mm extraction punch. Five 6-mm punches were placed in each well. The steps were identical to the protocol for the treatment of hair follicles.

Specific Target Amplification

Specific target amplification (STA) was performed for the 88 SNP panel as described in the Fluidigm® SNPtype™ Assays for SNP Genotyping on the Dynamic Array™ IFCs protocol ("Fluidigm SNP Genotyping Analysis Software User Guide," 2011). For the 25 SNP panel, we used KBioscience's Pre-Amplification Buffer for Fluidigm® Users. The buffer was utilized specifically for preparing the DNA for genotyping using the Fluidigm system. In our research, we found that genotyping results improved greatly when the STA and pre-amplification steps were done on every sample, even when DNA concentrations exceeded the parameters (<60 ng/μL) necessitating STA.

For the 88 SNP panel, 10X SNPtype™ STA primer pools were prepared by mixing 100 μM of each SNPtype™ Assay STA Primer with 100 μM of each Assay Locus Specific Primer (LSP). DNA Suspension Buffer (10 mM Tris [pH 8], 0.1 mM EDTA) was added for a final total volume of 400 μL and final primer concentrations of 500 nM (Fluidigm Corporation, 2011b).

For the 25 SNP panel, the pre-amplification primer pool was prepared by mixing 100 μM of each KASPar™ SNP Genotyping primer with 100 μM of each constant primer (KBiosciences, 2015). DNA Suspension Buffer (10 mM Tris [pH 8], 0.1 mM EDTA) was added for a final total volume of 400 μL and final primer concentrations of 500 nM (Fluidigm Corporation, 2011c).

Next, STA Pre-Mix aliquots were made by combining Qiagen® 2X Multiplex PCR Master Mix (PN 206143, Qiagen® Inc., Valencia, CA, USA) with the prepared 10X primer pool (SNPtype™ Primer Pool or KASPar™ SNP Genotyping Primer Pool). PCR-certified water was added for a final concentration of 1X each for the Qiagen® Master

Mix and Primer Pool (SNPtype™ Primer Pool or KASPar™ SNP Genotyping Primer Pool). The Pre-Mix 1.25 μL of each genomic DNA sample was added to the Pre-Mix in each well of a 96-well PCR plate to make a final volume of 5 μL per reaction.

Thermal cycling of the samples proceeded with a 95°C hold for 15 minutes followed by 14 cycles of 95°C denaturation for 15 seconds and 60°C annealing/extension for 4 minutes. The STA products were then diluted 1:100 in DNA Suspension Buffer and stored at −20°C.

Genotyping Reaction

SNPtype™ Assay Mixes were assembled by mixing 100 μM each of SNPtype™ Assay Allele Specific Primers (ASP1/ASP2) and 100 μM SNPtype™ Assay LSP. DNA Suspension Buffer was added for a total volume of 40 μL (final concentration of ASP1/2 was 7.5 μM and of LSP was 20 μM). Following this, a 10X Assay was created for each SNP by combining 2X Assay Loading Reagent (final concentration of 1X), PCR-certified water, and the above SNPtype™ Assay Mixes (final concentration of 1X) for a total of 5 μL. A mix containing 2X Assay Loading Reagent (final concentration of 1X) mixed for a final volume of 5 μL was made to fill any unused inlets on the chip (Fluidigm Corporation, 2011b).

The 25 SNP panel samples were prepared by using a modified version of the KASPar Assay for SNP Genotyping on the 96.96 Dynamic Array IFC protocol (Fluidigm Corporation, 2011b). A Sample Pre-Mix was created that consisted of 2.5 μL of Qiagen 2X Multiplex PCR Master Mix (PN 206143), 0.6 μL of 10X STA Primers, 0.65 μL of DNase-free water, and then a total of 1.25 μL of the genomic DNA (Fluidigm

Corporation, 2011b). This brought the total sample mix to 5 μL. Two non-template controls (NTCs) were included on every chip run.

Samples for the 88 SNP panel were prepared by creating a Sample Pre-Mix which consisted of Biotium™ 2X Fast Probe Master Mix (PN 31005, Biotium Inc., Hayward, CA, USA), SNPtype™ 20X Sample Loading Reagent, SNPtype™ 60X Reagent, and ROX (PN 12223-012, Invitrogen™, Life Technologies, Grand Island, NY, USA). Water was added to create a final concentration of 1X each. 2.5 μL of the 1:100 diluted STA product discussed above was added to 3.5 μL of Sample Pre-Mix for a Sample Mix solution totaling 6 μL. Two non-template controls (NTCs) were included on every chip run (Fluidigm Corporation, 2011a).

Priming of the Dynamic Array™ IFC was completed by injecting control line fluid into the accumulators on the IFC chip. The blue protective film located on the bottom of the IFC was removed and the IFC loaded on the IFC Controller HX. The Prime (138X) script was then selected. Once primed, 4 μL of 10X Assay Mix was pipetted into each assay inlet and 5 μL of Sample Mix was pipetted into each sample inlet on the IFC. The IFC chip was again loaded on the IFC Controller HX and the Load Mix (138X) script selected. Once the reactions were loaded the IFC chip was moved to the Biomark™ HD for thermal cycling (Fluidigm Corporation, 2011a).

Biomark™ HD settings were selected as follows: Application (genotyping), passive reference (ROX), and probe types (SNPtype™-FAM and SNPtype™-HEX) for the 88 SNP panel and (FAM-MGB and VIC-MGB) for the 25 SNP panel, auto-exposure was confirmed, and the data collection run was started ("Fluidigm SNP Genotyping Analysis Software User Guide," 2011). Protocol SNPtype™ E 96x96 v1 was run as

follows: A thermal mix of 1 cycle of 70°C for 30 minutes and 25°C for 10 minutes was followed by a 5-minute hot start at 95°C. Four cycles of touchdown PCR then occurred where the first cycle consisted of 95°C for 15 seconds, 64°C (dropping 1°C per cycle to 61°C) for 45 seconds, and 72°C for 15 seconds. Thirty-four additional PCR cycles of 95°C for 15 seconds, 60°C for 45 seconds, and 72°C for 15 seconds then were run. A final cooling cycle at 20°C for 30 seconds was performed to complete the thermal cycling and capture data (Fluidigm Corporation, 2011). Once the thermal cycling was complete, the information saved on the chip was loaded onto the run file.

Genotype Calls Using Fluidigm® Software

IFC chips were analyzed using the Fluidigm® Genotyping Analysis Software. Sample and assay information was input into the software and the fluorescence data analyzed by color-coded scatter plots using the program's Auto-Call Analysis feature. The Confidence Threshold was left at the default value (65) and the Data Normalization Method set to NTC Normalization. "Analyze with Assay Reference Library" was also selected. The scatter plot data for each SNP was compared against an Assay Reference Library created in house. The Library was created by saving information from tightly clustered SNPs gathered from multiple chip runs making sure that outliers were removed. Once these clusters were gathered, genotype calls were made. After Auto-Call Analysis, each SNP scatter plot was reviewed to confirm the calls made by the software were correct. Occasionally manual calls were made to override incorrect calls by the computer (Figure 1). For instance, a sample that clustered very close to other samples given a clear call were manually converted to correspond to the call made in the nearby cluster. In the

case of extreme outliers, they were removed by selecting “Invalid.” Once the information was checked and necessary call changes made, the data was exported as a .csv file to be opened in Microsoft® Excel 2007 for further processing in preparation for use in the parentage determination software (Blanchard, 2013).

Auto-Call Analysis was performed in the Fluidigm® Genotyping Analysis Software and manual calls were made (Figure 1). The figure demonstrates examples of the genotype calls made by the software being changed manually during review after analysis. In panel a1, there are three distinct clusters, yet two clusters received the same XY (blue) call. In a2, the group in the circle has been changed manually (based on information gathered from the Assay Reference Library) to YY (green) to override the auto-call by the software. In b1, the circled YY (green) calls are not grouped close enough to the main YY cluster and are instead floating in “no-man’s-land” between the YY and XY groups. In b2, the calls in the circle have been manually converted to reflect “no-call.” Panel c1 shows a circled automatic no-call that is clustering very close to the XY group. In c2, the no-call has been changed manually to XY (blue). Rectangles in all panels indicate outliers that were marked “invalid” (orange) (Blanchard, 2013; Fluidigm Corporation, 2011a; Wang et al., 2009).

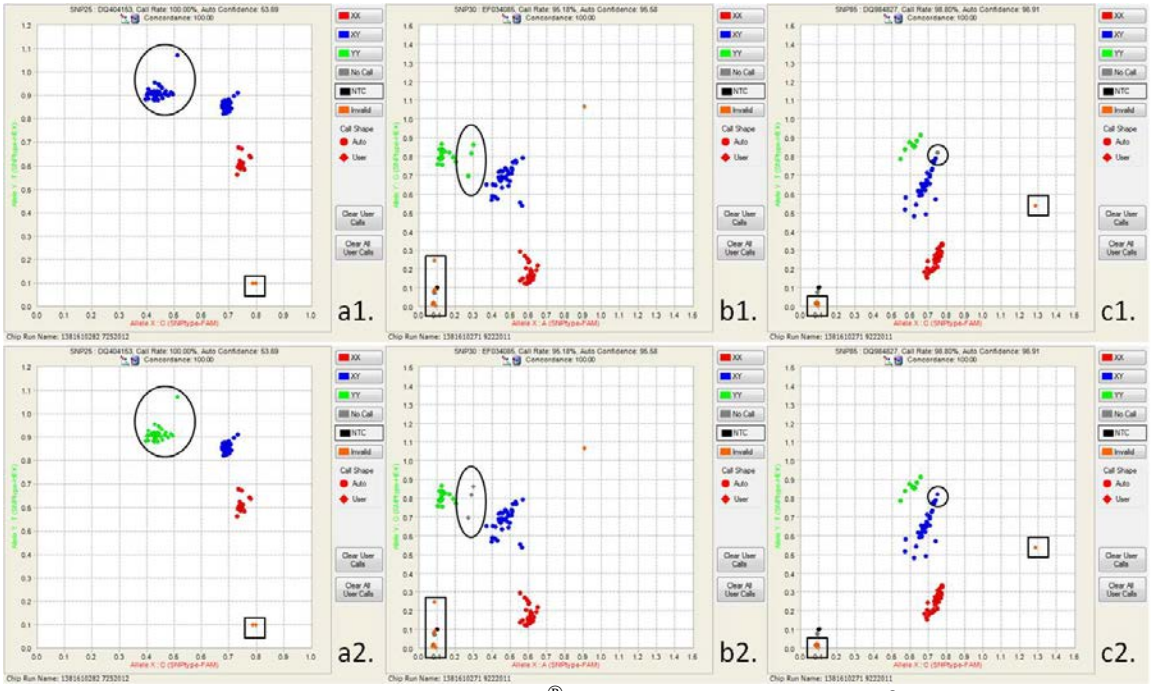

Figure 1. Manual adjustments in Fluidigm® Genotyping Analysis Software calls.

Preparation of Genotyping File

Using Excel macros created in house by Kim Blanchard (Blanchard, 2013), the exported .csv file from the Fluidigm® software was reorganized to fit the layout required by the downstream parentage determination software programs (CERVUS 3.0 and SireMatch 2.0). The use of Blanchard's macros allowed the conversion process to take just a few minutes as compared to the lengthy and tedious process of repetitive cut, copy, and paste steps needed to perform the reorganization manually for each run. The macros were easily recorded directly in Excel and broken into multiple steps to permit check points. Microsoft® Visual Basic code for these macros can be found in Appendix A and

Appendix B (Blanchard, 2013). A few brief steps described by Blanchard are listed below.

First, using macro “copy,” a copy of the .csv file is made so the original file is never altered during the conversion process. Another copy is made, this time to a new worksheet within the new book by means of the “copy2” macro. This ensures a backup is created in case errors are made. Macro “remove” deletes all the information that is unnecessary for CERVUS 3.0, leaving only the needed assay (SNP) names, sample names, and genotype calls. The “transpose” step comes next. Here information is oriented with the assay names in separate columns across the top row and sample names are shifted to the first column going down, one sample per row. The genotypes for each sample continue on the same row, one per column, corresponding to the assays listed at the top. Any genotypes labeled “No Call” in the Fluidigm® software are converted to“?:?” by the “no call” macro (Blanchard, 2013).

Next, a blank column is inserted between each assigned assay column to permit each allele to appear in a separate cell rather than in the same cell in the X:Y format as it was previously. The “loci” macro then copies the name of each SNP in the top row and pastes it into the blank cell next to it, which was created two steps previous. At this point an “x” or “y” is placed at the end of each assay name in order to distinguish between the two alleles using macro “alleles.” Blank assay columns, corresponding to non-assay containing chip inlets, are removed using macro “no assay,” and the NTC cells are deleted using the macro “ntc” (Blanchard, 2013).

Lastly, macro “csv” saves the final genotyping file in .csv format for use in the parentage determination software. One step that has to be completed manually is the

deleting of rows that will be left blank due to the removal of the “ntc” cells. This step cannot be made into a macro because their location varies depending on where they were placed on the chip. Once this is completed, the genotyping file is ready to be used for parentage determination (Blanchard, 2013).

Parentage Determination

CERVUS 3.0 (Kalinowski et al., 2007; Marshall, 2007) and SireMatch 2.0 software (Pollak, 2006) were utilized to process the genotype results for each sample group in analyzing the comparison of the 88 SNP panel and 25 SNP panel.

Software: CERVUS 3.0

CERVUS 3.0 was the first program used for parentage calls. It is accessible online (at http://www.fieldgenetics.com). The program is free for academic purposes or can be licensed for commercial use. CERVUS 3.0 works as a likelihood-based parentage determination program as described in the Review of Literature above. The software utilizes a .csv genotyping file created in Excel that was originally derived and modified from genotype calls made by the Fluidigm® software. From this, CERVUS 3.0 is able to perform a variety of analyses, including allele frequency analysis, simulation of parentage analysis, and parentage analysis as was also described above in the Review of Literature.

Allele Frequency Analysis. The analysis of allele frequency was performed by calculating the frequencies of every allele at each locus in the given tested population. Implementation of this analysis also provided a host of summary statistics including the Hardy–Weinberg equilibrium tests, percentages of null alleles, non-exclusion

probabilities, and observed and expected heterozygosities. The numbers were beneficial for determining whether or not the loci chosen in a panel were suitable for continuing on with downstream analysis.

In order for parentage to be assigned in CERVUS 3.0, a genotyping file specific to the operation, dairy, or group of animals being analyzed was selected along with "Header row" and "Read locus names." Next to "First allele in column" the number 2 was chosen, and next to "Number of loci" the number 88 was selected (for USDA 88 SNP panel screening; later, when testing the 25 SNP panel, the number was changed to reflect the 25 that were in the panel). A "summary output file" (file type .txt) name was chosen, which was automatically used for the allele frequency data file (an .alf file type used by CERVUS 3.0), and a location was specified for saving. All "Output options" (Hardy–Weinberg test and null allele frequency estimation) were selected and/or left at default settings. Computing of allele frequency analysis information was completed in under one second (Blanchard, 2013).

Simulation of Parentage Analysis. In order to estimate the resolving power of co-dominant SNP loci, CERVUS 3.0 was used to perform a simulation of parentage analysis utilizing the allele frequencies of those loci. This simulation also allowed the confidence of the parentage assignments made using the parentage analysis module (see below) to be evaluated statistically by estimating the critical values of the LOD or Δ (see Review of Literature for more details about the calculations involved).

First, a Simulation of Parentage Analysis corresponding to the type of parentage inquiry to be performed, was chosen: "Maternity," "Paternity," "Parent Pair (Sexes

Known),” or “Parent Pair (Sexes Unknown).” Next, the allele frequency file (.alf format) of interest was selected in the input file field (Blanchard, 2013).

Genotyping parameters were adjusted as follows: “Offspring” was set to 100,000; “Candidate parents” (software distinguishes between mothers and fathers) varied according to group of animals tested and consisted of all sires or dams presumed to be present at the time of breeding; “Prop. sampled” included the proportion of candidate sires or dams that were sampled. This was calculated as follows:

$$\text{Prop. Sampled} = \frac{\text{Number of total animals}}{\text{Number of total possible candidate parents}}$$

CERVUS 3.0 calculated “Prop. loci typed” during the allele frequency module, and therefore, that field was auto-filled by default upon selection of that allele frequency file in the input section of the simulation module; “Prop. loci mistyped” allows for typing error during calculations and was set to a default value of 0.01; the “Minimum typed loci” parameter was set to 1 (Blanchard, 2013).

The Output files (Summary output file [.txt] and Simulation data file [.sim]) were given names and a location to be saved to. CERVUS 3.0’s statistic Δ was used to calculate the Confidence value. Confidence levels were unchanged, resulting in defaults of “Relaxed” at 80% and “Strict” at 95%. Clicking the “Options” button followed by the “Distributions” tab, the “Generate table of Delta scores and confidence values” box was selected and the “Minimum Delta” and “Maximum Delta” values were changed to −50 and 50, respectively, allowing for ample coverage of Δ scores to be shown in the confidence table for each simulation (Blanchard, 2013).

After selecting “OK,” simulations took anywhere from minutes to over an hour to run. Run time was dependent on the number of candidate sires and dams being analyzed. In situations where simulation would take an extended period of time (hours), the number of offspring being simulated could be reduced from 100,000 to 50,000 or 10,000 to save time (Blanchard, 2013). However, the value was never set below 10,000.

Parentage Analysis. Parentage analysis was performed utilizing the Parentage Wizard. An offspring file, created by listing the sample ID name of each animal considered to be an offspring in a single column on an Excel sheet and saving it in the .csv format, was chosen. Sire and dam files were created in like manner. Inclusion of a header row in the offspring file (sometimes it was present and sometimes it was not) required that the “Includes header row” in the wizard be marked (Blanchard, 2013). The column for offspring IDs was located and its location was entered next to the “Offspring ID in column” box.

Next, the dam file (Maternity Analysis), sire file (Paternity Analysis), or both parent files (Maternal and Paternal Analysis with known sexes) were loaded into the Parentage Wizard software. (In the case of our study, all animals were of known sex, so Parentage Analysis [Sexes Unknown] was never used.) “One column for all offspring” was the parameter selected under “Candidate parent IDs appear as.” Next, the genotyping file was picked. The “Header row” box was checked and next to “ID in column” the number 1 was selected. Next to “First allele in column” the number 2 was picked. The fourth step was to choose the allele frequency file (.alf version) and simulation file (.sim) that were previously generated (Blanchard, 2013).

Lastly, output file names were chosen and destinations for the saved files were entered for the “Summary output file” (.txt) and the “Parentage data file” (.csv). Options for the output parameters were selected depending on how many results were needed for inspection; typically the “All parents” or “The two most-likely parents” choices were picked under “For each offspring include” (Blanchard, 2013). The results for the Joint LOD score were sorted. Finally, the box next to “Include non-exclusion probabilities” was selected. Upon completion of all five steps of the Parentage Wizard, the prompted “OK” box was clicked and the Parentage Analysis was initiated. The analysis was usually completed by the program in under one minute (Blanchard, 2013).

Software: SireMatch 2.0

In order to utilize the SireMatch 2.0 software (Pollak, 2006), E. J. Pollak, director of the U.S. Meat Animal Research Center, Agricultural Research Service, U.S. Department of Agriculture, was contacted. He provided a copy of the software and granted permission for its use. A sample file along with a few brief instructions on how to use the program were supplied. He did not, however, provide any true technical support, and a help manual was not available.

In order to run SireMatch 2.0, a few modifications were made to the genotyping files used in CERVUS 3.0. A new column was inserted to the left of all other columns with a heading of “S/D/C,” which stood for “Sire/Dam/Calf.” An S, D, or C was entered next to each sample ID, signifying potential sires (S), potential dams (D), and the offspring/calves (C). SireMatch 2.0 was originally designed simply for paternity testing. It can, however, determine maternity if dams are placed into the system as “Sires.” Because of this limitation, the software can only perform analysis for one potential type

of parent at a time, and two separate runs had to be made for groups requiring paternity and maternity analysis. To perform maternity analysis, the S and D notations were simply switched to “trick” SireMatch 2.0 into computing parentage analysis for the dams. The data for all were combined and saved in an .xlsx format (Blanchard, 2013).

The genotyping file was opened in SireMatch 2.0 and settings on the opening screen were amended as follows: “One header row” was chosen; all columns were set to “Marker” using the button provided; the first column heading was changed to “S/D/C” and the second column heading was switched to “ID” (all other columns remained “Marker”); “Misread or missing marker data” was identified as “?”. On the next screen, all markers were classified as SNPs by selecting the “All SNPs” button. Clicking “Next” initiated calculation of allele frequencies. On the next screen, the parameter for exclusions was set to 1 and toggled on, as was the option for allowing exclusion error (Blanchard, 2013).

Finally, the results were provided. The “Most-likely Sire” report was used to determine which animals were assigned parentage to the calves. Also useful was the “Exclusions” report which showed a grid of calves down one side and potential parents across the top, with the number of exclusions between each pairing in the body of the table. SireMatch 2.0 parentage analysis took just a few minutes from start to finish (Blanchard, 2013).

CHAPTER 3

RESULTS AND CONCLUSIONS

A total of 3,678 bovine samples (hair, semen, or blood) were collected from 3 dairy farms and 6 cow/calf beef operations (which totaled 1,637 cows, 229 bulls, and 1,796 calves and included 16 females that were considered both dams and offspring [multigenerational] because of herd retention) in the Northern Utah, Northern/Southern Idaho, and Southwestern Wyoming areas (Table 4). The samples were tested as described in the Materials and Methods section.

Table 4. *Cattle Numbers Breakdown*

	Dairies	Cow/Calf Operation	Total
Bulls	11	218	229
Cows	147	1490	1637
Calves	199	1597	1796
Total	373[a]	3305	3678

[A]Total includes 16 females that were considered both dams and offspring (multigenerational) because of herd retention (contained 4 generations) (Blanchard, 2013).

This sample pool was tested with the proven 88 SNP panel using the protocol as described in the Materials and Methods section (Table 5). From this test, 250 offspring/dam/sire trios were hand selected from the pool based on the merit of 100%

accuracy of the calls from both the SireMatch 2.0 (Pollak, 2006) and CERVUS 3.0 (Kalinowski et al., 2007; Marshall, 2007; Marshall et al., 1998) parentage software.

The 250 trios were tested using the 25 SNP panel as outlined in the Materials and Methods section. It was discovered that one of the SNPs in the 25-plex panel repeatedly produced unreliable clustering patterns with multiple samples receiving null calls. In order to assess the effects of this SNP on parentage assignment, parentage calling software runs were performed with (25 SNP panel) and without this SNP (24 SNP panel). Also, 9 samples resulted in excessive null calls and were therefore disregarded (along with their companions in the trio) in the final statistical analysis. The final results were arranged in an .xlsx file with the key (original and proven 100% calls as determined by the 88 SNP panel) placed on the left (Table 6). The information provided in this section included the ID numbers of the associated sire, dam, and offspring trios.

Table 5. *Portion of 88 SNP Panel Key*

Key		
Calf	**Sire**	**Dam**
1154	6	1154D
1324	813	1324D
1325	91	1325D
1328	93	1328D
1332	813	1332D
1335	813	1335D
1336	97	1336D
1340	2	1340D

Note. This is a sample of the 241 trio options that were tested.

24- and 25-Plex SNP panel results were compared to the 88 SNP key with parentage assignments, percentage confidence, and exclusions (mismatches) displayed. These were reported for dam, sire, and offspring trios (Tables 6–8). The lines were color coded according to correct and incorrect calls. Correct calls were left white (no color) while incorrect calls were highlighted light pink. Purple was utilized for an incorrect call that was not a match to the 88 SNP Panel key.

The results obtained from the analytical comparison were very different from our hypothesis. Within the 241 animal trios tested, 241 calves, 241 dams, and 31 bulls were utilized. The 24-SNP panel indicated 27 dams that matched perfectly for each calf sample. For the sires, at least four different sires matched perfectly in zero exclusions for each calf. It is important to note that even though four seems considerably lower, it is still significant because of the small sample size of sires. The results for the 25 SNP panel were similar, reporting 32 dams matching perfectly for each calf sample and 5 sires per calf sample. These results clearly indicate there was not enough strength in the smaller panel to make correct calls.

When testing the statistical outcome of the calls made for the 24- and 25-plex SNP panels, the total percentages of the incorrect calls from both SireMatch 2.0 (Pollak, 2006) and CERVUS 3.0 (Kalinowski et al., 2007; Marshall, 2007; Marshall et al., 1998) were averaged together for a total percentage. The dams, reporting in relation to calls made for calves, had an average of 55.6% incorrect calls for the 25 SNP panel. Dams came in with a report of 58.9% incorrect calls when tested on the 24 SNP panel. The sire data reported a percentage of incorrect calls for the 25 SNP panel at 47.2%. For the 24

Table 6. *Organized Comparison of Dams in 24- and 25-plex SNP Panels to 88 SNP Key*

Dam	Calf	25-Panel (Sirematch)	25-Panel (Cervus)	# of Dams with Zero Exclusions	24-Panel (Sirematch)	24-Panel (Cervus)	# of Dams with Zero Exclusions
1154	1149D	12	21%	33	11	17%	41
1324	1324D	98	78%	8	98	78%	8
1325	1325D	73	52%	44	73	53%	44
1328	1328D	85	50%	44	84	51%	44
1332	1332D	80	37%	27	80	48%	38
1335	1335D	40	38%	46	40	39%	64
1336	1336D	99	72%	6	99	71%	6
1340	1340D	38	30%	53	38	30%	53

Note. Each call made in the SNP panel section has a parentage call accuracy percentage for each corresponding parentage software. Zero exclusions are expressed in numbers of animals.

Table 7. *Organized Comparison of Sires in 24- and 25-plex SNP Panels to 88 SNP Key*

Sires	Calf	25-Panel (Sirematch)	25-Panel (Cervus)	# of Sires with Zero Exclusions	24-Panel (Sirematch)	24-Panel (Cervus)	# of Sires with Zero Exclusions
6	1149D	66	87%	4	51	71%	5
813	1324D	99	98%	2	99	97%	2
91	1325D	58	90%	8	57	89%	8
93	1328D	44	77%	6	43	75%	6
813	1332D	89	94%	3	89	96%	4
813	1335D	28	72%	8	31	81%	10
97	1336D	100	100%	1	100	100%	1
2	1340D	31	70%	6	31	69%	6

Note. Each call made in the SNP panel section has a parentage call accuracy percentage for each corresponding parentage software. Zero exclusions are expressed in numbers of animals.

Table 8. *Organized Comparison of Trios (Calves, Dams, Sires) in 24- and 25-plex SNP Panels to 88 SNP Key*

Sire	Dam	Calf	24-Plex Panel Total Trio Percentage	25-Plex Panel Total Trio Percentage
6	1154	1149D	30%	30%
813	1324	1324D	38%	38%
91	1325	1325D	30%	27%
93	1328	1328D	30%	27%
813	1332	1332D	21%	30%
813	1335	1335D	21%	22%
97	1336	1336D	64%	65%
2	1340	1340D	30%	30%

Note. Each call made in the SNP panel section has a total trio accuracy percentage for each corresponding parentage panel.

SNP panel, the sires reported an average of 42.3% incorrect calls. The data for the trio (calf, dam, sire) option was also calculated to find that 40.65% of the 24 SNP panel was incorrect. In regards to the 25 SNP panel, the trios reported calls that were incorrect at 42.73%.

More specifically, for each of the comparisons, percentages were evaluated to determine the confidence level (CERVUS 3.0) and probability (SireMatch 2.0) of the calls (see Tables 9–11). It is interesting to note that less than 2% of calls were made with 100% confidence. Conversely, the confidence of calls below 80% ranged from 99.6% and down. These numbers are significant, as was mentioned in the above Materials and Methods section, as generally any calls that are made below 95% do not hold significant statistical strength.

Conclusions

The original SNP panel comparison was carried out with hopes of being able to discover a panel that would have SNP allele numbers low enough to correlate with lower associated costs. For the average cow/calf producer it was deemed unrealistic to pay the fee that was associated with the 88 SNP panel cost which could spike upwards to around $25.00 per sample (Van Eenennaam, 2012). This could easily get very expensive for even a relatively small operation.

Within the analytical comparison of the proven USDA 88 SNP panel versus the 25 SNP and 24 SNP panels, many concerns of validity and panel strength were raised. In the comparison, the study was able to determine that the smaller panel did not have

Table 9. *Display of Percentages and Confidence Levels for Dams*

Software	Sirematch (Probability)	Cervus (Confidence)	Dams	Software	Sirematch (Probability)	Cervus (Confidence)	Dams
No Calls	0	0	25 Loci	No Calls	0	0	24 Loci
% accuracy	44.40%	44.40%		% accuracy	41.10%	41.10%	
Calls = 100%	3/241 (1.2%)	0		Calls = 100%	2/241 (0.8%)	0	
Calls ≥ 95%	14/241 (5.8%)	0		Calls ≥ 95%	9/241 (3.7%)	0	
Calls ≥ 80%; < 95%	17/241 (7.05%)	4/241 (1.7%)		Calls ≥ 80%; < 95%	8/241 (3.3%)	2/241 (0.8%)	
Calls < 80%	207/241 (85.9%)	237/241 (98.3%)		Calls < 80%	222/241 (92.1%)	239/241 (99.2%)	
Calls made with error (1 mismatch)	0	0		Calls made with error (1 mismatch)	0	0	

Note: These percentages are in comparison to 100% confidence calls made by the proven 88 SNP panel.

Table 10. Display of Percentages and Confidence Levels for Sires

Software	Sirematch (Probability)	Cervus (Confidence)	Sires	Software	Sirematch (Probability)	Cervus (Confidence)	Sires
No Calls	6	0	25 Loci	No Calls	0	0	24 Loci
% accuracy	52.80%	52.80%		% accuracy	57.50%	57.70%	
Calls = 100%	17/241 (7.1%)	2/241 (0.8%)		Calls = 100%	18/241 (7.47%)	1/241 (0.4%)	
Calls ≥ 95%	27/241 (11.2%)	41/241 (17%)		Calls ≥ 95%	21/241 (8.7%)	45/241 (18.7%)	
Calls ≥ 80%; < 95%	31/241 (12.86%)	68/241 (28.2%)	1 Split Call	Calls ≥ 80%; < 95%	23/241 (9.5%)	71/241 (29.5%)	1 Split Call
Calls < 80%	166/241 (68.9%)	130/241 (53.9%)	3 Split Calls (1 involved No Calls)	Calls < 80%	179/241 (74.3%)	124/241 (51.5%)	1 Split Call
Calls made with error (1 mismatch)	0	5		Calls made with error (1 mismatch)	0	0	

Note. These percentages are in comparison to 100% confidence calls made by the proven 88 SNP panel.

Table 11. *Display of Percentages and Confidence Levels for Trios (Calves, Dams, Sires)*

Software	Cervus (Dams)	Cervus (Sires)	Cervus (Trio)	Trio (Dam, Sire, Calf)	Software	Sirematch (Probability)	Cervus (Confidence)	Cervus (Trio)	Trio (Dam, Sire, Calf)
No Calls	0	0		24 Loci	No Calls	0	0		25 Loci
% accuracy	55.60%	63.10%			% accuracy	48.97%	65.57%		
Calls = 100%	0	9/241 (3.7%)	0		Calls = 100%	0	6/241 (2.5%)	0	
Calls ≥ 95%	0	38/241 (15.8%)	0		Calls ≥ 95%	0	54/241 (22.4%)	0	
Calls ≥ 80%; < 95%	2/241 (0.8%)	57/241 (23.7%)	1/241 (0.4%)		Calls ≥ 80%; < 95%	1/241 (0.4%)	51/241 (21.2%)	1/241 (0.4%)	
Calls < 80%	239/241 (99.2%)	137/241 (56.8%)	240/241 (99.6%)		Calls < 80%	240/241 (99.6%)	130/241 (53.9%)	240/241 (99.6%)	
Calls made with error (1 mismatch)	0	0	0		Calls made with error (1 mismatch)	0	0	0	

Note. These percentages are in comparison to 100% confidence calls made by the proven 88 SNP panel.

significant strength in numbers to be able to provide results of only one sire or dam with zero exclusions. This is key to the validity of a parentage testing program because the panel should provide a large enough SNP test to be able to accurately determine the parentage of the calf.

For those samples that happened to be called, the confidence levels and probability levels averaged out to be around 49% incorrect in their accuracy. This would translate into half the herd being assigned to the correct sire or dam.

This analytical comparison identified that the current number of 121 USDA-MARC SNPs approved for parentage, traceback, and animal ID is relevant. The 88 SNP panel has been proven to have enough strength and diversity to allow for complete and accurate calls in bovine parentage analysis. Although it has been proven that SNP panels slightly less in number than the 88 have statistical power and significance (Blanchard, 2013), the 25 SNP panel was proven to be weak and incapable of providing enough diversity to make distinct and accurate parentage calls for cattle.

REFERENCES

Abdel Dayem, A. M. H., Mahmoud, K. G. M., Nawito, M. F., Ayoub, M. M., & Darwish, S. F. (2009). Genotyping of kappa-casein gene in Egyptian buffalo bulls. *Livestock Science*, *122*(2–3), 286–289.

Adams, J. (2008). Paternity testing: Blood types and DNA. *Nature Education*, *1*(1), 146. Retrieved from http://www.nature.com/scitable/topicpage/paternity-testing-blood-types-and-dna-374

Agencourt DNAdvance Genomic DNA Isolation Kit. (2007). Beverly, MA: Beckman Coulter.

Anderson, E. C., & Garza, J. C. (2006). The power of single-nucleotide polymorphisms for large-scale parentage inference. *Genetics*, *172*(4), 2567–82. http://doi.org/10.1534/genetics.105.048074

Baruch, E., & Weller, J. I. (2008). Estimation of the number of SNP genetic markers required for parentage verification. *Animal Genetics*, *39*(2004), 474–479. http://doi.org/10.1111/j.1365-2052.2008.01754.x

Blanchard, K. (2013). *Analysis of single nucleotide polymorphism panels for bovine DNA identification [*master's thesis*]*. Utah State University. Retrieved from http://digitalcommons.usu.edu/etd/1712/

Brookes, A. J. (1999). The essence of SNPs. *Gene*, *234*, 177–186. http://doi.org/10.1016/S0378-1119(99)00219-X

Brumfield, R. T., Beerli, P., Nickerson, D. A., & Edwards, S. V. (2003). The utility of single nucleotide polymorphisms in inferences of population history. *Trends in Ecology and Evolution*, *18*(5), 249–256. http://doi.org/10.1016/S0169-5347(03)00018-1

Calafell, F., Shuster, A., Speed, W. C., Kidd, J. R., & Kidd, K. K. (1998). Short tandem repeat polymorphism evolution in humans. *European Journal of Human Genetics*, *6*, 38–49. http://doi.org/10.1038/sj.ejhg.5200151

Campbell, A. M. (2001). RFLP Method. Retrieved from http://www.bio.davidson.edu/genomics/method/RFLP.html

Chakraborty, R., Stivers, D. N., Su, B., Zhong, Y., & Budowle, B. (1999). The utility of short tandem repeat loci beyond human identification: Implications for development of new DNA typing systems. *Electrophoresis*, *20*, 1682–1696.

http://doi.org/10.1002/(SICI)1522-2683(19990101)20:8<1682::AID-ELPS1682>3.0.CO;2-Z

Chen, X., & Sullivan, P. F. (2003). Single nucleotide polymorphism genotyping: Biochemistry, protocol, cost and throughput. *The Pharmacogenomics Journal*, *3*, 77–96. http://doi.org/10.1038/sj.tpj.6500167

Chial, H. (n.d.). DNA fingerprinting using amplified fragment length polymorphisms (AFLP): No genome sequence required. Retrieved from http://www.nature.com/scitable/topicpage/dna-fingerprinting-using-amplified-fragment-length-polymorphisms-39051

Collins, F. S., Brooks, L. D., & Chakravarti, A. (1998). A DNA polymorphism discovery resource for research on human genetic variation. *Genome Research*, *8*, 1229–1231. http://doi.org/10.1101/gr.8.12.1229

Collins, F. S., Guyer, M. S., & Chakravarti, A. (1997). Variations on a theme: Cataloging human DNA sequence variation. *Science*, *278*(5343), 1580–1581. http://doi.org/10.1126/science.278.5343.1580

Crawford, M. H., & Beaty, K. G. (2013). DNA fingerprinting in anthropological genetics: Past, present, future. *Investigative Genetics*, *4*(1), 23. http://doi.org/10.1186/2041-2223-4-23

Deisseroth, A., Nienhuist, A., Lawrence, J., Gilest, R., Turnert, P., & Ruddle, F. H. (1978). Chromosomal localization of human beta globin gene on human chromosome 11 in somatic cell hybrids. *Genetics*, *75*(3), 1456–1460. http://doi.org/10.1073/pnas.75.3.1456

Dodds, K., Tate, M., & McEwan, J. (1996). Exclusion probabilities for pedigree testing farm animals. *Theoretical and Applied Genetics*, *92*(8), 966–975.

Drake, D. J., & Weber, K. L. (2012). What are herd bulls accomplishing in multiple sire breeding pastures? Retrieved from http://animalscience.ucdavis.edu/animalbiotech

East Midlands Forensic Pathology Unit. (n.d.). History of DNA profiling — University of Leicester. Retrieved from http://www2.le.ac.uk/departments/emfpu/genetics/explained/profiling-history

Edwards, A. (1999, November). *Likelihood*. Guanajuato, Mexico: CIMAT.

Ellegren, H. (2004). Microsatellites: Simple sequences with complex evolution. *Nature Reviews. Genetics*, *5*, 435–445. http://doi.org/10.1038/nrg1348

Fluidigm Corporation. (2011a). High-throughput SNP genotyping and parentage verification for North American beef and dairy cattle using a 96-SNP parentage, traceback, and animal ID panel. Retrieved from http://www.fluidigm.com/home/fluidigm/Posters/PAG2011_Cattle_Parentage.pdf

Fluidigm Corporation. (2011b). KASPar Assays for SNP Genotyping on the 96.96 Dynamic Array IFC. South San Fransico, CA.

Fluidigm Corporation. (2011c). KBioscience Pre-Amplification Buffer for Fluidigm Users. South San Fransico, CA.

Fluidigm Corporation. (2014). *Using the C1^{TM} Single-Cell Auto Prep System to Generate Single-Cell Libraries for DNA Sequencing [protocol]*. South San Francisco, CA.

Fluidigm SNP Genotyping Analysis Software User Guide. (2011). South San Francisco, CA: Fluidigm Corporation.

GenBank Home. (2015). Retrieved from http://www.ncbi.nlm.nih.gov/genbank/

Gomez-Raya, L., Priest, K., Rauw, W. M., Okomo-Adhiambo, M., Thain, D., Bruce, B., … Beattie, C. W. (2008). The value of DNA paternity identification in beef cattle: Examples from Nevada's free-range ranches. *Journal of Animal Science*, *86*(1), 17–24. http://doi.org/10.2527/jas.2007-0068

Gusella, J. F. (1986). DNA polymorphism and human disease. *Annual Review of Biochemistry*, *55*, 831–854. http://doi.org/10.1146/annurev.bi.55.070186.004151

Hardenbol, P., Yu, F., Belmont, J., MacKenzie, J., Bruckner, C., Brundage, T., … Gibbs, R. A. (2005). Highly multiplexed molecular inversion probe genotyping: Over 10,000 targeted SNPs genotyped in a single tube assay. *Genome Research*, *15*, 269–275. http://doi.org/10.1101/gr.3185605

Hartl, D. L., & Lewontin, R. C. (1994). DNA fingerprinting. *Science*, *266*, 201; author reply 202–203. Retrieved from http://www.britannica.com/EBchecked/topic/167155/DNA-fingerprinting

Heaton, M. (2008). Parentage-based DNA traceback in beef and dairy cattle. *Animal Research*, 1–20. Retrieved from http://www.ars.usda.gov/sp2UserFiles/Place/54380570/HeatonPublications/HeatonParentage-Traceback2008o.pdf

Hill, W. G., Salisbury, B. A., & Webb, A. J. (2008). Parentage identification using single nucleotide polymorphism genotypes: Application to product tracing. *Journal of Animal Science*, *86*(10), 2508–17. http://doi.org/10.2527/jas.2007-0276

Innis, M. A., & Gelfand, D. H. (1990). Optimization of PCRs. *PCR Protocols*. Retrieved from http://a32.lehman.cuny.edu/webwurtzel/course/PCR PROTOCOLS .pdf

Jamieson, A., & Taylor, S. S. (1997). Comparisons of three probability formulae for parentage exclusion. *Animal Genetics*, *28*(6), 397–400.

Jeffreys, A. J. (1979). DNA sequence variants in the Gγ-, Aγ-, δ- and β-globin genes of man. *Cell*, *18*(1), 1–10. http://doi.org/10.1016/0092-8674(79)90348-9

Jeffreys, A. J. (2005). Commentary: Genetic fingerprinting. *Nature Medicine*, *11*(10), 1035–1039.

Jeffreys, A. J., & Flavell, R. A. (1977). A physical map of the DNA regions flanking the rabbit beta-globin gene. *Cell*, *12*(2), 429–439.

Jeffreys, A. J., Wilson, V., Thein, S. L., Weatherall, D. J., & Ponder, B. A. (1986). DNA "fingerprints" and segregation analysis of multiple markers in human pedigrees. *American Journal of Human Genetics*, *39*, 11–24.

Jones, P. (2004). DNA forensics: From RFLP to PCR-STR and beyond. *Forensic Magazine*. Retrieved from http://www.forensicmag.com/articles/2004/09/dna-forensics-rflp-pcr-str-and-beyond

Kalinowski, S. T., Taper, M. L., & Marshall, T. C. (2007). Revising how the computer program CERVUS accommodates genotyping error increases success in paternity assignment. *Molecular Ecology*, *16*(5), 1099–106. http://doi.org/10.1111/j.1365-294X.2007.03089.x

Karniol, B., Shirak, A., Baruch, E., Singrün, C., Tal, A., Cahana, A., … Seroussi, E. (2009). Development of a 25-plex SNP assay for traceability in cattle. *Animal Genetics*, *40*(3), 353–356. http://doi.org/10.1111/j.1365-2052.2008.01846.x

KBiosciences. (2015). *KASPar SNP Genotyping System and KASPar Direct real-time PCR system* (Vol. 1). Beverly, MA.

Kwok, P. Y. (2001). Methods for genotyping single nucleotide polymorphisms. *Annual Review of Genomics and Human Genetics*, *2*(11), 235–258. Retrieved from http://arjournals.annualreviews.org/doi/full/10.1146/annurev.genom.2.1.235

Langlois, B. (2005). A review on the methods of parentage and inbreeding analysis with molecular markers. In I. Bodo, L. Alderson, & B. Langlois (Eds.), *Conservation genetics of endangered horse breeds* (pp. 35–54). Wageningen, The Netherlands: Wageningen Academic Publishing.

LGC Genomics. (n.d.). KASP™ - How it Works [Video file]. Retrieved from http://my.brainshark.com/KASP-How-it-Works-737376791

Luangtongkum, T., Morishira, T. Y., Martin, L., Choi, I., Sahin, O., Morishita, T. Y., … Martin, L. (2015). Prevalence of tetracycline-resistant Campylobacter in organic broilers a during a production cycle. *Avian Diseases*, *52*(3), 487–490.

Maniatis, T., Kee, S. G., Efstratiadis, A., & Kafatos, F. C. (1976). Amplification and characterization of a β-globin gene synthesized in vitro. *Cell*, *8*(2), 163–182.

Marshall, T. C. (2007). *Cervus Help Manual*. Manhattan, NY. Field Genetics Ltd.

Marshall, T. C., Slate, J., Kruuk, L. E. B., & Pemberton, J. M. (1998). Statistical confidence for likelihood-based paternity inference in natural populations. *Molecular Ecology*, *7*(5), 639–655. http://doi.org/10.1046/j.1365-294x.1998.00374.x

Marth, G. T., Korf, I., Yandell, M. D., Yeh, R. T., Gu, Z., Zakeri, H., … Gish, W. R. (1999). A general approach to single-nucleotide polymorphism discovery. *Nature Genetics*, *23*, 452–456. http://doi.org/10.1038/70570

Moreau-Horwin, R. (2015). The discovery of DNA fingerprinting. Retrieved from http://www.dnaforensics.com/DNAFingerprinting.aspx

Morrin, R., & Boscher, M. (2012). Cattle molecular markers and parentage testing workshop. In *ISAG Conference* (pp. 1–7). Retrieved from http://www.isag.us/docs/cattlemmptest_ct.pdf

Perkel, J. (2008). Erratum: SNP genotyping: Six technologies that keyed a revolution. *Nature Methods*, *5*(5), 575–575. http://doi.org/10.1038/nmeth0608-575b

Picoult-Newberg, L., Ideker, T. E., Pohl, M. G., Taylor, S. L., Donaldson, M. A., Nickerson, D. A., & Boyce-Jacino, M. (1999). Mining SNPs from EST databases. *Genome Research*, *9*, 167–174. http://doi.org/10.1101/gr.9.2.167

Pollak, J. (2006). SireMatch [computer software]. Ithaca, NY. Cornell University.

Renkin, E. M. (1954). Filtration, diffusion, and molecular sieving through porous cellulose membranes. *Journal of General Physiology*, *38*(1), 225–243. http://doi.org/10.1085/jgp.38.2.225

Robinson, P., & Holme, J. (2011). KASP version 4.0 SNP Genotyping Manual, 1–8.

Sachidanandam, R., Weissman, D., Schmidt, S. C., Kakol, J. M., Stein, L. D., Marth, G., … Altshuler, D. (2001). A map of human genome sequence variation containing

1.42 million single nucleotide polymorphisms. *Nature*, *409*(6822), 928–933. http://doi.org/10.1038/35057149

Saiki, R. K., Gelfand, D. H., Stoffel, S., Scharf, S. J., Higuchi, R., Horn, G. T., … Erlich, H. a. (1988). Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science*, *239*(4839), 487–491. http://doi.org/10.1126/science.2448875

Sambrook, J., & Russell, D. W. (2001). Isolation of genomic DNA from mouse tails and other small samples. In *Molecular cloning: A laboratory manual* (3rd ed., p. 6.24). Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press.

STR - Forensic DNA Testing System. (n.d.). Retrieved from http://www.forensicdnacenter.com/dna-str.html

STR Analysis - Short Tandem Repeat. (n.d.). Retrieved from http://www.exploredna.co.uk/str-analysis.html

Tiwari, R. (2013). DNA microarray (DNA chips). Retrieved from http://www.slideshare.net/Haddies/dna-microarray-dna-chips

University of Queensland Animal Genetics Lab. (2013). *DNA-based parentage testing of cattle*. DNA Parentage Testing Factsheet, final version P:\DOC\FORMS\Fact Sheets\Word Documents, last modified 21 May 2013, final version.

USDA-MARC SNPS for parentage, traceback, and animal ID. (2008). Retrieved from http://louisville.edu/medschool/biochemistry/louisville-links/USDA/cattle/marc_snps_parent_trace_id.html

Van Eenennaam, A. L. (2012). Reduced SNP panels – creation, realistic expectations, and use in different livestock industries. Retrieved from animalscience.ucdavis.edu/animalbiotech

Van Eenennaam, A. L., Weaber, R. L., Drake, D. J., Penedo, M. C. T., Quaas, R. L., Garrick, D. J., & Pollak, E. J. (2007). DNA-based paternity analysis and genetic evaluation in a large, commercial cattle ranch setting. *Journal of Animal Science*, *85*(12), 3159–69. http://doi.org/10.2527/jas.2007-0284

Visscher, P. M., Woolliams, J. A., Smith, D., & Williams, J. L. (2002). Estimation of pedigree errors in the UK dairy population using microsatellite markers and the impact on selection. *Journal of Dairy Science*, *85*(9), 2368–75. http://doi.org/10.3168/jds.S0022-0302(02)74317-8

Vos, P., Hogers, R., Bleeker, M., Reijans, M., van de Lee, T., Hornes, M., … Kuiper, M. (1995). AFLP: A new technique for DNA fingerprinting. *Nucleic Acids Research, 23*(21), 4407–4414. http://doi.org/10.1093/nar/23.21.4407

Wang, J., Lin, M., Crenshaw, A., Hutchinson, A., Hicks, B., Yeager, M., Ramakrishnan, R. (2009). High-throughput single nucleotide polymorphism genotyping using nanofluidic dynamic arrays. *BMC Genomics*, *10*, 561. http://doi.org/10.1186/1471-2164-10-561

Watson, J. D., & Crick, F. H. C. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, *171*(4356), 737–738.

Weller J.I., Seroussi E. & Ron M. (2006) Estimation of the number of genetic markers required for individual animal identification accounting for genotyping errors. Animal Genetics 37, 387–9.

Werner F.A., Durstewitz G., Habermann F.A. et al. (2004) Detection and characterization of SNPs useful for ID control and parentage testing in major European dairy breeds. Animal Genetics 35, 44–9.

APPENDICES

APPENDIX A. MICROSOFT® VISUAL BASIC® MACRO CODE FOR PREPARATION OF GENOTYPING FILES FOR 88 SNP PANEL (Blanchard, 2013)

```
Sub copy()
' 1_copy Macro
    Workbooks.Add
    Windows("1381610275
macro test.csv").Activate
    Cells.Select
    Selection.copy

Windows("Book1").Activate
    ActiveSheet.Paste
    Sheets("Sheet1").Select
    Sheets("Sheet1").Name =
"copy"
End Sub

Sub copy2()
' 2_copy2 Macro
    Cells.Select
    Selection.copy
    Sheets("Sheet2").Select
    ActiveSheet.Paste
    Sheets("copy").Select

Application.CutCopyMode
= False
    ActiveCell.FormulaR1C1
= ""
    Sheets("Sheet2").Select
    Sheets("Sheet2").Name =
"remove"
End Sub

Sub remove()
' 3_remove Macro
    Rows("1:15").Select
    Selection.Delete
Shift:=xlUp
    Range("A:A").Select
    Selection.Delete
Shift:=xlToLeft
    Range("B:C").Select
    Selection.Delete
Shift:=xlToLeft
    Range("C:F").Select
    Selection.Delete
Shift:=xlToLeft
    Range("D:E").Select
    Selection.Delete
Shift:=xlToLeft
End Sub

Sub transpose()
' 4_transpose Macro
    Range("A2:C97").Select
    Selection.copy
    Sheets("Sheet3").Select
    Sheets("Sheet3").Name =
"transpose"
    Range("B1").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B2").Select
    Selection.copy
    Range("A3").Select
    ActiveSheet.Paste
    Rows("2:2").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B98:C193").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B3").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B3").Select
    Selection.copy
    Range("A4").Select
    ActiveSheet.Paste
    Rows("3:3").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B194:C289").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B4").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B4").Select
    Selection.copy
    Range("A5").Select
    ActiveSheet.Paste
    Rows("4:4").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B290:C385").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B5").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B5").Select
    Selection.copy
    Range("A6").Select
    ActiveSheet.Paste
    Rows("5:5").Select
    Selection.Delete
Shift:=xlUp
```

```
    Sheets("remove").Select
Range("B386:C481").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B6").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B6").Select
    Selection.copy
    Range("A7").Select
    ActiveSheet.Paste
    Rows("6:6").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select
Range("B482:C577").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B7").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B7").Select
    Selection.copy
    Range("A8").Select
    ActiveSheet.Paste
    Rows("7:7").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select
Range("B578:C673").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B8").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B8").Select
    Selection.copy
    Range("A9").Select
    ActiveSheet.Paste
    Rows("8:8").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select
Range("B674:C769").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B9").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B9").Select
    Selection.copy
    Range("A10").Select
    ActiveSheet.Paste
    Rows("9:9").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select
Range("B770:C865").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B10").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B10").Select
    Selection.copy
    Range("A11").Select
    ActiveSheet.Paste
    Rows("10:10").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select
Range("B866:C961").Select
    Selection.copy
    Sheets("transpose").Select
    Range("B11").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B11").Select
    Selection.copy
    Range("A12").Select
    ActiveSheet.Paste
    Rows("11:11").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select
Range("B962:C1057").Selec
t
    Selection.copy
    Sheets("transpose").Select
    Range("B12").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B12").Select
    Selection.copy
    Range("A13").Select
    ActiveSheet.Paste
    Rows("12:12").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select
Range("B1058:C1153").Sele
ct
    Selection.copy
    Sheets("transpose").Select
```

```
    Range("B13").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B13").Select
    Selection.copy
    Range("A14").Select
    ActiveSheet.Paste
    Rows("13:13").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1154:C1249").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B14").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B14").Select
    Selection.copy
    Range("A15").Select
    ActiveSheet.Paste
    Rows("14:14").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1250:C1345").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B15").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B15").Select
    Selection.copy
    Range("A16").Select
    ActiveSheet.Paste
    Rows("15:15").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1346:C1441").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B16").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B16").Select
    Selection.copy
    Range("A17").Select
    ActiveSheet.Paste
    Rows("16:16").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1442:C1537").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B17").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B17").Select
    Selection.copy
    Range("A18").Select
    ActiveSheet.Paste
    Rows("17:17").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1538:C1633").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B18").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B18").Select
    Selection.copy
    Range("A19").Select
    ActiveSheet.Paste
    Rows("18:18").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1634:C1729").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B19").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B19").Select
    Selection.copy
    Range("A20").Select
    ActiveSheet.Paste
    Rows("19:19").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1730:C1825").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B20").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B20").Select
    Selection.copy
    Range("A21").Select
    ActiveSheet.Paste
    Rows("20:20").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1826:C1921").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B21").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B21").Select
    Selection.copy
    Range("A22").Select
    ActiveSheet.Paste
    Rows("21:21").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B1922:C2017").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B22").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B22").Select
    Selection.copy
    Range("A23").Select
    ActiveSheet.Paste
    Rows("22:22").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2018:C2113").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B23").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B23").Select
    Selection.copy
    Range("A24").Select
    ActiveSheet.Paste
    Rows("23:23").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2114:C2209").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B24").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B24").Select
    Selection.copy
    Range("A25").Select
    ActiveSheet.Paste
    Rows("24:24").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2210:C2305").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B25").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B25").Select
    Selection.copy
    Range("A26").Select
    ActiveSheet.Paste
    Rows("25:25").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2306:C2401").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B26").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B26").Select
    Selection.copy
    Range("A27").Select
    ActiveSheet.Paste
    Rows("26:26").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2402:C2497").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B27").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B27").Select
    Selection.copy
    Range("A28").Select
    ActiveSheet.Paste
    Rows("27:27").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2498:C2593").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B28").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B28").Select
    Selection.copy
    Range("A29").Select
    ActiveSheet.Paste
    Rows("28:28").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2594:C2689").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B29").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B29").Select
    Selection.copy
    Range("A30").Select
    ActiveSheet.Paste
    Rows("29:29").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2690:C2785").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B30").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B30").Select
    Selection.copy
    Range("A31").Select
    ActiveSheet.Paste
    Rows("30:30").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2786:C2881").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B31").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B31").Select
    Selection.copy
    Range("A32").Select
    ActiveSheet.Paste
    Rows("31:31").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2882:C2977").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B32").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B32").Select
    Selection.copy
    Range("A33").Select
    ActiveSheet.Paste
    Rows("32:32").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B2978:C3073").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B33").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B33").Select
    Selection.copy
    Range("A34").Select
    ActiveSheet.Paste
    Rows("33:33").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3074:C3169").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B34").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B34").Select
    Selection.copy
    Range("A35").Select
    ActiveSheet.Paste
    Rows("34:34").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3170:C3265").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B35").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B35").Select
    Selection.copy
    Range("A36").Select
    ActiveSheet.Paste
    Rows("35:35").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3266:C3361").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B36").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B36").Select
    Selection.copy
    Range("A37").Select
    ActiveSheet.Paste
    Rows("36:36").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3362:C3457").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B37").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B37").Select
    Selection.copy
    Range("A38").Select
    ActiveSheet.Paste
    Rows("37:37").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3458:C3553").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B38").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B38").Select
    Selection.copy
    Range("A38").Select
    ActiveSheet.Paste
    Range("A39").Select
    ActiveSheet.Paste
    Rows("38:38").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3554:C3649").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B39").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B39").Select
    Selection.copy
    Range("A40").Select
    ActiveSheet.Paste
    Rows("39:39").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3650:C3745").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B40").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B40").Select
    Selection.copy
    Range("A41").Select
    ActiveSheet.Paste
    Rows("40:40").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3746:C3841").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B41").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B41").Select
    Selection.copy
    Range("A42").Select
    ActiveSheet.Paste
    Rows("41:41").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3842:C3937").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B42").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B42").Select
    Selection.copy
    Range("A43").Select
    ActiveSheet.Paste
    Rows("42:42").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B3938:C4033").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B43").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B43").Select
    Selection.copy
    Range("A44").Select
    ActiveSheet.Paste
    Rows("43:43").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4034:C4129").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B44").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B44").Select
    Selection.copy
    Range("A45").Select
    ActiveSheet.Paste
    Rows("44:44").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4130:C4225").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B45").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B45").Select
    Selection.copy
    Range("A46").Select
    ActiveSheet.Paste
    Rows("45:45").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4226:C4321").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B46").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B46").Select
    Selection.copy
    Range("A47").Select
    ActiveSheet.Paste
    Rows("46:46").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4322:C4417").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B47").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B47").Select
    Selection.copy
    Range("A48").Select
    ActiveSheet.Paste
    Rows("47:47").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4418:C4513").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B48").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B48").Select
    Selection.copy
    Range("A49").Select
    ActiveSheet.Paste
    Rows("48:48").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4514:C4609").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B49").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B49").Select
    Selection.copy
    Range("A50").Select
    ActiveSheet.Paste
    Rows("49:49").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4610:C4705").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B50").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B50").Select
    Selection.copy
    Range("A51").Select
    ActiveSheet.Paste
    Rows("50:50").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4706:C4801").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B51").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B51").Select
    Selection.copy
    Range("A52").Select
    ActiveSheet.Paste
    Rows("51:51").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4802:C4897").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B52").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B52").Select
    Selection.copy
    Range("A53").Select
    ActiveSheet.Paste
    Rows("52:52").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4898:C4993").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B53").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B53").Select
    Selection.copy
    Range("A54").Select
    ActiveSheet.Paste
    Rows("53:53").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B4994:C5089").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B54").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B54").Select
    Selection.copy
    Range("A55").Select
    ActiveSheet.Paste
    Rows("54:54").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5090:C5185").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B55").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B55").Select
    Selection.copy
    Range("A56").Select
    ActiveSheet.Paste
    Rows("55:55").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5186:C5281").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B56").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B56").Select
    Selection.copy
    Range("A57").Select
    ActiveSheet.Paste
    Rows("56:56").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5282:C5377").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B57").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B57").Select
    Selection.copy
    Range("A58").Select
    ActiveSheet.Paste
    Rows("57:57").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5378:C5473").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B58").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B58").Select
    Selection.copy
    Range("A59").Select
    ActiveSheet.Paste
    Rows("58:58").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5474:C5569").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B59").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B59").Select
    Selection.copy
    Range("A60").Select
    ActiveSheet.Paste
    Rows("59:59").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5570:C5665").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B60").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B60").Select
    Selection.copy
    Range("A61").Select
    ActiveSheet.Paste
    Rows("60:60").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5666:C5761").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B61").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B61").Select
    Selection.copy
    Range("A62").Select
    ActiveSheet.Paste
    Rows("61:61").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5762:C5857").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B62").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B62").Select
    Selection.copy
    Range("A63").Select
    ActiveSheet.Paste
    Rows("62:62").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5858:C5953").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B63").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B63").Select
    Selection.copy
    Range("A64").Select
    ActiveSheet.Paste
    Rows("63:63").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B5954:C6049").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B64").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B64").Select
    Selection.copy
    Range("A65").Select
    ActiveSheet.Paste
    Rows("64:64").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6050:C6145").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B65").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B65").Select
    Selection.copy
    Range("A66").Select
    ActiveSheet.Paste
    Rows("65:65").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6146:C6241").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B66").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B66").Select
    Selection.copy
    Range("A67").Select
    ActiveSheet.Paste
    Rows("66:66").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6242:C6337").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B67").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B67").Select
    Selection.copy
    Range("A68").Select
    ActiveSheet.Paste
    Rows("67:67").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6338:C6433").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B68").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B68").Select
    Selection.copy
    Range("A69").Select
    ActiveSheet.Paste
    Rows("68:68").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6434:C6529").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B69").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B69").Select
    Selection.copy
    Range("A70").Select
    ActiveSheet.Paste
    Rows("69:69").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6530:C6625").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B70").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B70").Select
    Selection.copy
    Range("A71").Select
    ActiveSheet.Paste
    Rows("70:70").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6626:C6721").Sele
ct
    Selection.copy
    Range("C6726").Select
    Sheets("transpose").Select
    Range("B71").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B71").Select
    Selection.copy
    Range("A72").Select
    ActiveSheet.Paste
    Rows("71:71").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6722:C6817").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B72").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B72").Select
    Selection.copy
    Range("A73").Select
    ActiveSheet.Paste
    Rows("72:72").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6818:C6913").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B73").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B73").Select
    Selection.copy
    Range("A74").Select
    ActiveSheet.Paste
    Rows("73:73").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B6914:C7009").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B74").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B74").Select
    Selection.copy
    Range("A75").Select
    ActiveSheet.Paste
    Rows("74:74").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7010:C7105").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B75").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B75").Select
    Selection.copy
    Range("A76").Select
    ActiveSheet.Paste
    Rows("75:75").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7106:C7201").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B76").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B76").Select
    Selection.copy
    Range("A77").Select
    ActiveSheet.Paste
    Rows("76:76").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7202:C7297").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B77").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B77").Select
    Selection.copy
    Range("A78").Select
    ActiveSheet.Paste
    Rows("77:77").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7298:C7393").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B78").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B78").Select
    Selection.copy
    Range("A79").Select
    ActiveSheet.Paste
    Rows("78:78").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7394:C7489").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B79").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B79").Select
    Selection.copy
    Range("A80").Select
    ActiveSheet.Paste
    Rows("79:79").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7490:C7585").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B80").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B80").Select
    Selection.copy
    Range("A81").Select
    ActiveSheet.Paste
    Rows("80:80").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7586:C7681").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B81").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B81").Select
    Selection.copy
    Range("A82").Select
    ActiveSheet.Paste
    Rows("81:81").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7682:C7777").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B82").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B82").Select
    Selection.copy
    Range("A83").Select
    ActiveSheet.Paste
    Rows("82:82").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7778:C7873").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B83").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B83").Select
    Selection.copy
    Range("A84").Select
    ActiveSheet.Paste
    Rows("83:83").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7874:C7969").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B84").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B84").Select
    Selection.copy
    Range("A85").Select
    ActiveSheet.Paste
    Rows("84:84").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B7970:C8065").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B85").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B85").Select
    Selection.copy
    Range("A86").Select
    ActiveSheet.Paste
    Rows("85:85").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8066:C8161").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B86").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B86").Select
    Selection.copy
    Range("A87").Select
    ActiveSheet.Paste
    Rows("86:86").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8162:C8257").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B87").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B87").Select
    Selection.copy
    Range("A88").Select
    ActiveSheet.Paste
    Rows("87:87").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8258:C8353").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B88").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B88").Select
    Selection.copy
    Range("A89").Select
    ActiveSheet.Paste
    Rows("88:88").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8354:C8449").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B89").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B89").Select
    Selection.copy
    Range("A90").Select
    ActiveSheet.Paste
    Rows("89:89").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8450:C8545").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B90").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B90").Select
    Selection.copy
    Range("A91").Select
    ActiveSheet.Paste
    Rows("90:90").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8546:C8641").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B91").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B91").Select
    Selection.copy
    Range("A92").Select
    ActiveSheet.Paste
    Rows("91:91").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8642:C8737").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B92").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B92").Select
    Selection.copy
    Range("A93").Select
    ActiveSheet.Paste
    Rows("92:92").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8738:C8833").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B93").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B93").Select
    Selection.copy
    Range("A94").Select
    ActiveSheet.Paste
    Rows("93:93").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8834:C8929").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B94").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B94").Select
    Selection.copy
    Range("A95").Select
    ActiveSheet.Paste
    Rows("94:94").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B8930:C9025").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B95").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B95").Select
    Selection.copy
    Range("A96").Select
    ActiveSheet.Paste
    Rows("95:95").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B9026:C9121").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B96").Select
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B96").Select
    Selection.copy
    Range("A97").Select
    ActiveSheet.Paste
    Rows("96:96").Select
    Selection.Delete
Shift:=xlUp
    Sheets("remove").Select

Range("B9122:C9217").Sele
ct
    Selection.copy
    Sheets("transpose").Select
    Range("B97").Select
```

```
    Selection.PasteSpecial
Paste:=xlPasteAll,
Operation:=xlNone,
SkipBlanks:= _
        False, transpose:=True
    Range("B97").Select
    Selection.copy
    Range("A98").Select
    ActiveSheet.Paste
    Rows("97:97").Select
    Selection.Delete
Shift:=xlUp
End Sub

Sub nocall()
' 5_nocall Macro
    Range("A1:CS97").Select
    Selection.copy
    Sheets.Add
After:=Sheets(Sheets.Count)
    Sheets("Sheet4").Select
    Sheets("Sheet4").Name =
"no call"
    Range("A1").Select
    ActiveSheet.Paste
    Selection.Replace
What:="No Call",
Replacement:="?:?",
LookAt:=xlPart, _

SearchOrder:=xlByRows,
MatchCase:=False,
SearchFormat:=False, _
        ReplaceFormat:=False
End Sub

Sub columns()
' 6_columns Macro
    Range("A1:CS97").Select
    Selection.copy
    Sheets.Add
After:=Sheets(Sheets.Count)
    ActiveSheet.Paste
    Sheets("Sheet5").Select
    Sheets("Sheet5").Name =
"columns"
    Rows("1:1").Select
    Selection.Insert
Shift:=xlDown,
CopyOrigin:=xlFormatFrom
LeftOrAbove
    Range("B1").Select
    ActiveCell.FormulaR1C1
= "1"
    Selection.AutoFill
Destination:=Range("B1:CS
1"), Type:=xlFillDefault
    Range("B1:CS1").Select
    Selection.DataSeries
Rowcol:=xlRows,
Type:=xlLinear,
Date:=xlDay, Step _
        :=1, Trend:=False
    Selection.copy
    Range("CT1").Select
    ActiveSheet.Paste
    Range("B1:GK98").Select

ActiveWorkbook.Worksheet
s("columns").Sort.SortFields
.Clear

ActiveWorkbook.Worksheet
s("columns").Sort.SortFields
.Add
Key:=Range("B1:GK1") _
        ,
SortOn:=xlSortOnValues,
Order:=xlAscending,
DataOption:=xlSortNormal
    With
ActiveWorkbook.Worksheet
s("columns").Sort
        .SetRange
Range("B1:GK98")
        .Header = xlYes
        .MatchCase = False
        .Orientation =
xlLeftToRight
        .SortMethod =
xlPinYin
        .Apply
    End With
    Rows("1:1").Select
    Selection.Delete
Shift:=xlUp
End Sub

Sub split()
' 7_split Macro
    Cells.Select
    Selection.copy
    Sheets.Add
After:=Sheets(Sheets.Count)
    ActiveSheet.Paste
    Sheets("Sheet6").Select
    Sheets("Sheet6").Name =
"split"
    Range("B:B").Select
    Selection.TextToColumns
Destination:=Range("B1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("D:D").Select
    Selection.TextToColumns
Destination:=Range("D1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
```

```
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=T
    Range("F:F").Select
    Selection.TextToColumns
Destination:=Range("F1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("H:H").Select
    Selection.TextToColumns
Destination:=Range("H1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("J:J").Select
    Selection.TextToColumns
Destination:=Range("J1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("L:L").Select
    Selection.TextToColumns
Destination:=Range("L1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("N:N").Select
    Selection.TextToColumns
Destination:=Range("N1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("P:P").Select
    Selection.TextToColumns
Destination:=Range("P1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("R:R").Select
    Selection.TextToColumns
Destination:=Range("R1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
```

```
TrailingMinusNumbers:=True
    Range("T:T").Select
    Selection.TextToColumns
Destination:=Range("T1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("V:V").Select
    Selection.TextToColumns
Destination:=Range("V1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("X:X").Select
    Selection.TextToColumns
Destination:=Range("X1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("Z:Z").Select
    Selection.TextToColumns
Destination:=Range("Z1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("AB:AB").Select
    Selection.TextToColumns
Destination:=Range("AB1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("AD:AD").Select
    Selection.TextToColumns
Destination:=Range("AD1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("AF:AF").Select
    Selection.TextToColumns
Destination:=Range("AF1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("AH:AH").Select
    Selection.TextToColumns
Destination:=Range("AH1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
```

```
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("AJ:AJ").Select
    Selection.TextToColumns
Destination:=Range("AJ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("AL:AL").Select
    Selection.TextToColumns
Destination:=Range("AL1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("AN:AN").Select
    Selection.TextToColumns
Destination:=Range("AN1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("AP:AP").Select
    Selection.TextToColumns
Destination:=Range("AP1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("AR:AR").Select
    Selection.TextToColumns
Destination:=Range("AR1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("AT:AT").Select
    Selection.TextToColumns
Destination:=Range("AT1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("AV:AV").Select
    Selection.TextToColumns
Destination:=Range("AV1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
```

```
TrailingMinusNumbers:=True
    Range("AX:AX").Select
    Selection.TextToColumns
Destination:=Range("AX1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("AZ:AZ").Select
    Selection.TextToColumns
Destination:=Range("AZ1"),
DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("BB:BB").Select
    Selection.TextToColumns
Destination:=Range("BB1"),
DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("BD:BD").Select
    Selection.TextToColumns
Destination:=Range("BD1"),
DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("BF:BF").Select
    Selection.TextToColumns
Destination:=Range("BF1"),
DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("BH:BH").Select
    Selection.TextToColumns
Destination:=Range("BH1"),
DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("BJ:BJ").Select
    Selection.TextToColumns
Destination:=Range("BJ1"),
DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("BL:BL").Select
    Selection.TextToColumns
Destination:=Range("BL1"),
DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote,
```

```
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("BN:BN").Select
    Selection.TextToColumns
Destination:=Range("BN1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("BP:BP").Select
    Selection.TextToColumns
Destination:=Range("BP1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("BR:BR").Select
    Selection.TextToColumns
Destination:=Range("BR1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("BT:BT").Select
    Selection.TextToColumns
Destination:=Range("BT1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("BV:BV").Select
    Selection.TextToColumns
Destination:=Range("BV1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("BX:BX").Select
    Selection.TextToColumns
Destination:=Range("BX1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("BZ:BZ").Select
    Selection.TextToColumns
Destination:=Range("BZ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
```

```
TrailingMinusNumbers:=True
    Range("CB:CB").Select
    Selection.TextToColumns
Destination:=Range("CB1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("CD:CD").Select
    Selection.TextToColumns
Destination:=Range("CD1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("CF:CF").Select
    Selection.TextToColumns
Destination:=Range("CF1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("CH:CH").Select
    Selection.TextToColumns
Destination:=Range("CH1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("CJ:CJ").Select
    Selection.TextToColumns
Destination:=Range("CJ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("CL:CL").Select
    Selection.TextToColumns
Destination:=Range("CL1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("CN:CN").Select
    Selection.TextToColumns
Destination:=Range("CN1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("CP:CP").Select
    Selection.TextToColumns
Destination:=Range("CP1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
```

```
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("CR:CR").Select
    Selection.TextToColumns
Destination:=Range("CR1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("CT:CT").Select
    Selection.TextToColumns
Destination:=Range("CT1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("CV:CV").Select
    Selection.TextToColumns
Destination:=Range("CV1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("CX:CX").Select
    Selection.TextToColumns
Destination:=Range("CX1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("CZ:CZ").Select
    Selection.TextToColumns
Destination:=Range("CZ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("DB:DB").Select
    Selection.TextToColumns
Destination:=Range("DB1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("DD:DD").Select
    Selection.TextToColumns
Destination:=Range("DD1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
```

```
TrailingMinusNumbers:=True
    Range("DF:DF").Select
    Selection.TextToColumns
Destination:=Range("DF1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("DH:DH").Select
    Selection.TextToColumns
Destination:=Range("DH1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("DJ:DJ").Select
    Selection.TextToColumns
Destination:=Range("DJ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("DL:DL").Select
    Selection.TextToColumns
Destination:=Range("DL1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("DN:DN").Select
    Selection.TextToColumns
Destination:=Range("DN1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("DP:DP").Select
    Selection.TextToColumns
Destination:=Range("DP1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("DR:DR").Select
    Selection.TextToColumns
Destination:=Range("DR1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("DT:DT").Select
    Selection.TextToColumns
Destination:=Range("DT1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
```

```
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("DV:DV").Select
    Selection.TextToColumns
Destination:=Range("DV1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("DX:DX").Select
    Selection.TextToColumns
Destination:=Range("DX1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("DZ:DZ").Select
    Selection.TextToColumns
Destination:=Range("DZ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("EB:EB").Select
    Selection.TextToColumns
Destination:=Range("EB1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("ED:ED").Select
    Selection.TextToColumns
Destination:=Range("ED1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("EF:EF").Select
    Selection.TextToColumns
Destination:=Range("EF1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("EH:EH").Select
    Selection.TextToColumns
Destination:=Range("EH1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
```

```
TrailingMinusNumbers:=True
    Range("EJ:EJ").Select
    Selection.TextToColumns
Destination:=Range("EJ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("EL:EL").Select
    Selection.TextToColumns
Destination:=Range("EL1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("EN:EN").Select
    Selection.TextToColumns
Destination:=Range("EN1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("EP:EP").Select
    Selection.TextToColumns
Destination:=Range("EP1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("ER:ER").Select
    Selection.TextToColumns
Destination:=Range("ER1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("ET:ET").Select
    Selection.TextToColumns
Destination:=Range("ET1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("EV:EV").Select
    Selection.TextToColumns
Destination:=Range("EV1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("EX:EX").Select
    Selection.TextToColumns
Destination:=Range("EX1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
```

```
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("EZ:EZ").Select
    Selection.TextToColumns
Destination:=Range("EZ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("FB:FB").Select
    Selection.TextToColumns
Destination:=Range("FB1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("FD:FD").Select
    Selection.TextToColumns
Destination:=Range("FD1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("FF:FF").Select
    Selection.TextToColumns
Destination:=Range("FF1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("FH:FH").Select
    Selection.TextToColumns
Destination:=Range("FH1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("FJ:FJ").Select
    Selection.TextToColumns
Destination:=Range("FJ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=Tr
ue
    Range("FL:FL").Select
    Selection.TextToColumns
Destination:=Range("FL1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQu
ote,
ConsecutiveDelimiter:=Fals
e, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
```

```
TrailingMinusNumbers:=True
    Range("FN:FN").Select
    Selection.TextToColumns
Destination:=Range("FN1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("FP:FP").Select
    Selection.TextToColumns
Destination:=Range("FP1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("FR:FR").Select
    Selection.TextToColumns
Destination:=Range("FR1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("FT:FT").Select
    Selection.TextToColumns
Destination:=Range("FT1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("FV:FV").Select
    Selection.TextToColumns
Destination:=Range("FV1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("FX:FX").Select
    Selection.TextToColumns
Destination:=Range("FX1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("FZ:FZ").Select
    Selection.TextToColumns
Destination:=Range("FZ1"),
DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False,
Comma:=False,
Space:=False, Other:=True,
OtherChar _
        :=":",
FieldInfo:=Array(Array(1,
1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("GB:GB").Select
    Selection.TextToColumns
Destination:=Range("GB1")
, DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
```

```
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False, Comma:=False,
Space:=False, Other:=True, OtherChar _
        :=":",
FieldInfo:=Array(Array(1, 1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("GD:GD").Select
    Selection.TextToColumns
Destination:=Range("GD1"), DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False, Comma:=False,
Space:=False, Other:=True, OtherChar _
        :=":",
FieldInfo:=Array(Array(1, 1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("GF:GF").Select
    Selection.TextToColumns
Destination:=Range("GF1"), DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False, Comma:=False,
Space:=False, Other:=True, OtherChar _
        :=":",
FieldInfo:=Array(Array(1, 1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("GH:GH").Select
    Selection.TextToColumns
Destination:=Range("GH1"), DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False, Comma:=False,
Space:=False, Other:=True, OtherChar _
        :=":",
FieldInfo:=Array(Array(1, 1), Array(2, 1)),
TrailingMinusNumbers:=True
    Range("GJ:GJ").Select
    Selection.TextToColumns
Destination:=Range("GJ1"), DataType:=xlDelimited, _

TextQualifier:=xlDoubleQuote,
ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False, Comma:=False,
Space:=False, Other:=True, OtherChar _
        :=":",
FieldInfo:=Array(Array(1, 1), Array(2, 1)),
TrailingMinusNumbers:=True
End Sub

Sub loci()
' 8_loci Macro
    Cells.Select
    Selection.copy
    Sheets.Add
After:=Sheets(Sheets.Count)
    ActiveSheet.Paste
    Sheets("Sheet7").Select
    Sheets("Sheet7").Name =
"loci"
    Range("B1").Select
    Selection.copy
    Range("C1").Select
    ActiveSheet.Paste
    Range("D1").Select
    Selection.copy
    Range("E1").Select
    ActiveSheet.Paste
    Range("F1").Select
    Selection.copy
    Range("G1").Select
    ActiveSheet.Paste
    Range("H1").Select
    Selection.copy
    Range("I1").Select
    ActiveSheet.Paste
    Range("J1").Select
    Selection.copy
    Range("K1").Select
    ActiveSheet.Paste
    Range("L1").Select
    Selection.copy
    Range("M1").Select
    ActiveSheet.Paste
    Range("N1").Select
    Selection.copy
    Range("O1").Select
    ActiveSheet.Paste
    Range("P1").Select
    Selection.copy
    Range("Q1").Select
    ActiveSheet.Paste
    Range("R1").Select
    Selection.copy
    Range("S1").Select
    ActiveSheet.Paste
    Range("T1").Select
    Selection.copy
    Range("U1").Select
```

```
ActiveSheet.Paste
Range("V1").Select
Selection.copy
Range("W1").Select
ActiveSheet.Paste
Range("X1").Select
Selection.copy
Range("Y1").Select
ActiveSheet.Paste
Range("Z1").Select
Selection.copy
Range("AA1").Select
ActiveSheet.Paste
Range("AB1").Select
Selection.copy
Range("AC1").Select
ActiveSheet.Paste
Range("AD1").Select
Selection.copy
Range("AE1").Select
ActiveSheet.Paste
Range("AF1").Select
Selection.copy
Range("AG1").Select
ActiveSheet.Paste
Range("AH1").Select
Selection.copy
Range("AI1").Select
ActiveSheet.Paste
Range("AJ1").Select
Selection.copy
Range("AK1").Select
ActiveSheet.Paste
Range("AL1").Select
Selection.copy
Range("AM1").Select
ActiveSheet.Paste
Range("AN1").Select
Selection.copy
Range("AO1").Select
ActiveSheet.Paste
Range("AP1").Select
Selection.copy
Range("AQ1").Select
ActiveSheet.Paste
Range("AR1").Select
Selection.copy
Range("AS1").Select
ActiveSheet.Paste
Range("AT1").Select
Selection.copy
Range("AU1").Select
ActiveSheet.Paste
Range("AV1").Select
Selection.copy
Range("AW1").Select
ActiveSheet.Paste
Range("AX1").Select
Selection.copy
Range("AY1").Select
ActiveSheet.Paste
Range("AZ1").Select
Selection.copy
Range("BA1").Select
ActiveSheet.Paste
Range("BB1").Select
Selection.copy
Range("BC1").Select
ActiveSheet.Paste
Range("BD1").Select
Selection.copy
Range("BE1").Select
ActiveSheet.Paste
Range("BF1").Select
Selection.copy
Range("BG1").Select
ActiveSheet.Paste
Range("BH1").Select
Selection.copy
Range("BI1").Select
ActiveSheet.Paste
Range("BJ1").Select
Selection.copy
Range("BK1").Select
ActiveSheet.Paste
Range("BL1").Select
Selection.copy
Range("BM1").Select
ActiveSheet.Paste
Range("BN1").Select
Selection.copy
Range("BO1").Select
ActiveSheet.Paste
Range("BP1").Select
Selection.copy
Range("BQ1").Select
ActiveSheet.Paste
Range("BR1").Select
Selection.copy
Range("BS1").Select
ActiveSheet.Paste
Range("BT1").Select
Selection.copy
Range("BU1").Select
ActiveSheet.Paste
Range("BV1").Select
Selection.copy
Range("BW1").Select
ActiveSheet.Paste
Range("BX1").Select
Selection.copy
Range("BY1").Select
ActiveSheet.Paste
Range("BZ1").Select
Selection.copy
Range("CA1").Select
ActiveSheet.Paste
Range("CB1").Select
Selection.copy
Range("CC1").Select
ActiveSheet.Paste
Range("CD1").Select
Selection.copy
Range("CE1").Select
ActiveSheet.Paste
Range("CF1").Select
Selection.copy
Range("CG1").Select
ActiveSheet.Paste
Range("CH1").Select
Selection.copy
Range("CI1").Select
ActiveSheet.Paste
Range("CJ1").Select
Selection.copy
```

```
Range("CK1").Select
ActiveSheet.Paste
Range("CL1").Select
Selection.copy
Range("CM1").Select
ActiveSheet.Paste
Range("CN1").Select
Selection.copy
Range("CO1").Select
ActiveSheet.Paste
Range("CP1").Select
Selection.copy
Range("CQ1").Select
ActiveSheet.Paste
Range("CR1").Select
Selection.copy
Range("CS1").Select
ActiveSheet.Paste
Range("CT1").Select
Selection.copy
Range("CU1").Select
ActiveSheet.Paste
Range("CV1").Select
Selection.copy
Range("CW1").Select
ActiveSheet.Paste
Range("CX1").Select
Selection.copy
Range("CY1").Select
ActiveSheet.Paste
Range("CZ1").Select
Selection.copy
Range("DA1").Select
ActiveSheet.Paste
Range("DB1").Select
Selection.copy
Range("DC1").Select
ActiveSheet.Paste
Range("DD1").Select
Selection.copy
Range("DE1").Select
ActiveSheet.Paste
Range("DF1").Select
Selection.copy
Range("DG1").Select
ActiveSheet.Paste
Range("DH1").Select
Selection.copy
Range("DI1").Select
ActiveSheet.Paste
Range("DJ1").Select
Selection.copy
Range("DK1").Select
ActiveSheet.Paste
Range("DL1").Select
Selection.copy
Range("DM1").Select
ActiveSheet.Paste
Range("DN1").Select
Selection.copy
Range("DO1").Select
ActiveSheet.Paste
Range("DP1").Select
Selection.copy
Range("DQ1").Select
ActiveSheet.Paste
Range("DR1").Select
Selection.copy
Range("DS1").Select
ActiveSheet.Paste
Range("DT1").Select
Selection.copy
Range("DU1").Select
ActiveSheet.Paste
Range("DV1").Select
Selection.copy
Range("DW1").Select
ActiveSheet.Paste
Range("DX1").Select
Selection.copy
Range("DY1").Select
ActiveSheet.Paste
Range("DZ1").Select
Selection.copy
Range("EA1").Select
ActiveSheet.Paste
Range("EB1").Select
Selection.copy
Range("EC1").Select
ActiveSheet.Paste
Range("ED1").Select
Selection.copy
Range("EE1").Select
ActiveSheet.Paste
Range("EF1").Select
Selection.copy
Range("EG1").Select
ActiveSheet.Paste
Range("EH1").Select
Selection.copy
Range("EI1").Select
ActiveSheet.Paste
Range("EJ1").Select
Selection.copy
Range("EK1").Select
ActiveSheet.Paste
Range("EL1").Select
Selection.copy
Range("EM1").Select
ActiveSheet.Paste
Range("EN1").Select
Selection.copy
Range("EO1").Select
ActiveSheet.Paste
Range("EP1").Select
Selection.copy
Range("EQ1").Select
ActiveSheet.Paste
Range("ER1").Select
Selection.copy
Range("ES1").Select
ActiveSheet.Paste
Range("ET1").Select
Selection.copy
Range("EU1").Select
ActiveSheet.Paste
Range("EV1").Select
Selection.copy
Range("EW1").Select
ActiveSheet.Paste
Range("EX1").Select
Selection.copy
Range("EY1").Select
ActiveSheet.Paste
Range("EZ1").Select
```

```
Selection.copy
Range("FA1").Select
ActiveSheet.Paste
Range("FB1").Select
Selection.copy
Range("FC1").Select
ActiveSheet.Paste
Range("FD1").Select
Selection.copy
Range("FE1").Select
ActiveSheet.Paste
Range("FF1").Select
Selection.copy
Range("FG1").Select
ActiveSheet.Paste
Range("FH1").Select
Selection.copy
Range("FI1").Select
ActiveSheet.Paste
Range("FJ1").Select
Selection.copy
Range("FK1").Select
ActiveSheet.Paste
Range("FL1").Select
Selection.copy
Range("FM1").Select
ActiveSheet.Paste
Range("FN1").Select
Selection.copy
Range("FO1").Select
ActiveSheet.Paste
Range("FP1").Select
Selection.copy
Range("FQ1").Select
ActiveSheet.Paste
Range("FR1").Select
Selection.copy
Range("FS1").Select
ActiveSheet.Paste
Range("FT1").Select
Selection.copy
Range("FU1").Select
ActiveSheet.Paste
Range("FV1").Select
Selection.copy
    Range("FW1").Select
    ActiveSheet.Paste
    Range("FX1").Select
    Selection.copy
    Range("FY1").Select
    ActiveSheet.Paste
    Range("FZ1").Select
    Selection.copy
    Range("GA1").Select
    ActiveSheet.Paste
    ActiveSheet.Paste
    Range("GB1").Select
    Selection.copy
    Range("GC1").Select
    ActiveSheet.Paste
    Range("GD1").Select
    Selection.copy
    Range("GE1").Select
    ActiveSheet.Paste
    Range("GF1").Select
    Selection.copy
    Range("GG1").Select
    ActiveSheet.Paste
    Range("GH1").Select
    Selection.copy
    Range("GI1").Select
    ActiveSheet.Paste
    Range("GJ1").Select
    Selection.copy
    Range("GK1").Select
    ActiveSheet.Paste
End Sub

Sub alleles()
' 9_alleles Macro
    Cells.Select
    Selection.copy
    Sheets.Add
After:=Sheets(Sheets.Count)
    ActiveSheet.Paste
    Sheets("Sheet8").Select
    Sheets("Sheet8").Name =
"alleles"
    Range("B1").Select
    ActiveCell.FormulaR1C1
= "EF034080x"
    Range("C1").Select
    ActiveCell.FormulaR1C1
= "EF034080y"
    Range("D1").Select
    ActiveCell.FormulaR1C1
= "DQ470475x"
    Range("E1").Select
    ActiveCell.FormulaR1C1
= "DQ470475y"
    Range("F1").Select
    ActiveCell.FormulaR1C1
= "DQ995977x"
    Range("G1").Select
    ActiveCell.FormulaR1C1
= "DQ995977y"
    Range("H1").Select
    ActiveCell.FormulaR1C1
= "AY844963x"
    Range("I1").Select
    ActiveCell.FormulaR1C1
= "AY844963y"
    Range("J1").Select
    ActiveCell.FormulaR1C1
= "DQ995976x"
    Range("K1").Select
    ActiveCell.FormulaR1C1
= "DQ995976y"
    Range("L1").Select
    ActiveCell.FormulaR1C1
= "EF093511x"
    Range("M1").Select
    ActiveCell.FormulaR1C1
= "EF093511y"
    Range("N1").Select
    ActiveCell.FormulaR1C1
= "AY860426x"
    Range("O1").Select
    ActiveCell.FormulaR1C1
= "AY860426y"
    Range("P1").Select
    ActiveCell.FormulaR1C1
= "EF026087x"
    Range("Q1").Select
```

```
    ActiveCell.FormulaR1C1
= "EF026087y"
    Range("R1").Select
    ActiveCell.FormulaR1C1
= "AY851162x"
    Range("S1").Select
    ActiveCell.FormulaR1C1
= "AY851162y"
    Range("T1").Select
    ActiveCell.FormulaR1C1
= "AY863214x"
    Range("U1").Select
    ActiveCell.FormulaR1C1
= "AY863214y"
    Range("V1").Select
    ActiveCell.FormulaR1C1
= "DQ837643x"
    Range("W1").Select
    ActiveCell.FormulaR1C1
= "DQ837643y"
    Range("X1").Select
    ActiveCell.FormulaR1C1
= "AY853302x"
    Range("Y1").Select
    ActiveCell.FormulaR1C1
= "AY853302y"
    Range("Z1").Select
    ActiveCell.FormulaR1C1
= "DQ984825x"
    Range("AA1").Select
    ActiveCell.FormulaR1C1
= "DQ984825y"
    Range("AB1").Select
    ActiveCell.FormulaR1C1
= "DQ647186x"
    Range("AC1").Select
    ActiveCell.FormulaR1C1
= "DQ647186y"
    Range("AD1").Select
    ActiveCell.FormulaR1C1
= "EF042091x"
    Range("AE1").Select
    ActiveCell.FormulaR1C1
= "EF042091y"
    Range("AF1").Select
    ActiveCell.FormulaR1C1
= "DQ404149x"
    Range("AG1").Select
    ActiveCell.FormulaR1C1
= "DQ404149y"
    Range("AH1").Select
    ActiveCell.FormulaR1C1
= "EF034086x"
    Range("AI1").Select
    ActiveCell.FormulaR1C1
= "EF034086y"
    Range("AJ1").Select
    ActiveCell.FormulaR1C1
= "DQ990833x"
    Range("AK1").Select
    ActiveCell.FormulaR1C1
= "DQ990833y"
    Range("AL1").Select
    ActiveCell.FormulaR1C1
= "DQ846691x"
    Range("AM1").Select
    ActiveCell.FormulaR1C1
= "DQ846691y"
    Range("AN1").Select
    ActiveCell.FormulaR1C1
= "AY842474x"
    Range("AO1").Select
    ActiveCell.FormulaR1C1
= "AY842474y"
    Range("AP1").Select
    ActiveCell.FormulaR1C1
= "DQ451555x"
    Range("AQ1").Select
    ActiveCell.FormulaR1C1
= "DQ451555y"
    Range("AR1").Select
    ActiveCell.FormulaR1C1
= "DQ916059x"
    Range("AS1").Select
    ActiveCell.FormulaR1C1
= "DQ916059y"
    Range("AT1").Select
    ActiveCell.FormulaR1C1
= "AY856094x"
    Range("AU1").Select
    ActiveCell.FormulaR1C1
= "AY856094y"
    Range("AV1").Select
    ActiveCell.FormulaR1C1
= "AY939849x"
    Range("AW1").Select
    ActiveCell.FormulaR1C1
= "AY939849y"
    Range("AX1").Select
    ActiveCell.FormulaR1C1
= "DQ404153x"
    Range("AY1").Select
    ActiveCell.FormulaR1C1
= "DQ404153y"
    Range("AZ1").Select
    ActiveCell.FormulaR1C1
= "DQ846693x"
    Range("BA1").Select
    ActiveCell.FormulaR1C1
= "DQ846693y"
    Range("BB1").Select
    ActiveCell.FormulaR1C1
= "EF093510x"
    Range("BC1").Select
    ActiveCell.FormulaR1C1
= "EF093510y"
    Range("BD1").Select
    ActiveCell.FormulaR1C1
= "AY842473x"
    Range("BE1").Select
    ActiveCell.FormulaR1C1
= "AY842473y"
    Range("BF1").Select
    ActiveCell.FormulaR1C1
= "DQ888311x"
    Range("BG1").Select
    ActiveCell.FormulaR1C1
= "DQ888311y"
    Range("BH1").Select
    ActiveCell.FormulaR1C1
= "EF034085x"
    Range("BI1").Select
    ActiveCell.FormulaR1C1
= "EF034085y"
    Range("BJ1").Select
```

```
    ActiveCell.FormulaR1C1 = "AY850194x"
    Range("BK1").Select
    ActiveCell.FormulaR1C1 = "AY850194y"
    Range("BL1").Select
    ActiveCell.FormulaR1C1 = "AY943841x"
    Range("BM1").Select
    ActiveCell.FormulaR1C1 = "AY943841y"
    Range("BN1").Select
    ActiveCell.FormulaR1C1 = "DQ647188x"
    Range("BO1").Select
    ActiveCell.FormulaR1C1 = "DQ647188y"
    Range("BP1").Select
    ActiveCell.FormulaR1C1 = "EF034083x"
    Range("BQ1").Select
    ActiveCell.FormulaR1C1 = "EF034083y"
    Range("BR1").Select
    ActiveCell.FormulaR1C1 = "EF042090x"
    Range("BS1").Select
    ActiveCell.FormulaR1C1 = "EF042090y"
    Range("BT1").Select
    ActiveCell.FormulaR1C1 = "DQ846695x"
    Range("BU1").Select
    ActiveCell.FormulaR1C1 = "DQ846695y"
    Range("BV1").Select
    ActiveCell.FormulaR1C1 = "AY942198x"
    Range("BW1").Select
    ActiveCell.FormulaR1C1 = "AY942198y"
    Range("BX1").Select
    ActiveCell.FormulaR1C1 = "DQ837646x"
    Range("BY1").Select
    ActiveCell.FormulaR1C1 = "DQ837646y"
    Range("BZ1").Select
    ActiveCell.FormulaR1C1 = "AY851163x"
    Range("CA1").Select
    ActiveCell.FormulaR1C1 = "AY851163y"
    Range("CB1").Select
    ActiveCell.FormulaR1C1 = "DQ674265x"
    Range("CC1").Select
    ActiveCell.FormulaR1C1 = "DQ674265y"
    Range("CD1").Select
    ActiveCell.FormulaR1C1 = "DQ789028x"
    Range("CE1").Select
    ActiveCell.FormulaR1C1 = "DQ789028y"
    Range("CF1").Select
    ActiveCell.FormulaR1C1 = "AY916666x"
    Range("CG1").Select
    ActiveCell.FormulaR1C1 = "AY916666y"
    Range("CH1").Select
    ActiveCell.FormulaR1C1 = "DQ404150x"
    Range("CI1").Select
    ActiveCell.FormulaR1C1 = "DQ404150y"
    Range("CJ1").Select
    ActiveCell.FormulaR1C1 = "DQ786757x"
    Range("CK1").Select
    ActiveCell.FormulaR1C1 = "DQ786757y"
    Range("CL1").Select
    ActiveCell.FormulaR1C1 = "AY841151x"
    Range("CM1").Select
    ActiveCell.FormulaR1C1 = "AY841151y"
    Range("CN1").Select
    ActiveCell.FormulaR1C1 = "DQ468384x"
    Range("CO1").Select
    ActiveCell.FormulaR1C1 = "DQ468384y"
    Range("CP1").Select
    ActiveCell.FormulaR1C1 = "DQ786761x"
    Range("CQ1").Select
    ActiveCell.FormulaR1C1 = "DQ786761y"
    Range("CR1").Select
    ActiveCell.FormulaR1C1 = "EF150946x"
    Range("CS1").Select
    ActiveCell.FormulaR1C1 = "EF150946y"
    Range("CT1").Select
    ActiveCell.FormulaR1C1 = "AY941204x"
    Range("CU1").Select
    ActiveCell.FormulaR1C1 = "AY941204y"
    Range("CV1").Select
    ActiveCell.FormulaR1C1 = "DQ500958x"
    Range("CW1").Select
    ActiveCell.FormulaR1C1 = "DQ500958y"
    Range("CX1").Select
    ActiveCell.FormulaR1C1 = "AY776154x"
    Range("CY1").Select
    ActiveCell.FormulaR1C1 = "AY776154y"
    Range("CZ1").Select
    ActiveCell.FormulaR1C1 = "DQ846690x"
    Range("DA1").Select
    ActiveCell.FormulaR1C1 = "DQ846690y"
    Range("DB1").Select
    ActiveCell.FormulaR1C1 = "EF093512x"
    Range("DC1").Select
```

```
    ActiveCell.FormulaR1C1 = "EF093512y"
    Range("DD1").Select
    ActiveCell.FormulaR1C1 = "EF089234x"
    Range("DE1").Select
    ActiveCell.FormulaR1C1 = "EF089234y"
    Range("DF1").Select
    ActiveCell.FormulaR1C1 = "DQ404151x"
    Range("DG1").Select
    ActiveCell.FormulaR1C1 = "DQ404151y"
    Range("DH1").Select
    ActiveCell.FormulaR1C1 = "DQ404152x"
    Range("DI1").Select
    ActiveCell.FormulaR1C1 = "DQ404152y"
    Range("DJ1").Select
    ActiveCell.FormulaR1C1 = "AY842472x"
    Range("DK1").Select
    ActiveCell.FormulaR1C1 = "AY842472y"
    Range("DL1").Select
    ActiveCell.FormulaR1C1 = "EF028073x"
    Range("DM1").Select
    ActiveCell.FormulaR1C1 = "EF028073y"
    Range("DN1").Select
    ActiveCell.FormulaR1C1 = "DQ832700x"
    Range("DO1").Select
    ActiveCell.FormulaR1C1 = "DQ832700y"
    Range("DP1").Select
    ActiveCell.FormulaR1C1 = "DQ650635x"
    Range("DQ1").Select
    ActiveCell.FormulaR1C1 = "DQ650635y"
    Range("DR1").Select
    ActiveCell.FormulaR1C1 = "AY919868x"
    Range("DS1").Select
    ActiveCell.FormulaR1C1 = "AY919868y"
    Range("DT1").Select
    ActiveCell.FormulaR1C1 = "DQ866817x"
    Range("DU1").Select
    ActiveCell.FormulaR1C1 = "DQ866817y"
    Range("DV1").Select
    ActiveCell.FormulaR1C1 = "DQ381153x"
    Range("DW1").Select
    ActiveCell.FormulaR1C1 = "DQ381153y"
    Range("DX1").Select
    ActiveCell.FormulaR1C1 = "AY914316x"
    Range("DY1").Select
    ActiveCell.FormulaR1C1 = "AY914316y"
    Range("DZ1").Select
    ActiveCell.FormulaR1C1 = "DQ866818x"
    Range("EA1").Select
    ActiveCell.FormulaR1C1 = "DQ866818y"
    Range("EB1").Select
    ActiveCell.FormulaR1C1 = "DQ647190x"
    Range("EC1").Select
    ActiveCell.FormulaR1C1 = "DQ647190y"
    Range("ED1").Select
    ActiveCell.FormulaR1C1 = "DQ435443x"
    Range("EE1").Select
    ActiveCell.FormulaR1C1 = "DQ435443y"
    Range("EF1").Select
    ActiveCell.FormulaR1C1 = "DQ984828x"
    Range("EG1").Select
    ActiveCell.FormulaR1C1 = "DQ984828y"
    Range("EH1").Select
    ActiveCell.FormulaR1C1 = "EF164803x"
    Range("EI1").Select
    ActiveCell.FormulaR1C1 = "EF164803y"
    Range("EJ1").Select
    ActiveCell.FormulaR1C1 = "DQ381152x"
    Range("EK1").Select
    ActiveCell.FormulaR1C1 = "DQ381152y"
    Range("EL1").Select
    ActiveCell.FormulaR1C1 = "DQ888310x"
    Range("EM1").Select
    ActiveCell.FormulaR1C1 = "DQ888310y"
    Range("EN1").Select
    ActiveCell.FormulaR1C1 = "DQ786763x"
    Range("EO1").Select
    ActiveCell.FormulaR1C1 = "DQ786763y"
    Range("EP1").Select
    ActiveCell.FormulaR1C1 = "DQ846692x"
    Range("EQ1").Select
    ActiveCell.FormulaR1C1 = "DQ846692y"
    Range("ER1").Select
    ActiveCell.FormulaR1C1 = "DQ422949x"
    Range("ES1").Select
    ActiveCell.FormulaR1C1 = "DQ422949y"
    Range("ET1").Select
    ActiveCell.FormulaR1C1 = "DQ888313x"
    Range("EU1").Select
    ActiveCell.FormulaR1C1 = "DQ888313y"
    Range("EV1").Select
```

```
    ActiveCell.FormulaR1C1
= "DQ786758x"
    Range("EW1").Select
    ActiveCell.FormulaR1C1
= "DQ786758y"
    Range("EX1").Select
    ActiveCell.FormulaR1C1
= "DQ489377x"
    Range("EY1").Select
    ActiveCell.FormulaR1C1
= "DQ489377y"
    Range("EZ1").Select
    ActiveCell.FormulaR1C1
= "DQ990835x"
    Range("FA1").Select
    ActiveCell.FormulaR1C1
= "DQ990835y"
    Range("FB1").Select
    ActiveCell.FormulaR1C1
= "DQ846689x"
    Range("FC1").Select
    ActiveCell.FormulaR1C1
= "DQ846689y"
    Range("FD1").Select
    ActiveCell.FormulaR1C1
= "DQ786759x"
    Range("FE1").Select
    ActiveCell.FormulaR1C1
= "DQ786759y"
    Range("FF1").Select
    ActiveCell.FormulaR1C1
= "EF093509x"
    Range("FG1").Select
    ActiveCell.FormulaR1C1
= "EF093509y"
    Range("FH1").Select
    ActiveCell.FormulaR1C1
= "AY849381x"
    Range("FI1").Select
    ActiveCell.FormulaR1C1
= "AY849381y"
    Range("FJ1").Select
    ActiveCell.FormulaR1C1
= "AY929334x"
    Range("FK1").Select
    ActiveCell.FormulaR1C1
= "AY929334y"
    Range("FL1").Select
    ActiveCell.FormulaR1C1
= "DQ916058x"
    Range("FM1").Select
    ActiveCell.FormulaR1C1
= "DQ916058y"
    Range("FN1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("FO1").Select
    ActiveCell.FormulaR1C1
= "y"
    Range("FP1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("FQ1").Select
    ActiveCell.FormulaR1C1
= "y"
    Range("FR1").Select
    ActiveCell.FormulaR1C1
= "AY761135x"
    Range("FS1").Select
    ActiveCell.FormulaR1C1
= "AY761135y"
    Range("FT1").Select
    ActiveCell.FormulaR1C1
= "DQ888309x"
    Range("FU1").Select
    ActiveCell.FormulaR1C1
= "DQ888309y"
    Range("FV1").Select
    ActiveCell.FormulaR1C1
= "AY857620x"
    Range("FW1").Select
    ActiveCell.FormulaR1C1
= "AY857620y"
    Range("FX1").Select
    ActiveCell.FormulaR1C1
= "DQ984827x"
    Range("FY1").Select
    ActiveCell.FormulaR1C1
= "DQ984827y"
    Range("FZ1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("GA1").Select
    ActiveCell.FormulaR1C1
= "y"
    Range("GB1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("GC1").Select
    ActiveCell.FormulaR1C1
= "y"
    Range("GD1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("GE1").Select
    ActiveCell.FormulaR1C1
= "y"
    Range("GF1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("GG1").Select
    ActiveCell.FormulaR1C1
= "y"
    Range("GH1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("GI1").Select
    ActiveCell.FormulaR1C1
= "y"
    Range("GJ1").Select
    ActiveCell.FormulaR1C1
= "x"
    Range("GK1").Select
    ActiveCell.FormulaR1C1
= "y"
End Sub

Sub noassay()
‘ 10_noassay Macro

Range("A1:GK97").Select
    Selection.copy
    Sheets.Add
After:=Sheets(Sheets.Count)
    ActiveSheet.Paste
```

```
    Sheets("Sheet9").Select      End Sub
    Sheets("Sheet9").Name =
"no assay"

Range("FN:FQ,FZ:GK").Sel
ect
    Selection.Delete
Shift:=xlToLeft
End Sub

Sub ntc()
' 11_ntc Macro
    Range("A1:FU97").Select
    Selection.copy
    Sheets.Add
After:=Sheets(Sheets.Count)
    ActiveSheet.Paste
    Sheets("Sheet10").Select
    Sheets("Sheet10").Name
= "ntc"
    Sheets("ntc").Select
    Cells.Replace
What:="NTC",
Replacement:="",
LookAt:=xlPart,
SearchOrder _
        :=xlByRows,
MatchCase:=False,
SearchFormat:=False,
ReplaceFormat:=False
End Sub

Sub csv()
' 12_csv Macro
    Cells.Select
    Selection.copy
    Workbooks.Add
    ActiveSheet.Paste
    ActiveWorkbook.SaveAs
Filename:= _
        "I:\genotyping
project\genotyping file.csv"
_
        , FileFormat:=xlCSV,
CreateBackup:=False
```

APPENDIX B. MICROSOFT® VISUAL BASIC® MACRO CODE FOR PREPARATION OF GENOTYPING FILES FOR 25 SNP PANEL (Blanchard, 2013)

The following Macro Code is the same as the Macro Code above in APPENDIX A until the alleles section is reached. At this point the code changes to format the 25 SNP Panel.

```
Sub ip_alleles()
'
' ip_alleles Macro
'

'
    Sheets.Add
After:=Sheets(Sheets.Co
unt)

Sheets("Sheet8").Select

Sheets("Sheet8").Name
= "alleles"
    Sheets("loci").Select
    Cells.Select
    Selection.copy

Sheets("alleles").Select
    ActiveSheet.Paste
    Range("B1").Select

Application.CutCopyMo
de = False

ActiveCell.FormulaR1C
1 = "W2ax"
    Range("C1").Select

ActiveCell.FormulaR1C
1 = "W2ay"
    Range("D1").Select

ActiveCell.FormulaR1C
1 = "W2bx"
    Range("E1").Select

ActiveCell.FormulaR1C
1 = "W2by"
    Range("F1").Select

ActiveCell.FormulaR1C
1 = "W2cx"
    Range("G1").Select

ActiveCell.FormulaR1C
1 = "W2cy"
    Range("H1").Select

ActiveCell.FormulaR1C
1 = "W14ax"
    Range("I1").Select

ActiveCell.FormulaR1C
1 = "W14ay"
    Range("J1").Select

ActiveCell.FormulaR1C
1 = "W14bx"
    Range("K1").Select

ActiveCell.FormulaR1C
1 = "W14by"
    Range("L1").Select

ActiveCell.FormulaR1C
1 = "W14c"

ActiveCell.FormulaR1C
1 = "W14cx"
    Range("M1").Select

ActiveCell.FormulaR1C
1 = "W14cy"
    Range("N1").Select

ActiveCell.FormulaR1C
1 = "W25ax"
    Range("O1").Select

ActiveCell.FormulaR1C
1 = "W25ax"
    Range("O1").Select

ActiveCell.FormulaR1C
1 = "W25ay"
    Range("P1").Select

ActiveCell.FormulaR1C
1 = "W25bx"
    Range("Q1").Select

ActiveCell.FormulaR1C
1 = "W25by"
    Range("R1").Select

ActiveCell.FormulaR1C
1 = "W25cx"
    Range("S1").Select

ActiveCell.FormulaR1C
1 = "W25cy"
    Range("T1").Select

ActiveCell.FormulaR1C
1 = "W38ax"
    Range("U1").Select

ActiveCell.FormulaR1C
1 = "W38ay"
    Range("V1").Select

ActiveCell.FormulaR1C
1 = "blankx"
    Range("W1").Select
```

```
ActiveCell.FormulaR1C1 = "blanky"
    Range("X1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("Y1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("Z1").Select

ActiveCell.FormulaR1C1 = "W5ax"
    Range("AA1").Select

ActiveCell.FormulaR1C1 = "W5ay"
    Range("AB1").Select

ActiveCell.FormulaR1C1 = "W5bx"
    Range("AC1").Select

ActiveCell.FormulaR1C1 = "W5by"
    Range("AD1").Select

ActiveCell.FormulaR1C1 = "W5cx"
    Range("AE1").Select

ActiveCell.FormulaR1C1 = "W5cy"
    Range("AF1").Select

ActiveCell.FormulaR1C1 = "W15ax"
    Range("AG1").Select

ActiveCell.FormulaR1C1 = "W15ay"
    Range("AH1").Select

ActiveCell.FormulaR1C1 = "W15bx"
    Range("AI1").Select

ActiveCell.FormulaR1C1 = "W15by"
    Range("AJ1").Select

ActiveCell.FormulaR1C1 = "W15cx"
    Range("AK1").Select

ActiveCell.FormulaR1C1 = "W15cy"
    Range("AL1").Select

ActiveCell.FormulaR1C1 = "W26ax"
    Range("AM1").Select

ActiveCell.FormulaR1C1 = "W26ay"
    Range("AN1").Select

ActiveCell.FormulaR1C1 = "W26bx"
    Range("AO1").Select

ActiveCell.FormulaR1C1 = "W26by"
    Range("AP1").Select

ActiveCell.FormulaR1C1 = "W26cx"
    Range("AQ1").Select

ActiveCell.FormulaR1C1 = "W26cy"
    Range("AR1").Select

ActiveCell.FormulaR1C1 = "W38bx"
    Range("AS1").Select

ActiveCell.FormulaR1C1 = "W38by"
    Range("AT1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("AU1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("AV1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("AW1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("AX1").Select

ActiveCell.FormulaR1C1 = "W6ax"
    Range("AY1").Select

ActiveCell.FormulaR1C1 = "W6ay"
    Range("AZ1").Select

ActiveCell.FormulaR1C1 = "W6bx"
    Range("BA1").Select

ActiveCell.FormulaR1C1 = "W6by"
    Range("BB1").Select

ActiveCell.FormulaR1C1 = "W6cx"
    Range("BC1").Select

ActiveCell.FormulaR1C1 = "W6cy"
    Range("BD1").Select
```

```
ActiveCell.FormulaR1C1 = "W16ax"
    Range("BE1").Select

ActiveCell.FormulaR1C1 = "W16ay"
    Range("BF1").Select

ActiveCell.FormulaR1C1 = "W16bx"
    Range("BG1").Select

ActiveCell.FormulaR1C1 = "W16by"
    Range("BH1").Select

ActiveCell.FormulaR1C1 = "W16cx"
    Range("BI1").Select

ActiveCell.FormulaR1C1 = "W16cy"
    Range("BJ1").Select

ActiveCell.FormulaR1C1 = "W27ax"
    Range("BK1").Select

ActiveCell.FormulaR1C1 = "W27ay"
    Range("BL1").Select

ActiveCell.FormulaR1C1 = "W27bx"
    Range("BM1").Select

ActiveCell.FormulaR1C1 = "W27by"
    Range("BN1").Select

ActiveCell.FormulaR1C1 = "W27cx"
    Range("BO1").Select

ActiveCell.FormulaR1C1 = "W27cy"
    Range("BP1").Select

ActiveCell.FormulaR1C1 = "W38cx"
    Range("BQ1").Select

ActiveCell.FormulaR1C1 = "W38cy"
    Range("BR1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("BS1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("BT1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("BU1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("BV1").Select

ActiveCell.FormulaR1C1 = "W7ax"
    Range("BW1").Select

ActiveCell.FormulaR1C1 = "W7ay"
    Range("BX1").Select

ActiveCell.FormulaR1C1 = "W7bx"
    Range("BY1").Select

ActiveCell.FormulaR1C1 = "W7by"
    Range("BZ1").Select

ActiveCell.FormulaR1C1 = "W7cx"
    Range("CA1").Select

ActiveCell.FormulaR1C1 = "W7cy"
    Range("CB1").Select

ActiveCell.FormulaR1C1 = "W17ax"
    Range("CC1").Select

ActiveCell.FormulaR1C1 = "W17ay"
    Range("CD1").Select

ActiveCell.FormulaR1C1 = "W17bx"
    Range("CE1").Select

ActiveCell.FormulaR1C1 = "W17by"
    Range("CF1").Select

ActiveCell.FormulaR1C1 = "W17cx"
    Range("CG1").Select

ActiveCell.FormulaR1C1 = "W17cy"
    Range("CH1").Select

ActiveCell.FormulaR1C1 = "W28ax"
    Range("CI1").Select

ActiveCell.FormulaR1C1 = "W28ay"
    Range("CJ1").Select

ActiveCell.FormulaR1C1 = "W28bx"
    Range("CK1").Select
```

```
ActiveCell.FormulaR1C1 = "W28by"
    Range("CL1").Select

ActiveCell.FormulaR1C1 = "W28cx"
    Range("CM1").Select

ActiveCell.FormulaR1C1 = "W28cy"
    Range("CN1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("CO1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("CP1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("CQ1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("CR1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("CS1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("CT1").Select

ActiveCell.FormulaR1C1 = "W18cx"
    Range("CU1").Select

ActiveCell.FormulaR1C1 = "W18cy"
    Range("CV1").Select

ActiveCell.FormulaR1C1 = "W18bx"
    Range("CW1").Select

ActiveCell.FormulaR1C1 = "W18by"
    Range("CX1").Select

ActiveCell.FormulaR1C1 = "W18ax"
    Range("CY1").Select

ActiveCell.FormulaR1C1 = "W18ay"
    Range("CZ1").Select

ActiveCell.FormulaR1C1 = "W8cx"
    Range("DA1").Select

ActiveCell.FormulaR1C1 = "W8cy"
    Range("DB1").Select

ActiveCell.FormulaR1C1 = "W8bx"
    Range("DC1").Select

ActiveCell.FormulaR1C1 = "W8by"
    Range("DD1").Select

ActiveCell.FormulaR1C1 = "W8ax"
    Range("DE1").Select

ActiveCell.FormulaR1C1 = "W8ay"
    Range("DF1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("DG1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("DH1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("DI1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("DJ1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("DK1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("DL1").Select

ActiveCell.FormulaR1C1 = "W30cx"
    Range("DM1").Select

ActiveCell.FormulaR1C1 = "W30cy"
    Range("DN1").Select

ActiveCell.FormulaR1C1 = "W30bx"
    Range("DO1").Select

ActiveCell.FormulaR1C1 = "W30by"
    Range("DP1").Select

ActiveCell.FormulaR1C1 = "W30ax"
    Range("DQ1").Select

ActiveCell.FormulaR1C1 = "W30ay"
    Range("DR1").Select
```

```
ActiveCell.FormulaR1C1 = "W19cx"
    Range("DS1").Select

ActiveCell.FormulaR1C1 = "W19cy"
    Range("DT1").Select

ActiveCell.FormulaR1C1 = "W19bx"
    Range("DU1").Select

ActiveCell.FormulaR1C1 = "W19by"
    Range("DV1").Select

ActiveCell.FormulaR1C1 = "W19ax"
    Range("DW1").Select

ActiveCell.FormulaR1C1 = "W19ay"
    Range("DX1").Select

ActiveCell.FormulaR1C1 = "W9cx"
    Range("DY1").Select

ActiveCell.FormulaR1C1 = "W9cy"
    Range("DZ1").Select

ActiveCell.FormulaR1C1 = "W9bx"
    Range("EA1").Select

ActiveCell.FormulaR1C1 = "W9by"
    Range("EB1").Select

ActiveCell.FormulaR1C1 = "W9ax"
    Range("EC1").Select

ActiveCell.FormulaR1C1 = "W9ay"
    Range("ED1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("EE1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("EF1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("EG1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("EH1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("EI1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("EJ1").Select

ActiveCell.FormulaR1C1 = "W31cx"
    Range("EK1").Select

ActiveCell.FormulaR1C1 = "W31cy"
    Range("EL1").Select

ActiveCell.FormulaR1C1 = "W31bx"
    Range("EM1").Select

ActiveCell.FormulaR1C1 = "W31by"
    Range("EN1").Select

ActiveCell.FormulaR1C1 = "W31ax"
    Range("EO1").Select

ActiveCell.FormulaR1C1 = "W31ay"
    Range("EP1").Select

ActiveCell.FormulaR1C1 = "W20cx"
    Range("EQ1").Select

ActiveCell.FormulaR1C1 = "W20cy"
    Range("ER1").Select

ActiveCell.FormulaR1C1 = "W20bx"
    Range("ES1").Select

ActiveCell.FormulaR1C1 = "W20by"
    Range("ET1").Select

ActiveCell.FormulaR1C1 = "W20ax"
    Range("EU1").Select

ActiveCell.FormulaR1C1 = "W20ay"
    Range("EV1").Select

ActiveCell.FormulaR1C1 = "W11cx"
    Range("EW1").Select

ActiveCell.FormulaR1C1 = "W11cy"
    Range("EX1").Select

ActiveCell.FormulaR1C1 = "W11bx"
    Range("EY1").Select
```

```
ActiveCell.FormulaR1C1 = "W11by"
    Range("EZ1").Select

ActiveCell.FormulaR1C1 = "W11ax"
    Range("FA1").Select

ActiveCell.FormulaR1C1 = "W11ay"
    Range("FB1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("FC1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("FD1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("FE1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("FF1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("FG1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("FH1").Select

ActiveCell.FormulaR1C1 = "W32cx"
    Range("FI1").Select

ActiveCell.FormulaR1C1 = "W32cy"
    Range("FJ1").Select

ActiveCell.FormulaR1C1 = "W32bx"
    Range("FK1").Select

ActiveCell.FormulaR1C1 = "W32by"
    Range("FL1").Select

ActiveCell.FormulaR1C1 = "W32ax"
    Range("FM1").Select

ActiveCell.FormulaR1C1 = "W32ay"
    Range("FN1").Select

ActiveCell.FormulaR1C1 = "W21cx"
    Range("FO1").Select

ActiveCell.FormulaR1C1 = "W21cy"
    Range("FP1").Select

ActiveCell.FormulaR1C1 = "W21bx"
    Range("FQ1").Select

ActiveCell.FormulaR1C1 = "W21by"
    Range("FR1").Select

ActiveCell.FormulaR1C1 = "W21ax"
    Range("FS1").Select

ActiveCell.FormulaR1C1 = "W21ay"
    Range("FT1").Select

ActiveCell.FormulaR1C1 = "W13cx"
    Range("FU1").Select

ActiveCell.FormulaR1C1 = "W13cy"
    Range("FV1").Select

ActiveCell.FormulaR1C1 = "W13bx"
    Range("FW1").Select

ActiveCell.FormulaR1C1 = "W13by"
    Range("FX1").Select

ActiveCell.FormulaR1C1 = "W13ax"
    Range("FY1").Select

ActiveCell.FormulaR1C1 = "W13ay"
    Range("FZ1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("GA1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("GB1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("GC1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("GD1").Select

ActiveCell.FormulaR1C1 = "blankx"
    Range("GE1").Select

ActiveCell.FormulaR1C1 = "blanky"
    Range("GF1").Select
```

```
ActiveCell.FormulaR1C
1 = "W35cx"
    Range("GH1").Select

ActiveCell.FormulaR1C
1 = "W35by"
    Range("GI1").Select

ActiveCell.FormulaR1C
1 = "W35bx"
    Range("GJ1").Select

ActiveCell.FormulaR1C
1 = "W35ay"
    Range("GK1").Select

ActiveCell.FormulaR1C
1 = "W35a"
    Range("GG1").Select

ActiveCell.FormulaR1C
1 = "W35cy"
    Range("GH1").Select

ActiveCell.FormulaR1C
1 = "W35bx"
    Range("GI1").Select

ActiveCell.FormulaR1C
1 = "W35by"
    Range("GJ1").Select

ActiveCell.FormulaR1C
1 = "W35ax"
    Range("GK1").Select

ActiveCell.FormulaR1C
1 = "W35ay"
    Range("GL1").Select
End Sub

Sub ip_noassay()
'
' ip_noassay Macro
'
'
    Sheets.Add
After:=Sheets(Sheets.Co
unt)

Sheets("Sheet9").Select

Sheets("Sheet9").Name
= "no assay"

Sheets("alleles").Select
    Cells.Select
    Selection.copy
    Sheets("no
assay").Select
    ActiveSheet.Paste
    Range("A1").Select

Cells.Find(What:="blan
k", After:=ActiveCell,
LookIn:=xlFormulas,
LookAt _
        :=xlPart,
SearchOrder:=xlByRow
s,
SearchDirection:=xlNex
t, MatchCase:= _
        False,
SearchFormat:=False).A
ctivate
    Range("V:Y").Select

Application.CutCopyMo
de = False
    Selection.Delete
Shift:=xlToLeft
    Range("V1").Select

Cells.FindNext(After:=
ActiveCell).Activate

Range("AP:AQ").Select
    Selection.Delete
Shift:=xlToLeft
    Selection.Delete
Shift:=xlToLeft
    Range("AP1").Select

Cells.FindNext(After:=
ActiveCell).Activate

Range("BJ:BM").Select
    Selection.Delete
Shift:=xlToLeft
    Range("BJ1").Select

Cells.FindNext(After:=
ActiveCell).Activate

Range("CB:CG").Select
    Selection.Delete
Shift:=xlToLeft
    Range("CB1").Select

Cells.FindNext(After:=
ActiveCell).Activate

Range("CN:CS").Select
    Selection.Delete
Shift:=xlToLeft
    Range("CN1").Select

Cells.FindNext(After:=
ActiveCell).Activate

Range("DF:DK").Select
    Selection.Delete
Shift:=xlToLeft
    Range("DF1").Select

Cells.FindNext(After:=
ActiveCell).Activate

Range("DX:EC").Select
    Selection.Delete
Shift:=xlToLeft
    Range("DX1").Select

Cells.FindNext(After:=
ActiveCell).Activate
```

```
Range("EP:EU").Select
    Selection.Delete
Shift:=xlToLeft
    Range("EP1").Select
End Sub
Sub ip_ntc()
'
' ip_ntc Macro
'

'
    Sheets.Add
After:=Sheets(Sheets.Co
unt)

ActiveCell.FormulaR1C
1 = ""

Sheets("Sheet10").Selec
t

Sheets("Sheet10").Name
= "ntc"
    Sheets("no
assay").Select

ActiveWindow.ScrollCo
lumn = 135

ActiveWindow.ScrollCo
lumn = 134

ActiveWindow.ScrollCo
lumn = 133

ActiveWindow.ScrollCo
lumn = 130

ActiveWindow.ScrollCo
lumn = 128

ActiveWindow.ScrollCo
lumn = 127

ActiveWindow.ScrollCo
lumn = 122

ActiveWindow.ScrollCo
lumn = 118

ActiveWindow.ScrollCo
lumn = 114

ActiveWindow.ScrollCo
lumn = 103

ActiveWindow.ScrollCo
lumn = 98

ActiveWindow.ScrollCo
lumn = 94

ActiveWindow.ScrollCo
lumn = 73

ActiveWindow.ScrollCo
lumn = 62

ActiveWindow.ScrollCo
lumn = 49

ActiveWindow.ScrollCo
lumn = 21

ActiveWindow.ScrollCo
lumn = 12

ActiveWindow.ScrollCo
lumn = 8

ActiveWindow.ScrollCo
lumn = 1
    Cells.Select
    Selection.copy
    Sheets("ntc").Select
    ActiveSheet.Paste
    Range("A1").Select

Cells.Find(What:="ntc",
After:=ActiveCell,
LookIn:=xlFormulas,
LookAt:= _
        xlPart,
SearchOrder:=xlByRow
s,
SearchDirection:=xlNex
t, MatchCase:=False _
        ,
SearchFormat:=False).A
ctivate
    Rows("2:2").Select

Application.CutCopyMo
de = False
    Selection.Delete
Shift:=xlUp
    Range("A2").Select
    Rows("25:25").Select
    Selection.Delete
Shift:=xlUp
End Sub
```