Fall 12-2003

Feasibility of Using Multiple Samples to Improve the Least Square Deconvolution Approach to Resolving 3-Person STR DNA Mixture Samples

John David White

University of Tennessee - Knoxville

Follow this and additional works at: https://trace.tennessee.edu/utk_chanhonoproj

Recommended Citation
White, John David, "Feasibility of Using Multiple Samples to Improve the Least Square Deconvolution Approach to Resolving 3-Person STR DNA Mixture Samples" (2003). University of Tennessee Honors Thesis Projects.
https://trace.tennessee.edu/utk_chanhonoproj/696

This is brought to you for free and open access by the University of Tennessee Honors Program at Trace: Tennessee Research and Creative Exchange. It has been accepted for inclusion in University of Tennessee Honors Thesis Projects by an authorized administrator of Trace: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.
UNIVERSITY HONORS PROGRAM

SENIOR PROJECT - APPROVAL

Name: John D. White

College: Engineering Department: Electrical and Computer (ECE)

Faculty Mentor: Dr. J. Douglas Birdwell and Dr. Tse wei Wang

PROJECT TITLE: Feasibility of Using Multiple Samples to Improve At Least Square Deconvolution Approach to Resolving 3-Person STR DNA Mixture Samples

I have reviewed this completed senior honors thesis with this student and certify that it is a project commensurate with honors level undergraduate research in this field.

Signed: J.D. Birdwell, Tse wei Wang, Faculty Mentor

Date: 12/5/2003

Comments (Optional):
Senior Honors Thesis
Feasibility of Using Multiple Samples to Improve the Least Square Deconvolution Approach to Resolving 3-Person STR DNA Mixture Samples

John D. White  December 5, 2003
University of Tennessee – Honors Program

Mentors:
Dr. J. Douglas Birdwell – Professor of Electrical and Computer Engineering
Dr. Tsewei Wang – Professor of Chemical Engineering
Abstract:

This document outlines a research project carried out for the fulfillment of the requirements for a senior honors thesis. The motivation for this project stems from the fact that there is currently no computational method for determining the contributors of a three-person STR DNA mixture sample. Such samples, as obtained from crime scenes, could contain incriminating evidence that could help criminal investigators to solve a variety of crimes, particularly violent crimes. A Least Squares Deconvolution (or LSD) approach currently exists to resolve two-person mixtures, but current work proves that the simple extension to three-person mixtures (LSD-3) is inadequate to make resolutions with high confidence [2]. My research explores bringing in more information to the LSD-3 method in the form of multiple samples. This new method (LSD-MS3) has been shown to dramatically increase the capabilities of LSD in some cases. While it does not address every problem associated with LSD-3, it is an exciting new step towards solving the overall problem.

This paper will outline the background of LSD needed to understand the motivation and formulation of LSD-MS3. It will also provide the results that show how LSD-MS3 improves the current methods. In addition, the complete MatLab source code of the simulation is provided as appendices.
Special thanks to Chris Lucas for letting me use the results of his thesis before it is published; to Dr. Wang and Dr. Birdwell for their guidance; and to everyone at LIT for their input and support.
Table of Contents

Abstract: .............................................................................................................................. 2
Table of Contents ................................................................................................................ 4
Table of Figures and Tables .............................................................................................. 4
Introduction ......................................................................................................................... 5
Problem Statement .............................................................................................................. 5
Background ......................................................................................................................... 5
Formulation of LSD MS-3 ................................................................................................. 9
Results and Interpretation ................................................................................................. 12
Conclusions ....................................................................................................................... 22
References ......................................................................................................................... 24
Appendix A – List of M-files and Their Descriptions ...................................................... 25
Appendix B – hasUniqueColSpace.m ............................................................................. 26
Appendix C – uniqueColSpace.m ................................................................................... 26
Appendix D – thesis.m ...................................................................................................... 27
Appendix E – plotOverallResults.m ................................................................................ 30
Appendix F – plotThesisResults.m ................................................................................... 31

Table of Figures and Tables

Figure 1 – Simulation Results: 5% Noise, Non-Degenerate Case...................................... 13
Figure 2 - Simulation Results: 10% Noise, Non-Degenerate Case ................................... 14
Figure 3 - Simulation Results: 15% Noise, Non-Degenerate Case .................................... 15
Figure 4 – Illustration of LSD-MS3 Working Despite 3 Failures in LSD-3 ..................... 16
Figure 5 - Illustration of LSD-MS3 Failing When All 3 Samples Fail in LSD-3 .......... 17
Figure 6 - Illustration of LSD-MS3 Failing When Only 1 Sample Fails in LSD-3 ........ 19
Figure 7 – Simulation Results: Degenerate Case Example 1 ........................................... 20
Figure 8 - Simulation Results: Degenerate Case Example 2 ........................................... 21
Figure 9 – Simulation Results: Degenerate Case Example 3 ........................................... 21
Introduction

As DNA evidence becomes more widely accepted in the field of forensic science, the need for DNA analysis tools grows with it. The current standard for identifying potential suspects using DNA is limited by some fundamental problems. One of these problems lies in the difficulty of determining individual contributors to DNA mixtures. Early work by Dr. Tsewei Wang at the Laboratory for Information Technologies at UT shows that a mathematical approach can be taken to DNA mixture analysis. Furthermore, she has developed an algorithm that is capable of determining the individual contributors to two-person DNA mixtures using a method called Least Square Deconvolution (LSD).

While LSD works with a high degree of confidence for two-person mixtures, the extension to three-person mixtures (LSD-3) introduces a variety of new problems that cause great uncertainty in the LSD method. The subject of my research was to explore why LSD-3 fails, and to propose a new method to improve it by adding more information in the form of multiple samples. By performing the LSD method on three different samples of the same DNA mixture, it has been shown that a great deal of confidence can be gained. This document will discuss the basic principles behind LSD, and cover the details of my research. While this method does not completely solve the problem of LSD-3, it does bring the algorithm one step closer to being a complete DNA analysis tool.

Problem Statement

By modifying the LSD-3 algorithm to perform mixture resolution on three different samples of the same STR DNA mixture at the same time, can we gain confidence in our method, and if so, how much confidence can we gain?

Background

The impact DNA has had on forensic science is well known. Popular television shows, movies, and highly publicized criminal investigations have glamorized DNA into a sort of catch all for criminal investigators. The reality is there is still a good deal of guesswork left in DNA analysis. While methods for identifying humans by their DNA profile is down to a relatively exact science, there are still some common problems associated with collecting DNA evidence.
A DNA sample found at a crime scene is called a DNA stain. At the time that field agents collect DNA evidence from the crime scene, nothing is necessarily known about the owner of the evidence. Whether it came from the suspect, the victim, or a combination of both, the sample is treated in the same manner. In DNA labs, the sample is analyzed to determine the profile of the contributor(s). It is useful to understand how DNA goes from a physical sample to the DNA profile by which we can identify humans.

The abilities of DNA profiling have grown tremendously over the last decade. Today, an exact DNA profile can be determined from samples as small as a nanogram. The sample undergoes a process called PCR, by which the physical DNA sample is amplified. A DNA sample contains many copies of a person’s DNA. PCR makes copies of the copies in order to increase the amount analysts have to work with. After PCR, the amplified DNA goes through a process of gel electrophoresis. This process puts the DNA sample in a form that a computer can read to determine the profile [4].

The profile is created by examining roughly 13 areas, or loci, of the entire DNA strand. At each locus, a human has two DNA patterns that can be recognized as alleles. These alleles can be different (heterozygous) at a locus, or they can be the same (homozygous). Without going into too much detail about how alleles are identified at each of the 13 loci, we can proceed knowing that at the end of the DNA lab analysis we know two important pieces of information: the alleles present, and the mass of each allele that is present.

Now we can begin to understand the problem of DNA mixtures. At no point in the lab analysis is there a chance to separate two DNA profiles if the original sample is a mixture. A mixture is amplified and scanned just as if it came from one person. Additionally, if all that was known was the alleles that were present in the sample, a profile could be read; but if it the sample was a mixture it would not be effective in identifying the owners of the DNA. But we also know the mass of the alleles present, which as we will see, allows us to gain more insight into the makeup of the DNA mixture.

If a DNA profile comes from a mixture, there are a number of ways in which the profiles of the contributors could combine to form the mixture profile. For instance, if a locus with two alleles is known to come from a two-person mixture, there are four
possible ways those two alleles could be found. Both people could be homozygous at their own alleles, both could be heterozygous and share the same alleles, or either one could be homozygous and the other be heterozygous. Needless to say, the analysis can get pretty tricky when we begin looking across all 13 loci having different numbers of alleles present. I could formulate this problem more fully, but instead I will pose the problem the way that LSD does, in matrix form.

I mentioned that there are many ways that two people can contribute to a mixture. In fact, if the number of alleles at a locus, and the number of contributors are known, we can determine every possible way those contributors’ DNA profiles can mix. The unique ways in which contributors can combine are called genotype combinations. Equation 1 shows how we use genotype combinations in the general formulation of LSD.

$$\begin{align*}
\begin{bmatrix}
V & S_1 & S_2 & \text{MR} \\
\end{bmatrix}
\begin{bmatrix}
x \\
pa \\
\end{bmatrix}
= b
\end{align*}$$

This is a linear algebra equation of the form $A \cdot x = b$. In English, it states that the Peak Area data, $b$, measured from the DNA sample is a linear combination of the genotype combination, $A$, and the mass ratio, $x$. The mass ratio comes from the second piece of information that we can gather from the DNA typing process. We will explore this formulation in more detail later. For now, it suffices to say that LSD works by finding the best genotype combination that matches the peak area data.

From EQ 1 we see that the columns of the genotype combination matrix belong to the contributors of the DNA sample. Once we solve for the best-fit genotype combination, it is known precisely which alleles each contributor has. Therefore the unique DNA profiles of all of the contributors can be found.

Needless to say, this formulation greatly expands the realm of possibilities for forensic DNA evidence. The success of the LSD on two-person mixtures has been published Ning Xue [1]. In general, LSD can determine to a very high degree of certainty, the makeup of a DNA mixture. In fact the LSD-2 algorithm is available through LIT in the form of a web-based program. Several organizations have begun using LSD in their own work, and have so far been impressed with its capabilities. But while two-
person mixtures are perhaps the most common types of mixtures used by these organizations, it is the three or more person mixtures that are the biggest challenge.

The size of the problem grows exponentially as more contributors are added to a mixture. A MS thesis proposal by Chris Lucas [2], to be published in 2004, began the exploration of the problem of three-person mixtures. The number of alleles at each locus grows with three people, as does the number of genotype combinations that are possible. The simple extension of LSD-2 to LSD-3 fails for many reasons. LSD-3 is plagued with degenerate cases in which it is very difficult if not impossible to determine the correct genotype resolution. Lucas has developed methods for improving the situation, such as bringing in a known profile, matrix concatenation, and mass ratio clustering. These methods help a variety of specific cases in which LSD fails, but are unable to solve every case. The focus of this research is to look at another method, using multiple samples, for improving LSD-3.

As mentioned before, there are degenerate cases in which LSD-3 cannot identify the correct resolution. Generally, these cases come about when the measured peak area data is in a certain range. Lucas has found, and I have confirmed, that in these degenerate cases, the correct genotype combination is usually ranked within the top 7 or so possibilities. Additionally, the fitting error (the method by which LSD results are ranked) is usually extremely close between the top 7 or so choices. Unfortunately, we need the error to be small for the correct combination and large for all the other incorrect combinations in order for LSD to be able to determine which combination is correct. The idea behind this research stems from the fact that not all cases are degenerate.

Since not all cases are prone to error, we are left with the problem of determining which cases are. And while some cases may not be able to rank the correct combination as first, they may still contain useful information. So what happens if instead of just one sample of a DNA mixture, analysts prepare multiple samples of the same mixture? These different samples would theoretically produce different peak area data because the mass ratios of the samples would be different. Therefore at the very least, having multiple samples increases the chances that we will get a good scenario. And perhaps, even if we have multiple bad scenarios, we can combine the data from the different samples to find the correct genotype combination.
The findings of this study are that involving multiple samples does improve the ability of LSD-3 to correctly resolve a three-person mixture in some cases. By performing LSD on all of the samples simultaneously, we are able to gain more information and reduce the risk of error in identifying the correct genotype than if the samples were taken individually.

**Formulation of LSD MS-3**

As mentioned before, LSD works by solving the Linear Algebra problem $A^*x=b$. Because the peak area data, $b$, is a linear combination of the contributors' genotype combinations and the mass ratio, we can determine which $A$ and $x$ produce the correct $b$. This problem however is inherently inconsistent. That is, in most cases there is no combination of $A$ and $x$ that produces exactly the measured peak area data. This is because the peak area data is subject to noise. So in actuality, the problem takes the form $x_0 = A^+b$, where $A^+$ represents the psuedo-inverse of the matrix $A$. Geometrically this projects the $b$ matrix onto the column space of $A$ and finds the matrix $x_0$ that produces that projection. Using this $x_0$ we can determine, $b_0$, the projection of $b$ onto the column space of $A$. We can then calculate the error of performing this operation as $\text{err} = \|b_0 - b\|$. This process can be simplified to the following expression:

\[
\text{EQ 2} \quad \text{err} = \|(A A^+ - I)b \|
\]

It is this error that LSD uses to rank the fit of a particular genotype combination.

Now, as long as all of the genotype combinations have unique column spaces, we can iterate through every possible $A$ in EQ 2 and find the error associated with every genotype combination. The genotype combination that produces the smallest error is therefore most likely the genotype combination that produced the peak area data.

This has been a very cursory overview of the LSD algorithm. Some details have been left out, but this explanation should be sufficient to understand the motivation and formulation of LSD MS-3.

As mentioned before, LSD works as long as the column spaces of $A$ are unique. In two person mixtures this is the case for most of the possibilities. In the degenerate cases (cases with non-unique column spaces) of LSD-2, it is easy to identify the correct genotype resolution by inspection. In three-person mixtures, however, the number of
possibilities increases exponentially as does the number of degenerate cases. Additionally, there are other cases in which LSD fails to make the correct resolution even when the matrix is not degenerate, and it becomes a non-trivial task to examine all of these cases by inspection. In these cases, LSD ranks several genotype combinations as the best fit, and there is no way to distinguish between them. It is therefore necessary to bring in some more information to improve the abilities of LSD.

One such way to do this is to bring in multiple samples of the same mixture. We can do this analytically by making the x and b vectors X and B matrices. In this way, Eq 2 can be written as follows:

\[
\text{err} = \text{norm}((AA^\dagger - I)B)
\]

where B is the peak area data in the following form:

\[
B = \begin{bmatrix}
    \text{Sample 1} & \text{Sample 2} & \text{Sample 3}
\end{bmatrix}
\]

Essentially the problem is the same. The peak area simply has the form of a matrix, the columns of which come from individual samples of the same DNA mixture. The norm function in Eq 3 finds the largest singular value of the error matrix that is formed by projecting the B matrix onto the columns space of A. This number can again be used to rank the fitting errors of all of the possible genotype combinations.

What we would like to be able to do is determine how much better this new formulation performs compared to taking one sample at a time. To do this, a simulation was designed to test LSD-MS3 against LSD-3, the simple extension of LSD to three-person mixtures. To keep the simulations simple, the scope of the experiment was limited to 4 allele mixtures with 3 contributors. The simulation followed the followings steps:

1. Pick a genotype combination A from the set of all possible combinations for four alleles, A.
2. Begin an iterative loop
   a. Randomly generate a 3x3 mass ratio matrix, MR
   b. Form a clean, synthetic peak area matrix \(B_{\text{clean}}\) by \(B_{\text{clean}} = A \ast MR\).
   c. Add uniform noise to the peak area data and normalize:
noise = rand(3,3) - 0.5
B = (B_clean + noise)
B = B * diag(sum(B))^-1
d. perform LSD-MS3:
   err_i = norm((A_i*A_i^+ - I)B) for each A_i in A.
e. perform LSD-3 on each of the columns of B.
f. sort results from d. and e. and identify the top ranked genotype
   combination as A_i where i is the index that minimizes err_i.
g. Determine whether the top ranked combination is the original A that
   produced B. If not, log the results as a failure.

To paraphrase these steps, we are essentially choosing a genotype combination
and synthetically creating peak area data. We add noise, and try to resolve the correct
genotype combination using the LSD-MS3. We also use each of the individual samples
from the LSD-MS3 method to try to use LSD-3 to resolve the mixture.

From this simulation, we can compare the LSD-MS3 method to LSD-3. The hope
is that we can find cases where LSD-3 fails in any number of the individual samples, and
LSD-MS3 is successful using all of the samples. If this is the case, then we can say that
LSD-MS3 gains us confidence about the resolution of the mixture by using more
information.

In order to visualize the results, we have graphed the data to determine the
number of times LSD-MS3 fails, and the number of times exactly one, two, and three of
the samples fails in LSD-3. We will also look at histogram of where the correct genotype
combination appears in the ranking of LSD in both LSD-MS3 and LSD-3. We do this to
insure that when LSD fails, the correct one is still ranked within the top few.

Before we begin discussing the results of the experiment, we should add a note
about the degenerate cases. As mentioned before, we expect LSD to fail when the
genotype combination does not have a unique column space. In four alleles there are 52
possible genotype combinations. Of those 52, only 16 have unique columns spaces.
Therefore, we will choose to run the experiments with one of the 16 with unique column
spaces to avoid problems with degeneracy. We will then examine how LSD-MS3
performs on the degenerate cases, but we fully expect the results to be poor because of the special case presented by the degenerate cases. Later in this paper I will discuss some possible ways to deal with the degenerate cases. For now, it is obvious that the current method of LSD is not adequate to resolve these cases, so we will focus on improving the non-degenerate situations.

Results and Interpretation

To start, we examine the following non-degenerate genotype combination:

\[ A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \]

We examined how LSD-3 and LSD-MS3 were able to resolve synthetic cases created by this genotype combination with different levels of noise. Specifically we added uniform noise at 5%, 10%, and 15% of the peak area data.

This particular genotype combination has been shown by Lucas’s work to be highly susceptible to noise despite the fact that it is a non-degenerate case. We ran 10,000 iterations, each time choosing a different mass ratio to generate the synthetic data, and each time adding randomly generated noise. First, let us examine the test run with 5% noise.
The top bar chart in figure 1 shows the overall results of the simulation. We see that with 5% noise, over 60% of the time, exactly one out of the three samples we generated causes LSD-3 to be unable to correctly resolve the mixture. About 10% of the time, exactly two out of the three samples generated cannot be resolved. At 0.64%, we see that there are very few cases where all three of the samples generated are not resolvable. And the result that we were looking for is shown the fourth column of the bar chart. The LSD-MS3 method, only fails to correctly resolve the mixture 0.05% of the time. So over 99% of the time, the LSD-MS3 method works correctly for 5% noise.

The bottom two charts display a histogram of where the correct genotype combination appears in the ranking when it is misclassified. As we expect, the correct genotype combination is always ranked within the top 7 choices for the LSD-MS3 method. Using the LSD-3 method we see that some rankings are as bad as 22, though they occur very rarely. We also see from the third chart, that we can expect LSD-3 to fail
roughly 20% of the time, which confirms our understanding that the simple extension of LSD does not work adequately.

So we can conclude from figure 1 that the LSD-MS3 is significantly better than using just one sample in LSD-3 for synthetic data with 5% noise. Next we will examine what happens when we begin adding more and more noise.

Figure 2 - Simulation Results: 10% Noise, Non-Degenerate Case
Figure 3 - Simulation Results: 15% Noise, Non-Degenerate Case

Figure 2 and Figure 3 show the most definitive results from the simulations. We see that as the magnitude of the added noise increases, the number of times 2 and 3 of the samples are incorrectly resolved by LSD-3 increases. Also, from the histograms we can tell that the number of times that LSD-3 correctly identifies the correct genotype combination decreases. The performance of LSD-MS3, however, does not suffer nearly as much. Even with 15% noise, LSD-MS3 fails less than 1% of the time. And the histogram of LSD-MS3 shows that even in these cases, the correct combination is identified within the top 7 choices except for one outlier at 13. The histograms for LSD-3, however still have misclassifications as far out as 22.

The results are similar across all of the non-degenerate genotype combinations. With 15% noise, we can expect LSD-MS3 to fail between 0.3% and 1.5% of the time depending on the genotype combination. And regardless of the genotype combination, LSD-MS3 always performs better than the simple extension of LSD. To better illustrate
how LSD-MS3 outperforms LSD-3, I have provided figures of specific cases that occurred during the simulations.

**Figure 4 – Illustration of LSD-MS3 Working Despite 3 Failures in LSD-3**

Figure 4 illustrates a case when all three of the synthetic samples fail in LSD-3, but LSD-MS3 is able to correctly resolve it. The red circles indicate the correct genotype combination. The x-axis is the rank order that is assigned to a particular genotype combination, and the y-axis shows the normalized error. We see in the bottom 3 graphs, that the correct genotype combination is not ranked anywhere near first for each of the three samples taken individually. Furthermore, in order for LSD to work, the error of the second ranked genotype combination must be at least 4 times greater than that of the first genotype combination. The bottom three graphs show that the second rank error is not that much greater than the first ranked in any of the three samples. The top graph, however, shows that LSD-MS3 was able to correctly identify the correct genotype.
combination. Not only that, but it also has relatively high confidence because the second rank error is about 4 times greater than the first. This result is particularly exciting because it states that LSD-MS3 is able to determine the correct genotype combination with high confidence even when all three of the samples fail individually. This means that LSD-MS3 is using all of the information in the three samples and not just improving its chances by taking more samples.

Now that we have examined how LSD-MS3 works, it is useful to determine why LSD-MS3 doesn’t work in some cases. After all, while it has been shown to be vastly superior to LSD-3 for non-degenerate cases, there is still a small chance that LSD-MS3 will fail. Below, we illustrate some of these cases.

Figure 5 - Illustration of LSD-MS3 Failing When All 3 Samples Fail in LSD-3
Figure 5 illustrates what we might expect to see when LSD-MS3 fails. In this situation, all three of the samples failed, and LSD-MS3 was unable to obtain enough information to correctly resolve the mixture. In fact we see that LSD-MS3 is really unable to discern between the first 6 samples, as the error ratios are very nearly all 1. This result can be thought of as a little better than the worst-case scenario. We see from the second and third graphs, that not only did LSD-3 resolve the mixture incorrectly, but also it assigned an error nearly 5 times as much as the first ranked case for the correct genotype combination. The LSD-MS3 result, however assigned an error very close to 1 for the correct genotype combination. This may not be typical across all such examples, but in this particular case LSD-MS3 at least does a better job of assigning an error to the correct genotype combination.

An interesting continuation of this study might be to look at the cases such as this and determine whether the other genotype combinations on the top ranked are degenerate cases. My suspicion is that they are. LSD is failing because the peak area data lies too close to the column spaces of generate genotype combinations. A more troublesome example of LSD-MS3 failing is shown below.
Figure 6 is troublesome because it shows that LSD-MS3 can still fail even when a majority of the samples can be correctly resolved in LSD-3. Actually that last statement should be taken with a grain of salt. While the second and third graphs show that LSD-3 correctly identifies the correct genotype combination, we must pay attention to the fact that the second ranked combination is too close to the first ranked for the algorithm to have high confidence. Nevertheless, LSD-MS3 does fail despite the fact that two out of three of the samples are correctly resolved in LSD-3. The fourth graph does not have a circled genotype combination. This is because the correct genotype combination is ranked greater than 6, so it is no shown on the chart. This may also contribute to the fact that LSD-MS3 fails in this case.

Again, despite the fact that LSD-MS3 fails, it does correctly identify the genotype combination in the top two ranks. In addition, it has a fairly high degree of confidence that one of the top two is the correct combination. We should note that there are not any
cases where all three of the samples are resolved correctly by LSD-3 and incorrectly by LSD-MS3.

So we have seen that LSD-MS3 works extremely well for genotype combinations that are not degenerate. Even in cases when it fails, it seems to fail better than just taking one sample at a time. So it is quite reasonable to conclude that LSD-MS3 is a vast improvement to the LSD-3 algorithm. However, we have yet to address the issue of degenerate genotype combinations. These situations are known to cause the LSD algorithm to lose all confidence, and we expected from the beginning that LSD-MS3 would not necessarily bring in enough information to help the situation. Below are a few typical results from simulations of LSD-MS3 on degenerate genotype combinations.

**Figure 7 – Simulation Results: Degenerate Case Example 1**

![Simulation Results: Degenerate Case Example 1](image)

[Graph showing simulation results for LSD-3 and LSD-MS3 in degenerate cases.]
Figure 8 – Simulation Results: Degenerate Case Example 2

Figure 9 – Simulation Results: Degenerate Case Example 3
At first glance, these results seem to be pretty hopeless. As expected, LSD-MS3 does not bring in enough information to correctly identify degenerate cases. This is to be expected. For one thing, when a genotype combination does not have a unique column space, there are several other genotype combinations that will always have the same fitting error. Therefore it is impossible to distinguish between these cases based on fitting error alone. And since LSD-MS3 only makes improvements on fitting error resolutions, we shouldn’t expect to see a vast improvement for degenerate cases.

Always the optimist, however, I would like to point out some interesting points in these graphs. The first bar charts in each of the examples does show that LSD-MS3 is at least a little better at resolving degenerate cases, however it is the histograms that are more interesting. In each of the histograms in the three examples, we see that the LSD-MS3 method ranks the correct genotype combinations closer to 1 more often than the LSD-3 method does. Though LSD-MS3 is not necessarily able to correctly identify the correct combination, it does do a better job at keeping it ranked within the top choices than the LSD-3 method.

Conclusions

In summary, we have explored the background of a method of resolving STR DNA mixtures using Least Squares Deconvolution in order to understand how it works for two-person mixtures, and why it fails for three-person mixtures. We have explained the problem of degeneracy and how it affects the LSD-3 method. We then described a method of improving the LSD-3 method by adding multiple samples. Finally, we have analyzed the results of some experiments using this improved method.

Our findings show that LSD-MS3 is a vast improvement over the simple extension of LSD, LSD-3. In cases when the peak area data comes from a genotype combination that is not degenerate, LSD-MS3 all but removes the problems that cause LSD-3 to perform badly. Of course LSD-MS3 is not perfect. It still suffers from some of the same problems as LSD-3, but it seems to handle these problems better. That is, despite the fact that it does not always correctly resolve the mixture, it does not seem to produce misleading resolutions as some cases in LSD-3 do. And while LSD-MS3 does not bring in enough information to solve the degenerate cases, it does at least seem to do a better job at keeping the correct genotype combination near the top of the rankings.
This research is the tip of what could be a very large and interesting iceberg, so to speak. There are some initial improvements and modifications that could be easily made to the simulations. For one thing, the mass ratios that were chosen were completely random. In real situations, there tends to be patterns in how mass ratios are found. For instance, in rape cases, DNA analysts can often reduce the amount of a female’s DNA found in a mixture, and amplify the amount of any male’s DNA. It would be interesting to view the results of this experiment when limiting the mass ratios to specific ranges. Also, as done in Lucas’s research [2], it might be interesting to use specific mass ratios and add varying amounts of noise iteratively to the peak area data.

There is a wealth of work left to be done in the degenerate cases. Since degenerate genotype combinations occur in groups, it would be interesting to see if LSD-MS3 can at least identify the correct group to which a degenerate case belongs. It might also prove fruitful to perform matrix concatenation, also explored by Lucas [2], using multiple samples. This method, which improves the ranking ability of LSD by bringing in requirements for mass ratios, might benefit from the information gained by using multiple samples. All in all, the results found in this study are an exciting first step towards finally being able to resolve three-person mixtures.
References


Appendix A – List of M-files and Their Descriptions

Appendix B. hasUniqueColSpace.m
- usage: bool = hasUniqueColSpace( Candidate, ALL)
- returns: true if Candidate has a unique column space, false otherwise
- description:
  Compares the columns space of Candidate to the columns spaces of the matrices in ALL to determine if the columns space is unique.

Appendix C. uniqueColSpace.m
- usage: [Anew, ind] = uniqueColSpace(ALL)
- returns: all the matrices in ALL that have unique column spaces and their indexes into ALL
- description:
  Uses hasUniqueColSpace to determine the matrices of ALL that have unique columns spaces

Appendix D. thesis.m
- usage: thesis(index, seed, percent_noise, iterations, ['nosave', '<resultsdir>', 'plot'])
- description:
  Performs the simulations using index as an index into the global Alleles matrix which contains the possible genotype combinations for four alleles cases. The seed parameter is a random seed that can be used so that results are reproducible. The percent_noise parameter is the magnitude of noise to add. Iterations is the number of iterations to perform. Additionally some optional arguments are supported such as ‘nosave’ which causes the simulation to not save intermittent results. ‘plot’ plots the individual cases as they occur. A directory to save the results can also be provided via ‘<resultsdir>’. If no directory is specified, a unique directory based on a time stamp is created.

Appendix E. plotOverallResults.m
- usage: plotOverallResults('dir')
- description:
  Plots the overall results saved in directory ‘dir’

Appendix F. plotThesisResults.m
- usage: plotThesisResults('filename')
- description:
  Plots the results of a specific run saved in ‘filename’
Appendix B – hasUniqueColSpace.m

% ret = hasUniqueColSpace( Canidate, All )
function ret = hasUniqueColSpace( Canidate, All )

ret = 1;
[m n z] = size(All);
for i=1:z,
    if (~isequal(Canidate, All(:,:,i))),
        if (rank(Canidate) == rank(cat(2, Canidate, All(:,:,i)))),
            ret = 0;
        end
    end
end

Appendix C – uniqueColSpace.m

function [Anew, ind] = uniqueColSpace( All ),

i=0;
[m n z] = size(All);
for j = 1:z,
    if (hasUniqueColSpace(All(:,:,j), All)),
        i = i+1;
        ind(:,:,i) = j;
        Anew(:,:,i) = All(:,:,j);
    end
end
Appendix D – thesis.m

% function thesis(Aind, seed, noise, iterations)
function thesis(Aind, seed, noise, iterations, varargin)

rand('seed', seed);
warning off
load Genotypes;
A = Alleles_4(:, :, Aind);
doplot = 0;
workdir = strcat('results/', datestr(clock, 30),'/');

for i=1:length(varargin),
    if (strcmp(varargin{i} , 'plot'))
        doplot = 1;
    elseif (strcmp(varargin{i}, 'nosave'))
        workdir = '';
    else
        workdir = strcat(varargin{i},'/');
    end;
end;

if (~strcmp(workdir, ' '))
    mkdir('.', workdir);
    delete(strcat(workdir, '/*.mat'));
end

saveE = 0;
save1e = 0;
save2e = 0;
save3e = 0;
maxEr = 0;
minEr = 10e+6;

histE = zeros(1, 52);
hist1 = zeros(1, 52);
hist2 = zeros(1, 52);
hist3 = zeros(1, 52);

for i = 1 : iterations,
    if (mod(i,round(iterations/20)) == 0)
        fprintf('%i%% done
', round(i/iterations*100));
    end;

    Mr = rand(3, 3);
    B = A * Mr;
    B = B + (rand(4, 3)-.5).*(noise.*B);
    B = B * diag(sum(B))^-1;

    for j = 1:length(Alleles_4),
E(j) = norm(LSD_4(:,:,j)*B);
e1(j) = norm(LSD_4(:,:,j)*B(:,1));
e2(j) = norm(LSD_4(:,:,j)*B(:,2));
e3(j) = norm(LSD_4(:,:,j)*B(:,3));
end

[E, Eind] = sort(E);
histE(find(Eind == Aind)) = histE(find(Eind == Aind)) + 1;

[e1, elind] = sort(e1);
histel(find(elind == Aind)) = histel(find(elind == Aind)) + 1;

[e2, e2ind] = sort(e2);
histe2(find(e2ind == Aind)) = histe2(find(e2ind == Aind)) + 1;

[e3, e3ind] = sort(e3);
histe3(find(e3ind == Aind)) = histe3(find(e3ind == Aind)) + 1;

% if (Eind(l) == Aind)
% saveE = saveE+1;
% fprintf('Big E error\n');
% if (~strcmp(workdir, 'I'))
% save(strcat(workdir, 'saveE', num2str(i)), 'E', 'Eind', ...
% 'e1', 'elind', 'e2', 'e2ind', 'e3', 'e3ind', 'A', 'Mr', 'B', 'noise');
% end;
% end;
if (((elind(1) == Aind) + (e2ind(1) == Aind) + (e3(1) == Aind)) == 1)
% savele = savele+1;
% fprintf('1 error\n');
% if (~strcmp(workdir, 'I'))
% save(strcat(workdir, 'savele', num2str(i)), 'E', 'Eind', ...
% 'el', 'elind', 'e2', 'e2ind', 'e3', 'e3ind', 'A', 'Mr', 'B', 'noise');
% end;
% end;
if (((elind(1) == Aind) + (e2ind(1) == Aind) + (e3(1) == Aind)) == 2)
% save2e = save2e+1;
% fprintf('2 errors\n');
% if (~strcmp(workdir, 'I'))
% save(strcat(workdir, 'save2e', num2str(i)), 'E', 'Eind', ...
% 'el', 'elind', 'e2', 'e2ind', 'e3', 'e3ind', 'A', 'Mr', 'B', 'noise');
% end;
% end;
if (((elind(1) == Aind) + (e2ind(1) == Aind) + (e3(1) == Aind)) == 3)
% save3e = save3e+1;
% fprintf('3 errors\n');
% if (~strcmp(workdir, 'I'))
% save(strcat(workdir, 'save3e', num2str(i)), 'E', 'Eind', ...
% 'el', 'elind', 'e2', 'e2ind', 'e3', 'e3ind', 'A', 'Mr', 'B', 'noise');
% end;
% end;
%save thesislast;

if (length(varargin) > 0)
    if (doplot == 1)
        figure(i);
        plotThesisResults;
    end;
end;

end

if (~strcmp(workdir,''))
    save(strcat(workdir,'Overall'),
         'saveE','save1e','save2e','save3e','Aind','A','noise','iterations','histE','hist1e','hist2e','hist3e','seed');
end;
Appendix E – plotOverallResults.m

%function plotOverallResults(directory, varargin)
function plotOverallResults(directory, varargin)
load Genotypes;
load(strcat(directory, '/Overall.mat'));

figure(1);
cf;

subplot(3,1,1);
bar([savele save2e save3e saveE]./iterations);
ax = axis;
text(.5, ax(4)*1.2, 'A = ');
text(.6, ax(4)*1.1, sprintf(' [%d %d %d]
A'));
if (hasUniqueColSpace(A, Alleles_4))
    text(.8,ax(4)*1.2, strcat('Unique Col Space:
index(',num2str(Aind), ')));
else
    text(.8,ax(4)*1.2, strcat('Non Unique Col Space:
index(',num2str(Aind), ')));
end

text(3, .1, strcat(num2str(save3e./iterations*100), '%'));
text(4,.1 , strcat(num2str(saveE./iterations*100), '1tracent'));
title(sprintf('Overall Statistics for %5.2f%% Noise, %d iterations '...
(Percent Misses) (rand seed = %d) ',noise*100,iterations, seed));
set(gca, 'xticklabel ' , ['misses in 1 sample (LSD-3)
misses in 2 samples (LSD-3)
misses in 3 samples (LSD-3)
misses in E (LSD-M3)'])

subplot(3,1,2);
bar(histE./iterations);
title('Histogram of Correct Resolution Ranking Using All Samples');
axis([0 53 0 1]);
set(gca,'xtick',1:2:52);
xlabel('Ranking');
ylabel('Percent');

subplot(3,1,3);
bar((histel + histe2 + histe3)./(3*iterations));
title('Histogram of Correct Resolution Ranking Using One Sample');
axis([0 53 0 1]);
set(gca,'xtick',1:2:52);
xlabel('Ranking');
ylabel('Percent');
Appendix F – plotThesisResults.m

function plotThesisResults(fname)

clf;
load Genotypes;
load(fname);

EA = Alleles_4(:,:,Eind);
e1A = Alleles_4(:,:,e1ind);
e2A = Alleles_4(:,:,e2ind);
e3A = Alleles_4(:,:,e3ind);

subplot(4,1,1);
hold on;
E = E/(min(E) + eps);
e1 = e1/(min(e1) + eps);
e2 = e2/(min(e2) + eps);
e3 = e3/(min(e3) + eps);
stairs(E, '.-')
axis([0 6 -.5 max([E(6),e1(6),e2(6),e3(6)]*1.3)]);
text(0,max([E(6),e1(6),e2(6),e3(6)]*1.4),fname);
text(0.2,max([E(6),e1(6),e2(6),e3(6)]*1.2), 'A = ');
text(0.35,max([E(6),e1(6),e2(6),e3(6)]*.90), sprintf('[$d $d $d] 
', A'));

text(.9,max([E(6),e1(6),e2(6),e3(6)]*1.2), 'B = ');
text(1.05,max([E(6),e1(6),e2(6),e3(6)]*.90), sprintf('[$5.2f $5.2f 
', B'));
text(2.0,max([E(6),e1(6),e2(6),e3(6)]*1.2), 'Mr = ');
text(2.2,max([E(6),e1(6),e2(6),e3(6)]*.95), sprintf('[$5.2f $5.2f 
', Mr'));

for i=1:length(EA),
    if (isequal(EA(:,:,i), A)),
        res = i;
    end
end
h = plot(res,E(res),'or');
set(h, 'markersize', 12);
title(strcat('Normalized Rank Ordered Errors Across All Three Samples', sprintf('($5.2f%% noise)',noise*100)));
ylabel('Normalized Error (E)');
hold off;

subplot(4,1,2);
hold on;
stairs(e1, '-.');
axis([0 6 -.5 max([E(6),e1(6),e2(6),e3(6)]*1.3)]);
for i=1:length(EA),
    if (isequal(e1A(:,:,i), A)),
        res = i;
    end
end
h = plot(res, e1(res), 'or');
set(h, 'markersize', 12);
title('Normalized Rank Ordered Errors Across Sample 1');
ylabel('Normalized Error (e1)');
hold off;

subplot(4,1,3);
hold on;
stairs(e2, '-.');
axis([0 6 -.5 max([E(6),e1(6),e2(6),e3(6)]*1.3)]);
for i=1:length(EA),
    if (isequal(e2A(:,:,i), A)),
        res = i;
    end
end
h = plot(res, e2(res), 'or');
set(h, 'markersize', 12);
title('Normalized Rank Ordered Errors Across Sample 2');
ylabel('Normalized Error (e2)');
hold off;

subplot(4,1,4);
hold on;
stairs(e3, '-.');
axis([0 6 -.5 max([E(6),e1(6),e2(6),e3(6)]*1.3)]);
for i=1:length(e3A),
    if (isequal(e3A(:,:,i), A)),
        res = i;
    end
end
h = plot(res, e3(res), 'or');
set(h, 'markersize', 12);
title('Normalized Rank Ordered Errors Across Sample 3');
xlabel('Rank Order'); ylabel('Normalized Error (e3)');
hold off;

save last