Distribution Agreement

Signature:

_______________________________ _______________
Hasan Ahmed Date

Similarity and diversity measures for flow cytometry data

By

Hasan Ahmed
Master of Public Health

Department of Biostatistics and Bioinformatics

____
Vicki Hertzberg, PhD
Committee Chair

____
Lisa Elon, MPH, MS
Committee Member

Similarity and diversity measures for flow cytometry data

By

Hasan Ahmed

AB
University of Chicago
2007

Thesis Committee Chair: Vicki Hertzberg, PhD

An abstract of
A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Public Health
in Biostatistics
2011

Abstract

Similarity and diversity measures for flow cytometry data
By Hasan Ahmed

This paper examines similarity measures (also known as dissimilarity or statistical distance measures) for flow cytometry data. Similarity measures quantify the similarity between two objects and could be used for clustering or neighborhood-based predictive modeling. I find that earth mover's distance is the most appropriate tool for creating similarity measures for flow cytometry data. I compare this approach to earlier approaches that relied on Kullback-Leibler divergence, Pearson correlation or Lp distance. This paper also examines diversity measures for flow cytometry data. It identifies two types of diversity measures, "nominal diversity measures" and "interval diversity measure", and it explains the connection between diversity measures and similarity measures.

The similarity and diversity measures in this paper were designed for flow cytometry data, but they can be used for any dataset that consists of multisets of equal-dimensional points. This paper includes R code for implementing many of the methods discussed. The datasets used in this paper have been made publicly available.

Similarity and diversity measures for flow cytometry data

By

Hasan Ahmed

AB
University of Chicago
2007

Thesis Committee Chair: Vicki Hertzberg, PhD

A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Public Health
in Biostatistics
2011

Table of Contents

Similarity and diversity measures for flow cytometry data

I. Introduction

I.A. Flow cytometry

Flow cytometry is a technique for examining the characteristics of individual cells. Typically the cells are first stained with 1 to 10 fluorescent dyes. After staining the cells are acquired by a flow cytometer (i.e. a flow cytometry machine). The flow cytometer creates a stream of fluid which passes the cells one by one through one or more beams of light. As the cells pass through the light, detectors measure the scattering of the light and the light emitted by the fluorescent dyes. Typically a flow cytometer gives two measures of scatter, forward scatter (FSC) and side scatter (SSC), and also measures the amount of fluorescent dye in each cell based on the light emitted. FSC measures the size of the cell, and SSC measures the granularity. Fluorescent dyes that measure various cell characteristics are available.

Flow cytometry is a powerful tool that is widely used in medicine and biomedical research. Consider an HIV+ patient whose blood is stained with 3 dyes: fluorescein isothiocyanate (FITC), phycoerythrin (PE) and peridinin chlorophyll protein (PerCP). FITC is conjugated with an antibody for CD3, a protein complex found on T-cells. PE is conjugated with an antibody for CD4, and PerCP is conjugated with an antibody for CCR5. If 10,000 cells are acquired by a flow cytometer, for each cell the FSC and SSC and the amount of FITC CD3, PE CD4 and PerCP CCR5 will be recorded. HIV typically infects CD4+ CCR5+ cells. These cells can be identified as those cells high in PE CD4 and PerCP CCR5. FSC and SSC can be used to identify important cell subsets such as lymphocytes. A low ratio of CD4+ cells to CD3+ cells is sign of progression to AIDS. Hence flow cytometry can provide a lot of important information.

I.B. Similarity and diversity measures

Similarity measures quantify the similarity of two objects. Euclidean distance is an example of a common similarity measure. The smaller the Euclidean distance between two points is, the more similar the two points are in terms of location. In this paper all similarity measures will be defined so that a smaller similarity score corresponds to greater similarity and 0 indicates perfect similarity. While this usage may be confusing, it is helpful when discussing the triangle inequality, and it is necessary for consistency with common similarity measures like Euclidean distance and Kullback–Leibler divergence. Similarity measures are also called similarity metrics or indices, dissimilarity measures or metrics or indices, divergence measures, distance measures or metrics and sometimes just metrics. However the term metric should be reserved for the subset of similarity measures that obey the following four properties: $S(X,Y)>=0$; $S(X,Y)=0$ if and only if $X=Y$; $S(X,Y)=S(Y,X)$; and $S(X,Z)<S(X,Y)+S(Y,Z)$ where S is the similarity measure and X, Y and Z are objects being compared.

Table 1: Various similarity measures

Similarity measure	*Typically used for comparing…*
Fréchet distance	Curves
Beta diversity	Ecosystems
Google similarity distance	Meaning of words and phrases
Chebyshev distance	Points
Great-circle distance	Points on a sphere
Bhattacharyya distance	Probability distributions
Jensen-Shannon divergence	Probability distributions
Kendall tau distance	Rankings
Levenshtein distance	Strings
Hamming distance	Strings of equal length

This paper will discuss similarity measures for evaluating the similarity between two multisets of points. This is quite different from evaluating the similarity between two points or between two strings.

A multiset is a collection of elements where multiplicity matters but order does not. For example {a,b,c} is equivalent to {c,b,a}, but {a,a,a,b,c} is not equivalent to {a,b,c}. Multisets can be contrasted with sequences, where both order and multiplicity matter, and sets where neither order or multiplicity matter. Union, intersection and subset are defined slightly differently for multisets than for sets. For example {1,2,3}U{1,2,3}={1,1,2,2,3,3}, {1,1,2,3}∩{1,1,2,2}={1,1,2} and {1,1} is a subset of {1,1,1} but not of {1,2,3,4}.

Nonetheless it is not convenient to completely ignore the order of a multiset. In this paper order is used to reference the elements of a multiset. For example the last two elements of {1,2,3,4} are 3 and 4. Two equivalent multisets with different orderings (e.g. {1,2,3} and {3,2,1}) are considered equal but not identical.

Diversity measures quantify the diversity of an object, usually a multiset or probability distribution. A multiset that contains one value repeated many times (e.g. {Apple,Apple,Apple,Apple} or {1,1,1,1}) is not very diverse. Likewise a probability distribution concentrated on a narrow range of values (e.g. X~uniform(0.999,1)) is not very diverse. On the other hand the following are more diverse: {Apple,Banana,Orange,Pear}, {1,99,-8,72}, Y~uniform(0,101). Shannon entropy and Simpson's index are examples of diversity measures.

Similarity measures for flow cytometry data have many potential uses. They could be used for clustering flow cytometry samples, k-nearest neighbor models or other neighborhood-based predictive models. The uses of flow cytometry diversity measures are less clear. But they may be useful in certain cases where diversity is found to correlate with important biological outcomes.

The similarity and diversity measures in this paper are designed for flow cytometry data, but most of them can be used to compare any two multisets of points as long as all the points have the same number of dimensions.

I.C. Previous work

There is little work on diversity measures for flow cytometry data, but several groups have investigated similarity measures for flow cytometry data.

Diaz-Romero et al [1] evaluated similarity measures for clustering tumor samples based on flow cytometry data. They stained samples with 11 fluorescent dyes, but the samples were only stained with one antibody at a time. Hence they ignored the correlations between the various markers they stained for. They represented each sample as a point (x1,…,x11) where xj is the mean fluorescent intensity of the jth fluorescent stain after normalization. They used Euclidean (L2) distance, Manhattan (L1) distance and Pearson correlation to evaluate the similarity of the points. They found that Pearson correlation outperformed L1 and L2 distance. This method has the advantage of being relatively simple, but it ignores the multivariate nature of the data.

Kaufman et al [2] created a similarity measure based on data binning. They divided the flow cytometry space into N equal-sized hyperrectangles. They represented each sample as a probability mass function P where P(x) is the proportion of cells from that sample that fall in bin x. They defined similarity as the L1 distance between two probability mass functions, $\Sigma|A(x)-B(x)|$ where A and B are probability mass functions. They also calculated the optimal number of bins based on the number of cell in the samples. But this number is likely to change based on the dimensionality of the dataset, and Kaufman et al only examined 6-dimensional data (FSC, SSC and 4 fluorescent markers). Kaufman's method is relatively simple and takes into account the multivariate nature of the data, but it is very susceptible to the curse of dimensionality. As the number of dimensions increases, either the number of bins will increase greatly and the number of cells in each bin will become very sparse or the length of the bins will need to increase greatly.

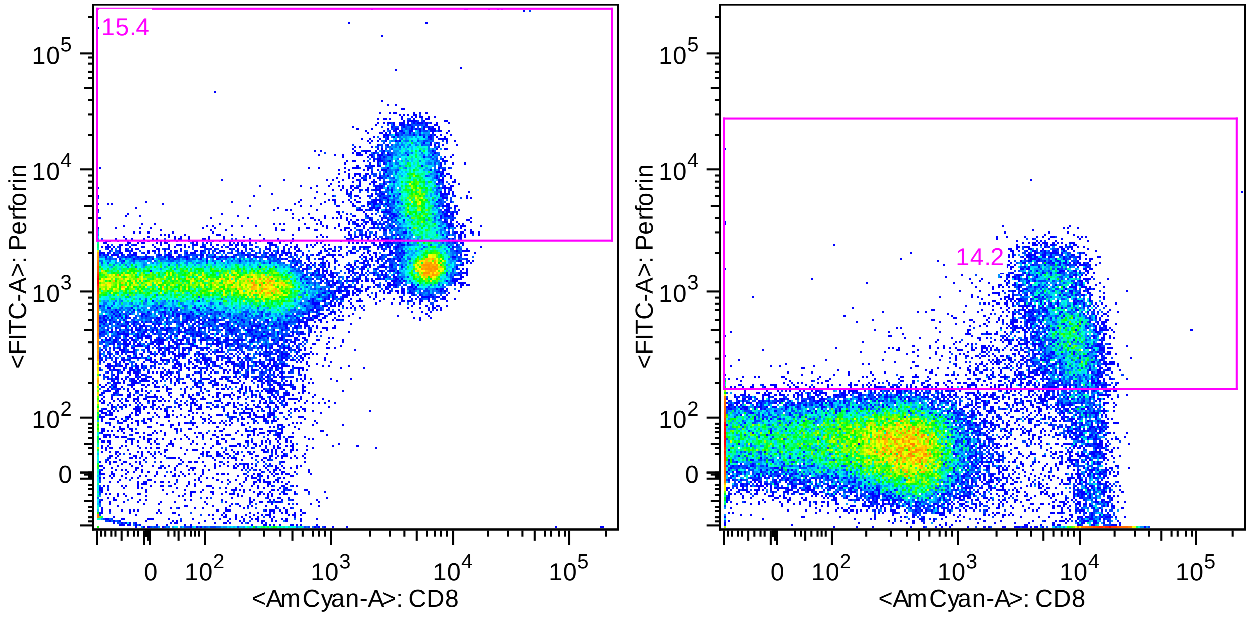
Alfred Hero and colleagues have published several papers [3,4,5] involving a similarity measure. They use Gaussian kernels to create a probability density function for each sample. They then use a symmetric version of Kullback-Leibler divergence to calculate the similarity between these probability density functions. The symmetric version of Kullback-Leibler divergence SKL is defined as SKL(X,Y) = KL(X,Y)+KL(Y,X) where KL is the standard version of Kullback-Leibler divergence and X and Y are probability distribution functions. This approach takes into account the multivariate nature of flow cytometry data and partly avoids the curse of dimensionality, but it is computationally difficult.

Roederer et al [6] do not propose a similarity measure, but they do introduce a data binning method that may be useful for flow cytometry similarity measures. Their algorithm picks the highest variance dimension and divides that dimension along its median to create two bins. The algorithm then divides the highest variance dimension in each bin along its median and repeats this process until a stopping criterion (e.g. total number of bins or the number of cells in each bin) is reached. At the end there should be 2^N bins, where N is a positive integer, each with an approximately equal number of cells.

I.D. Assay variation and Boolean gate data

Flow cytometry data can be susceptible to assay variation. If a sample is stained for longer, at higher temperature or in lower volume, the stain will be brighter. Likewise two different vials of FITC CD3 antibody may stain differently. Consider the two samples in figure 1.

Figure 1: Assay variation between two samples



Both samples have been stained with FITC perforin and AmCyan CD8. Perforin is a protein used by CD8+ cells to form pores in cell membranes. With rare exception only CD8+ cells should express perforin. The cells inside the rectangles can be considered perforin+. Most of these cells express perforin, whereas most of the cells outside the rectangles do not. In both samples there is a clear distinction between perforin+ and perforin- cells that is obvious to any trained human. But because the perforin- cells in the first sample are as bright as many of the perforin+ cells in the second sample, a similarity measure may judge these samples to be very different. This difference in brightness is almost entirely attributable to assay variation. Based on the percent of perforin+ cells, the two samples are fairly similar.

One solution to this problem is to use manual gating to distinguish between + and – cells. Cells that are positive for an antibody can be given a 1 in the appropriate dimension. A cell that is negative for an antibody can be given a 0 in the appropriate dimension. For example a cell that is FITC CCR5-, PE CD3+ and PerCP CD4+ would be represented as (0,1,1), and a cell that is FITC CCR5-, PE CD3+ and PerCP CD4- would be represented as (0,1,0). I call this type of data “Boolean gate” data because FlowJo, a program for analyzing flow cytometry data, has a Boolean gate tool which is useful for creating this data. I call the data outputted by the flow cytometer “raw” data, although this is not technically correct since this data has been adjusted for spillover [7].

Table 2 shows an example of raw data. Each row represents a cell. Table 3 shows the same data after it has been transformed into Boolean gate data.

Table 2: Example of raw data

FITC CCR7	*PE CCR5*	*PerCP CD4*	*PE-Cy7 CD28*	*Pacific Blue CD95*	*AmCyan CD8*	*Alexa 700 CD3*
898	1203	897	1069	1742	2823	2293
1166	625	718	1155	1223	2783	769
1742	1040	568	1003	1294	2476	774
1392	263	887	1325	1567	3052	2387
1353	552	883	1203	1376	1421	912
1002	110	636	720	1748	2887	2635
1024	779	473	544	569	2253	935
1314	1465	969	1370	1801	1784	1361

Table 3: Example of Boolean gate data

FITC CCR7	*PE CCR5*	*PerCP CD4*	*PE-Cy7 CD28*	*Pacific Blue CD95*	*AmCyan CD8*	*Alexa 700 CD3*
0	0	0	0	0	1	1
0	0	0	0	0	1	0
1	0	0	0	0	1	0
0	0	0	0	0	1	1
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Because points in Boolean gate data can have only 2^d locations where d is the number of dimensions, it is generally more efficient to express Boolean gate data in the below format (table 4) where weight is the proportion of cells from that sample at that location.

Table 4: Example of Boolean gate data in compact form

PerCP CD4	*AmCyan CD8*	*Alexa 700 CD3*	*Weight*
0	0	0	0.3722533
0	0	1	0.0711290
0	1	0	0.0009839
0	1	1	0.0000401
1	0	0	0.0209502
1	0	1	0.2486399
1	1	0	0.2629366
1	1	1	0.0230670

II. Similarity measures

II.A. Overview of similarity measures

Most of the similarity measures discussed in this paper have two steps. First the multisets of points are approximated by moments or probability distribution functions. Then an established similarity measure - namely Lp distance, Pearson correlation, Kullback-Leibler divergence or earth mover's distance - is used to compare the moments or probability distribution functions. Earth mover's distance is also used to directly measure similarity without calculating moments or probability distribution functions.

Lp distance is a family of similarity measures that includes Euclidean (or L2) distance and Manhattan (or L1) distance. The Lp distance between two objects A and B is defined as $(\Sigma|(A(i)-B(i))|^{p})^{(1/p)}$ or $(\int|A(x)-B(x)|^{p}\,dx)^{(1/p)}$.

The Pearson correlation between two objects A and B is defined as cov(A,B)/sqrt(var(A)*var(B)). Pearson correlation ranges from -1 (least similarity) to 1 (greatest similarity). In order to make Pearson correlation consistent with other similarity measures, I use a modified form of Pearson correlation 1-cov(A,B)/sqrt(var(A)*var(B)), which ranges from 0 (greatest similarity) to 2 (least similarity). This form of Pearson correlation is sometimes called Pearson correlation distance.

The Kullback-Leibler divergence between A and B is defined as $\Sigma(A(i)*\ln(A(i)/B(i)))$ or $\int A(x)*\ln(A(x)/B(x))\,dx$ where A and B are probability density functions or probability mass functions. Because Kullback-Leibler divergence is not symmetric, I use the symmetric form of Kullback-Leibler divergence that was used by

Alfred Hero and colleagues [3,4,5]. Symmetric Kullback-Leibler divergence is equal to KL(A,B)+KL(B,A) where KL is the normal form of Kullback-Leibler divergence.

The earth mover's distance [8] between A and B is the minimum cost of making A identical to B. If A and B are probability density functions or probability mass functions, cost is the amount of mass moved times the distance the mass is moved. In this paper both L1 distance and L2 distance are used to calculate cost. If A and B are multisets of points, cost is the sum of the distances that points are moved. For probability mass functions finding the minimum cost is equivalent to a transportation problem in linear programming. For multisets of points finding the minimum cost is equivalent to an assignment problem in linear programming. Earth mover's distance is also known as 1st Mallows distance or 1st Wasserstein distance.

II.B. Moment methods

II.B.1. First moment method

The simplest way to compare two multisets of points is to calculate the mean or first moment of each multiset. These means are then compared using L1 distance, L2 distance or Pearson correlation distance. This method is equivalent to the method that Diaz-Romero et al [1] used.

II.B.2. Second moment method

The second moment of each multiset is calculated. Then the second moments are compared using L1 distance, L2 distance or Pearson correlation distance.

The second moment of A, a multiset of n-dimensional points, is equal to E(A)*t(E(A))+cov(A) where E(A) is the mean of A represented as an n by 1 matrix, t(E(A)) is the transpose of E(A) and cov(A) is the maximum likelihood covariance matrix of A. The maximum likelihood covariance matrix is used instead of the unbiased covariance matrix so that the “size property” (see section III.A) applies. Unlike the first moment the second moment incorporates correlations between variables in the dataset. In practice this method gives results very similar to the first moment method, and the cov(A) term is dwarfed by the E(A)*t(E(A)) term.

II.C. Probability density function method

Gaussian kernel density estimation is used to create a probability density function for each multiset. Then L1 distance, L2 distance, symmetric Kullback-Leibler divergence or earthmover's distance are used to compare these probability density functions. This method is very computationally difficult. Kernel density estimation involves creating a probability density function for each point in the multiset and then adding these probability density functions. Comparing probability density functions using Lp distance or Kullback-Leibler divergence involves integrating over multidimensional space. In theory two probability density functions can also be compared using earth mover's distance, but I am not aware of any method for solving this problem except by approximating the probability density functions with probability mass functions. Gaussian kernel density estimation with symmetric Kullback-Leibler divergence is the method used by Alfred Hero and colleagues [3,4,5].

II.D. Probability mass function methods

All multisets in a dataset are combined to create a reference multiset. A binning method is used to create data bins based on the reference multiset. Then a probability mass function P is created for each multiset based on the data bins so that P(i) is the proportion of points from that multiset that are in the ith data bin. These probability mass functions are compared using L1 distance, L2 distance, Pearson correlation distance, symmetric Kullback-Leibler divergence or earth mover's distance. When calculating earth mover's distance, a bin's mean is treated as its location. Alternatively the bin's median or centroid could be used.

In this paper the reference multiset is created by combining all multisets in a dataset without any reweighting. But other approaches may be desirable in certain circumstances. If new multisets are constantly being added to a dataset, having a fixed reference multiset may be desirable. If certain multisets are much larger than other multisets, it may make sense to reweight the points in the reference multiset so that all multisets are equally represented. If the reference multiset does not contain all points in a dataset, it is possible some of these points will fall outside any bin. In this case a point should be assigned to its nearest bin.

Various data binning methods are described below.

II.D.1. Equal-sized bins method

If n is the dimensionality of the points in the multiset, s^n bins are created by dividing each dimension into s equal-length sections (where s is an integer greater than 1). The first of these sections begins at the minimum value observed for that dimension, and the last of these sections ends at the maximum value observed for that dimension. Boolean gate data is naturally divided into 2^n bins which makes this method especially attractive for Boolean gate data. On the other hand this method is very susceptible to the curse of dimensionality. If s is kept constant, the number of bins increases exponentially with the number of dimensions. This means that the computational difficulty of this method will increase greatly and the number of events in each bin will become very sparse, which could make the method less accurate. If s is reduced, the length of each bin will become larger, which could also make the method less accurate. Furthermore s cannot be reduced to less than 2.

This method is very similar to the method used by Kaufman et al [2].

II.D.2. K-means clustering method

K-means clustering [9] is used to create data bins. Unlike most other methods k-means clustering is not entirely deterministic. The k-means clustering algorithm begins by guessing k means, and the final binning varies depending on these guesses.

II.D.3. Sum of squares method

The dimension with the greatest sum of squares is divided along its mean to create two bins. Then the bin with the dimension with the greatest sum of squares is identified, and that dimension in that bin is divided along its mean. This process is repeated until the desired number of bins is created.

This method is inspired by the binning method described by Roederer et al [6] but differs from it in several ways. Roederer's method is designed to create 2^n bins (where n is a positive integer) with an approximately equal number of points. This approach is problematic for Boolean gate data or any other dataset with a large number of identical points. For example if 25% of the points in the reference multiset are (0,0,0,0), then at most four equal-weight bins can be created. Furthermore since Roederer’s method divides dimensions along their median, if many points are equal to the median, bins may have very unequal number of points.

II.D.4. Trees: a supervised method

Each point in the reference multiset is linked to an outcome based on the multiset it originated from. The outcome may be multivariate or univariate. A regression tree [9] that predicts the outcome based on the points is fit. The terminal nodes of this tree are used as data bins.

All of the previous data binning methods are unsupervised. They consider only the distribution of the points in determining how to create bins. In many cases some dimensions may be more important than others and this may not be apparent from the distribution of points. A potential solution to this problem is to link each multiset to some outcome. This outcome then acts as a “supervisor”. For example flow cytometry samples from HIV+ patients could be linked to the HIV viral load of the patients. A regression tree would find regions where there is a preponderance of cells from patients with low viral load or a preponderance of cells from patients with high viral loads. By using these regions as data bins, this binning strategy may focus on more important dimensions and regions. On the other hand this strategy may ignore regions that are not associated with viral load but nonetheless important in some other way. A solution to this problem is to use a multivariate outcome (e.g. HIV viral load and self-reported health).

A disadvantage to this method is that in some circumstances a tree will only create one data bin. In this paper the tree algorithm will only split a dimension if that split increases R^2 (the coefficient of determination) by at least 0.01. If no split increases R^2 by 0.01 or more, only one bin will be created. This problem could be avoided by allowing splits that increase R^2 by less than 0.01. But this defeats the purpose of this method which is to create bins that are meaningfully associated with the supervisor. If the tree algorithm creates only one bin, this should be interpreted as evidence that the supervisor is not appropriate for the dataset.

II.D.5. Bumped trees

Compared to other predictive models like linear regression, regression trees tend to do a poor job of predicting the outcome. (On the other hand there is no obvious way to use linear regression to create data bins.) Bumping [9] is one way to improve the performance of regression trees. Bootstrapping and random cost [10] can be used to generate many trees. These trees are tested using the original data, and the best performing tree is selected. Because trees are created using a greedy algorithm, bumping can greatly improve their predictive performance.

For some types of data it may be possible to measure the accuracy of the similarity measure. For example if a similarity measure is used to create a neighborhood-based predictive model, the accuracy of this predictive model can be considered the accuracy of the similarity measure. In this case trees could also be bumped to maximize this accuracy. In other words after bootstrapping and random cost are used to generate trees, the tree that gives the most accurate similarity measure could be chosen. From my experience this approach gives similarity measures that are very accurate for the reference multiset, but these similarity measures are likely to be overfit.

II.E. Cumulative distribution function method

Empirical cumulative distribution functions are calculated for each multiset. These cumulative distribution functions are compared using L1 distance or L2 distance, which involves integrating over multidimensional space. For this reason this method is very computationally difficult.

II.F. Pure earth mover's distance

The first N points from each multiset are chosen. This step is necessary to ensure equal cardinality. If A' is the first N points from A and B' is the first N points from B, then the similarity between A and B is calculated by finding the earth mover’s distance between A’ and B’. This problem is equivalent to an assignment problem in linear programming and can be solved using the Hungarian algorithm. If a multiset contains less than N points, then the points in the multiset should be repeated until N points are reached.

This method is very computationally intensive if N is large. For small N the first N points may not be representative of the multiset, which could make this method inaccurate.

III. Comparison of similarity measures

III.A. Desirable properties for similarity measures

If S(X,Y) is the similarity score between two multisets X and Y, then the similarity measure S should have the following properties.

1. S(X,Y)=S(X,Y). The similarity measure should not be random. More precisely the similarity measure should not vary based on an arbitrary parameter.

2. S(X,Y)>=0. The similarity score should be nonnegative.

3. The “perfect similarity property”: S(X,Y)=0 if X=Y. A multiset should be perfectly similar to an equivalent multiset.

4. S(X,Y)=S(Y,X). The similarity measure should be symmetric.

5. The “size property”: S(X,Y)=S(X,YUY...UY}) where YUY is the union of Y with itself. The number of cells in a flow cytometry sample depends more on assay variation (the amount of blood or tissue collected, the speed of the flow cytometer, etc.) than on biological variation. Therefore the similarity measure should only depend on the distribution of the cells and not on the number of cells.

6. S(X,Z)<=S(X,Y)+S(Y,Z). Ideally the similarity measure should obey the triangle inequality, although this property is not essential.

7. The similarity measure should be resistant to assay variation.

8. Computational difficulty. The similarity measure should be easy to implement and not computationally intensive.

9. Resistance to the curse of dimensionality. The performance or computational efficiency of the similarity measure should not decrease too much when the number of dimensions increases.

Table 5 shows how the similarity measures described in section II compare based on these properties. Note that each of the first nine methods must be paired with one of the next four methods to produce a multiset similarity measure. Only pure earth mover's distance is a multiset similarity measure by itself. Property 7, resistance to assay variation, is not included in table. All methods are susceptible to assay variation if raw data is used and resistant if Boolean gate data is used. Section III.D discusses this issue further. The evaluations for property 9, resistance to the curse to dimensionality, should be taken with a grain of salt since they are not based on simulations or rigorous calculation.

Table 5: Comparison of similarity measures based on properties

Method	*1. Not random*	*2. Nonnegative*	*3. Symmetric*	*4. Perfect similarity property*	*5. Size property*	*6. Triangle inequality*	*8. Computational difficulty*	*9. Resistance to the curse of dimensionality*
First moment	Yes				Yes		Low	Moderate
Second moment	Yes				Yes		Low/moderate	Moderate
Probability density function	Yes				Yes		High	?
Equal-sized bins	Yes				Yes		Moderate	Low
K-means clustering	No				Yes		Moderate	Moderate
Sum of squares	Yes				Yes		Moderate	Moderate
Tree method	Yes				Yes		Moderate	High
Bumped trees	No				Yes		Moderate	High
Cumulative distribution function	Yes				Yes		High	?
Lp distance (p>=1)		Yes	Yes	Yes		Yes	Low	
Pearson correlation distance		Yes	Yes	Yes		No	Low	
Symmetric Kullback-Leibler divergence		Yes	Yes	Yes		No	Low	
Earth mover's distance		Yes	Yes	Yes		Yes	Moderate	
Pure earth mover's distance	Yes	Yes	Yes	No	No	Yes	Moderate	Moderate

Similarity measures that obey properties 2, 3, 4 and 6 are pseudometrics. None of the multiset similarity measures described in section II are true metrics. In fact the size property contradicts a property (S(X,Y)=0 if and only if X=Y) that is required for true metrics. Pure earth mover’s distance is the only measure that does not obey property 4 (the perfect similarity property). The pure earth mover’s distance between two equivalent but not identical multisets is not necessarily zero (although it tends to be close to zero).

Note that Rubner et al [8] provide a simple proof showing that earth mover’s distance obeys the triangle inequality. The earth mover’s distance between X and Y is the minimum cost of transforming X into Y. If S(X,Z)>S(X,Y)+S(Y,Z), then transforming X into Y and then into Z costs less than the minimum cost of transforming X into Z, which is clearly a contradiction.

III.B. Other considerations

A major shortcoming of most probability mass function methods is that the results vary depending on which reference multiset is used to create data bins. Fortunately the equal-sized bins method with Boolean gate data can avoid this shortcoming. Because Boolean gate data is naturally divided into equal-sized bins, 2^d bins (where d is the number of dimensions in the dataset) can be created without even using a reference multiset.

Kullback-Leibler divergence and Pearson correlation have some undesirable qualities that are not apparent from table 5. Because the Kullback-Leibler divergence between two objects A and B is defined as $\Sigma(A(i)*ln(A(i)/B(i)))$ or $\int A(x)*ln(A(x)/B(x))\ dx$, Kullback-Leibler divergence will be infinite if there is any z where B(z) is equal to zero and A(z) is not. For this reason, even distributions that are intuitively very similar (e.g. uniform(0,100) and Uniform(0.001,100.001)) are infinitely dissimilar based on Kullback-Leibler divergence. This problem is inherited by the symmetric form of Kullback-Leibler divergence used in this paper. Similarly Pearson correlation often contradicts intuitive notions of similarity. For example the arrays [0, 1/6, 1/3, 0.5] and [5/23, 11/46, 6/23, 13/46] are intuitively quite different but are perfectly similar according to Pearson correlation.

The first moment method has a desirable quality that other methods lack. Because relationships between variables are ignored, this method will work even if variables have not been observed together. For example imagine that blood from HIV+ patients is stained three different times: once with FITC CCR5, once with PE CD3 and once with PerCP CD4. The first moment method will treat this data the same way it would treat samples that have been simultaneously stained FITC CCR5, PE CD3 and PerCP CD4: the mean fluorescent intensity of FITC CCR5, PE CD3 and PerCP CD4 will be calculated for each patient and these means will be combined into 3-dimensional means, which can then be compared using L1 distance, L2 distance or Pearson correlation distance.

III.C. Testing the accuracy and consistency of similarity measures using dataset 1

Dataset 1 consists of flow cytometry samples from eight SIV (simian immunodeficiency virus) infected monkeys. The samples were stained with FITC CCR7, PE CCR5, PerCP CD4, PE-Cy7 CD28, Alexa 700 CD3, Pacific Blue CD95 and AmCyan CD8 [11]. FSC and SSC were used to create a gate for lymphocytes. Only cells from the lymphocyte gate are included in the dataset. FSC and SSC are not included in the dataset. The dataset contains 30,000 cells from each sample.

Both Boolean gate and raw versions of this dataset were used. The raw data was not transformed or cleaned in any way, although in many cases transformation or data cleaning may be helpful. For example Diaz-Romero et al [1] found that log transformation followed by z-score transformation worked better than untransformed data.

Dataset 1 was used to test the accuracy of the methods discussed in section II. First the methods were used to calculate similarities. Then for each similarity measure I calculated the correlation between the log SIV viral load of the monkeys and the log SIV viral load of their nearest neighbor as determined by the similarity measure. Dataset 1 contains information that is highly relevant to SIV pathogenesis (e.g. the number of CD4+ cells, CD28 expression and CCR5 expression). If a similarity measure “accurately” measures the similarity between samples, we would expect monkeys with similar samples to have similar viral loads. The table 6 describes the results of this test.

Not all of the methods discussed in section II were tested. The probability density function method and the cumulative distribution function method were not tested because they are computationally very difficult. The k-means clustering method and the bumped trees method were not tested because they are partly random and therefore do not give consistent results. Pure earth mover's distance was only tested using raw data. For Boolean gate data the equal-sized bins method with earth mover's distance is equivalent to pure earth mover's distance but is more computationally efficient and uses all points in a dataset. All other methods were tested. The equal-sized bins method was implemented with 2^7 bins. The sum of squares method was implemented with 10 bins and with 50 bins. Pure earth mover's distance was implemented using the first 256 cells from each sample. For the tree method log SIV viral load was used as the supervisor.

Table 6: Comparison of similarity measures based on the correlation between log viral load and nearest-neighbor log viral load

Method (first step)	*Method (second step)*	*Data type*	*Correlation*
First moment	L1 distance	Raw	0.6813
First moment	L2 distance	Raw	0.6485
First moment	Pearson correlation distance	Raw	0.5884
First moment	L1 distance	Boolean	0.4325
First moment	L2 distance	Boolean	0.4027
First moment	Pearson correlation distance	Boolean	-0.7774
Second moment	L1 distance	Raw	0.6485
Second moment	L2 distance	Raw	0.6485
Second moment	Pearson correlation distance	Raw	0.5538
Second moment	L1 distance	Boolean	0.6752
Second moment	L2 distance	Boolean	0.5395
Second moment	Pearson correlation distance	Boolean	-0.7774
Equal-sized bins	L1 distance	Raw	0.3290
Equal-sized bins	L2 distance	Raw	0.1356
Equal-sized bins	Pearson correlation distance	Raw	-0.0271
Equal-sized bins	Symmetric Kullback-Leibler divergence	Raw	NA
Equal-sized bins	L1 earth mover's distance	Raw	0.5683
Equal-sized bins	L2 earth mover's distance	Raw	0.5683
Equal-sized bins	L1 distance	Boolean	0.4671
Equal-sized bins	L2 distance	Boolean	0.6817
Equal-sized bins	Pearson correlation distance	Boolean	0.2054
Equal-sized bins	Symmetric Kullback-Leibler divergence	Boolean	NA
Equal-sized bins	L1 earth mover's distance	Boolean	0.5683
Equal-sized bins	L2 earth mover's distance	Boolean	0.5683
Sum of squares (10 bins)	L1 distance	Raw	0.1893
Sum of squares (10 bins)	L2 distance	Raw	-0.2049
Sum of squares (10 bins)	Pearson correlation distance	Raw	-0.7513

Sum of squares (10 bins)	Symmetric Kullback-Leibler divergence	Raw	-0.1434
Sum of squares (10 bins)	L1 earth mover's distance	Raw	0.5395
Sum of squares (10 bins)	L2 earth mover's distance	Raw	0.5395
Sum of squares (10 bins)	L1 distance	Boolean	0.2839
Sum of squares (10 bins)	L2 distance	Boolean	0.6817
Sum of squares (10 bins)	Pearson correlation distance	Boolean	-0.5920
Sum of squares (10 bins)	Symmetric Kullback-Leibler divergence	Boolean	0.1462
Sum of squares (10 bins)	L1 earth mover's distance	Boolean	0.5574
Sum of squares (10 bins)	L2 earth mover's distance	Boolean	0.4671
Sum of squares (50 bins)	L1 distance	Raw	0.6380
Sum of squares (50 bins)	L2 distance	Raw	0.0917
Sum of squares (50 bins)	Pearson correlation distance	Raw	0.1990
Sum of squares (50 bins)	Symmetric Kullback-Leibler divergence	Raw	0.3751
Sum of squares (50 bins)	L1 earth mover's distance	Raw	0.5395
Sum of squares (50 bins)	L2 earth mover's distance	Raw	0.5490
Sum of squares (50 bins)	L1 distance	Boolean	0.4671
Sum of squares (50 bins)	L2 distance	Boolean	0.6817
Sum of squares (50 bins)	Pearson correlation distance	Boolean	0.2054
Sum of squares (50 bins)	Symmetric Kullback-Leibler divergence	Boolean	0.1151
Sum of squares (50 bins)	L1 earth mover's distance	Boolean	0.5683
Sum of squares (50 bins)	L2 earth mover's distance	Boolean	0.5683
Tree	L1 distance	Raw	0.4448
Tree	L2 distance	Raw	0.4448
Tree	Pearson correlation distance	Raw	0.2666
Tree	Symmetric Kullback-Leibler divergence	Raw	0.6782
Tree	L1 earth mover's distance	Raw	0.4448
Tree	L2 earth mover's distance	Raw	0.4448
Tree	L1 distance	Boolean	0.7413
Tree	L2 distance	Boolean	0.7413
Tree	Pearson correlation distance	Boolean	0.4906
Tree	Symmetric Kullback-Leibler divergence	Boolean	0.7392
Tree	L1 earth mover's distance	Boolean	0.7413
Tree	L2 earth mover's distance	Boolean	0.7413
Pure L1 earth mover's distance		Raw	0.6550
Pure L2 earth mover's distance		Raw	0.7553
First moment	*	*	0.3293
Second moment	*	*	0.3814
Equal-sized bins	*	*	0.4065
Sum of squares (10 bins)	*	*	0.1428
Sum of squares (50 bins)	*	*	0.4165
Tree	*	*	0.5766
*	L1 distance	*	0.4999
*	L2 distance	*	0.4578
*	Pearson correlation distance	*	-0.0347
*	Symmetric Kullback-Leibler divergence	*	0.3184
*	L1 earth mover's distance†	*	0.5659
*	L2 earth mover's distance†	*	0.5558
*	*	Raw	0.3883
*	*	Boolean	0.3907
*	*	*	0.3895
†does not include pure earth mover's distance			

Overall the tree method outperformed moment methods and other probability mass function methods. The tree method with Boolean gate data performed especially well. Pure L2 earth mover's distance also performed well. The sum of squares method with 10 bins performed poorly compared to other unsupervised probability mass function methods, which may suggest that 10 bins are too few for an unsupervised binning method. In contrast the tree method created only 3 bins and performed well.

Overall L1 earth mover's distance and L2 earthmover's distance outperformed competing methods. Furthermore they consistently performed well. The nearest-neighbor correlations for L1 earth mover's distance and L2 earth mover's distance ranged from 0.4448 to 0.7553. Pearson correlation distance performed especially poorly. Symmetric Kullback-Leibler divergence also performed relatively poorly. Furthermore when combined with the equal-sized bins method, symmetric Kullback-Leibler divergence could not identify a nearest neighbor for any sample because all samples were judged to be infinitely dissimilar from each other. (When combined with the sum of squares method with 50 bins and Boolean gate data, symmetric Kullback-Leibler divergence judged certain samples to be infinitely dissimilar but was able to identify a nearest-neighbor for all samples.) This possibility was discussed in section III.B.

Dataset 1 was also used to test the consistency of these methods. In order to do so, each similarity measure was represented by a vector of its similarity scores, [S(X1,X2),S(X1,X3),…,S(X2,X1),…,S(X8,X7)] where Xi is the ith sample and S is the similarity measure, and the correlations between these vectors were calculated. Note that these vectors do not contain the similarity scores between a sample and itself (e.g. S(X4,X4)) since these similarity scores are always zero. The entire correlation matrix can be found in the supplementary tables file. Table 7 shows the mean correlation between a similarity measure and all other similarity measure implementations. All correlation means exclude the necessarily perfect correlation between an implementation and itself.

Table 7: Mean correlation between a similarity measure and all other similarity measure implementations

Method (first step)	*Method (second step)*	*Data type*	*Mean correlation*
First moment	L1 distance	Raw	0.7244
First moment	L2 distance	Raw	0.7143
First moment	Pearson correlation distance	Raw	0.3955
First moment	L1 distance	Boolean	0.7641
First moment	L2 distance	Boolean	0.7590
First moment	Pearson correlation distance	Boolean	0.5580
Second moment	L1 distance	Raw	0.6643
Second moment	L2 distance	Raw	0.6642
Second moment	Pearson correlation distance	Raw	0.5195
Second moment	L1 distance	Boolean	0.7801
Second moment	L2 distance	Boolean	0.7778
Second moment	Pearson correlation distance	Boolean	0.6509
Equal-sized bins	L1 distance	Raw	0.6930
Equal-sized bins	L2 distance	Raw	0.6143
Equal-sized bins	Pearson correlation distance	Raw	0.5695
Equal-sized bins	Symmetric Kullback-Leibler divergence	Raw	NA
Equal-sized bins	L1 earth mover's distance	Raw	0.7729
Equal-sized bins	L2 earth mover's distance	Raw	0.7701
Equal-sized bins	L1 distance	Boolean	0.7093
Equal-sized bins	L2 distance	Boolean	0.6639
Equal-sized bins	Pearson correlation distance	Boolean	0.6874
Equal-sized bins	Symmetric Kullback-Leibler divergence	Boolean	NA
Equal-sized bins	L1 earth mover's distance	Boolean	0.7680
Equal-sized bins	L2 earth mover's distance	Boolean	0.7625
Sum of squares (10 bins)	L1 distance	Raw	0.6253
Sum of squares (10 bins)	L2 distance	Raw	0.6427
Sum of squares (10 bins)	Pearson correlation distance	Raw	0.4989
Sum of squares (10 bins)	Symmetric Kullback-Leibler divergence	Raw	0.7000
Sum of squares (10 bins)	L1 earth mover's distance	Raw	0.7547
Sum of squares (10 bins)	L2 earth mover's distance	Raw	0.7397
Sum of squares (10 bins)	L1 distance	Boolean	0.6820
Sum of squares (10 bins)	L2 distance	Boolean	0.6458
Sum of squares (10 bins)	Pearson correlation distance	Boolean	0.5756
Sum of squares (10 bins)	Symmetric Kullback-Leibler divergence	Boolean	0.6799
Sum of squares (10 bins)	L1 earth mover's distance	Boolean	0.7554
Sum of squares (10 bins)	L2 earth mover's distance	Boolean	0.7430
Sum of squares (50 bins)	L1 distance	Raw	0.6915
Sum of squares (50 bins)	L2 distance	Raw	0.6981
Sum of squares (50 bins)	Pearson correlation distance	Raw	0.6670
Sum of squares (50 bins)	Symmetric Kullback-Leibler divergence	Raw	0.7523
Sum of squares (50 bins)	L1 earth mover's distance	Raw	0.7686
Sum of squares (50 bins)	L2 earth mover's distance	Raw	0.7603
Sum of squares (50 bins)	L1 distance	Boolean	0.7088
Sum of squares (50 bins)	L2 distance	Boolean	0.6639
Sum of squares (50 bins)	Pearson correlation distance	Boolean	0.6844
Sum of squares (50 bins)	Symmetric Kullback-Leibler divergence	Boolean	NA
Sum of squares (50 bins)	L1 earth mover's distance	Boolean	0.7681
Sum of squares (50 bins)	L2 earth mover's distance	Boolean	0.7625

Tree	L1 distance	Raw	0.4112
Tree	L2 distance	Raw	0.4129
Tree	Pearson correlation distance	Raw	0.0905
Tree	Symmetric Kullback-Leibler divergence	Raw	0.4938
Tree	L1 earth mover's distance	Raw	0.4049
Tree	L2 earth mover's distance	Raw	0.4087
Tree	L1 distance	Boolean	0.6277
Tree	L2 distance	Boolean	0.6364
Tree	Pearson correlation distance	Boolean	0.5641
Tree	Symmetric Kullback-Leibler divergence	Boolean	0.5693
Tree	L1 earth mover's distance	Boolean	0.6459
Tree	L2 earth mover's distance	Boolean	0.6367
Pure L1 earth mover's distance		Raw	0.7144
Pure L2 earth mover's distance		Raw	0.7296
First moment	*	*	0.6525
Second moment	*	*	0.6762
Equal-sized bins	*	*	0.7011
Sum of squares (10 bins)	*	*	0.6703
Sum of squares (50 bins)	*	*	0.7205
Tree	*	*	0.4918
*	L1 distance	*	0.6735
*	L2 distance	*	0.6578
*	Pearson correlation distance	*	0.5384
*	Symmetric Kullback-Leibler divergence	*	0.6391
*	L1 earth mover's distance†	*	0.7048
*	L2 earth mover's distance†	*	0.6979
*	*	Raw	0.6151
*	*	Boolean	0.6868
*	*	*	0.6491
†does not include pure earth mover's distance			

With some exceptions the similarity measures are moderately to highly correlated with each other. The mean correlation between similarity measures is 0.6491. Similarity measures based on the tree method or Pearson correlation distance tend to correlate more poorly with other measures. If similarity measures based on Pearson correlation distance are excluded, the mean correlation between similarity measures rises to 0.7105. If similarity measures based on the tree method are excluded, the mean correlation rises to 0.7564. And if both are excluded, the mean correlation rises to 0.8326.

Table 8 shows the mean correlation between similarity measures in a certain family. Because certain families disproportionately contain measures based on Pearson correlation or trees, the results excluding these similarity measures are also shown.

Table 8: Mean correlation within a family of similarity measures

Family	*Mean correlation*	*Excluding measures based on Pearson correlation*	*Excluding measures based on trees*
First moment	0.6447	0.8887	NA
Second moment	0.6910	0.8528	NA
Equal-sized bins	0.8675	0.8895	NA
Sum of squares (10 bins)	0.7844	0.8606	NA
Sum of squares (50 bins)	0.8399	0.8718	NA
Tree	0.7470	0.8640	NA
L1 distance	0.6885	NA	0.7853
L2 distance	0.6617	NA	0.7724
Pearson correlation distance	0.4872	NA	0.6010
Symmetric Kullback-Leibler	0.6468	NA	0.8581
L1 earth mover's distance†	0.7687	NA	0.9579
L2 earth mover's distance†	0.7314	NA	0.9418

†does not include pure earth mover's distance

L1 earth mover’s distance and L2 earth mover’s distance are remarkably consistent when the tree method is excluded. For this dataset the tree method created only 3 bins, whereas other probability mass function methods created between 10 and 128 bins, which may explain why the tree method gives somewhat different results.

III.D. Testing resistance to assay variation using dataset 2

Dataset 2 consists of flow cytometry samples from 17 monkeys. The samples were stained with FITC perforin, PerCP CD4, PE-Cy7 CD28, Alexa 700 CD3, Pacific Blue CD95 and AmCyan CD8 [12]. FSC and SSC were used to create a gate for lymphocytes. Only cells from the lymphocyte gate are included in the dataset. FSC and SSC are not included in the dataset. The dataset contains 30,000 cells from each sample.

Dataset 2 was used to test resistance to assay variation. Whereas all the samples in dataset 1 were processed together and hence assay variation in that dataset is quite low, the samples in dataset 2 were processed in two batches. The first batch (samples 1 through 9) was processed several months before the second batch (samples 10 through 17). Not surprisingly assay variation between the two batches is quite substantial. In particular FITC perforin was much brighter in the first batch than in the second batch. (Section I.D discusses this problem in more detail.) Other sources of assay variation may also be present.

In order to test the effect of assay variation on a similarity measure, an assay variation score $AVS=\Sigma(C(i)-D(i))$ was devised, where C(i) is the mean similarity rank between sample i and other sample from the same batch and D(i) is the mean similarity rank between sample i and samples from the other batch. Here the similarity rank SR between two samples, X and Y, means that Y is SR(X,Y)th most similar sample to X. Note that SR(X,X)=1 and SR may not be symmetric even if the similarity measure it is based on is symmetric. Also note that C excludes the similarity rank between a sample and itself. For this dataset the minimum assay variation score is -136. Permutation tests indicate that if assay variation is absent or a similarity measure is unaffected by assay variation then the expected value of the assay variation score is zero or very close to zero.

In other words if a similarity measure is unaffected by assay variation, we would expect a sample to be as similar to other samples in its batch as it is to samples in a different batch. But when raw data was used, samples were judged to be much more similar to samples in their own batch. For example table 9 shows the similarity rank matrix produced by the first moment method and L1 distance using raw data. The (i,j) cell of this matrix contains the similarity rank between sample i and sample j. According to this matrix, every sample is more similar to every sample in its own batch than it is to any sample in the other batch. This situation corresponds to the lowest possible assay variation score of -136. Needless to say this situation is very unlikely to occur by chance alone (p-value = 9!*8!/17! = 0.00004114). When Boolean gate data is used, the situation is greatly improved. Table 10 shows the similarity matrix produced by the first moment method and L1 distance using Boolean gate data.

Table 9: Similarity matrix produced by first moment method and L1 distance using raw data

1	6	5	9	8	3	7	4	2	11	10	14	13	15	16	17	12
3	1	8	4	9	5	7	2	6	15	10	17	12	11	16	13	14
5	9	1	7	4	3	2	8	6	11	10	13	14	15	16	17	12
6	7	5	1	9	3	4	2	8	17	14	12	15	13	10	16	11
6	9	2	7	1	4	3	8	5	11	10	13	14	15	16	17	12
4	9	3	7	6	1	2	8	5	11	10	13	15	14	16	17	12
5	9	2	7	4	3	1	8	6	11	10	13	15	14	16	17	12
4	2	8	3	9	5	7	1	6	12	10	15	17	16	11	14	13
2	8	5	9	4	3	6	7	1	11	10	13	14	15	16	17	12
11	17	10	16	13	14	12	15	9	1	2	4	5	6	7	8	3
9	13	11	17	15	14	12	16	10	2	1	6	3	4	8	7	5
11	17	10	9	16	13	12	15	14	6	7	1	3	4	5	8	2
9	14	11	10	17	13	12	15	16	8	5	3	1	2	6	7	4
10	11	12	9	17	14	13	15	16	8	5	3	2	1	6	7	4
11	14	16	9	17	12	15	10	13	8	7	3	5	4	1	2	6
11	12	16	9	17	13	15	10	14	8	7	5	4	3	2	1	6
9	17	11	10	16	13	12	14	15	5	6	2	3	4	7	8	1

Table 10: Similarity matrix produced by first moment method and L1 distance using Boolean gate data

1	10	8	15	3	14	2	4	6	9	11	12	16	17	7	13	5
8	1	3	9	10	15	11	7	12	14	6	5	17	16	13	4	2
9	3	1	8	11	15	13	7	14	12	4	6	17	16	10	5	2
14	4	3	1	16	10	17	9	15	13	6	5	12	8	11	2	7
5	9	11	16	1	12	7	3	2	6	10	13	15	17	4	14	8
16	13	14	15	11	1	17	9	10	4	5	3	2	7	6	8	12
2	8	9	15	4	13	1	3	6	10	11	12	16	17	7	14	5
5	9	10	16	2	13	7	1	3	11	8	12	15	17	4	14	6
6	11	12	16	2	10	7	3	1	5	9	13	15	17	4	14	8
9	15	11	17	4	5	13	8	6	1	3	7	12	16	2	14	10
15	9	6	16	14	8	17	10	11	7	1	2	13	12	3	5	4
16	5	6	13	15	4	17	12	14	9	2	1	11	10	8	3	7
16	13	14	12	15	3	17	9	10	8	5	4	1	2	7	6	11
16	10	12	7	15	4	17	13	14	11	6	3	2	1	9	5	8
7	15	12	17	5	9	11	6	4	2	3	8	14	16	1	13	10
15	5	6	2	16	9	17	13	14	12	4	3	10	8	11	1	7
6	3	2	14	9	15	8	5	11	13	4	7	17	16	12	10	1

Overall similarity measures based on Boolean gate data are much more resistant to assay variation than similarity measures based on raw data. Table 11 shows the assay variation score for all similarity measures tested in section III.C except similarity measures based on the tree method. These measures were excluded because no suitable supervisor is available for this dataset. This exclusion is unfortunate since a supervised method may be more resistant to assay variation as it might ignore regions or dimensions like FITC perforin where assay variation is greatest.

The equal-sized bins method was implemented with 2^6 bins. The sum of squares method was implemented with 10 bins and with 50 bins. Pure earth mover's distance was implemented using the first 256 cells from each sample.

Table 11: Assay variation scores

Method (first step)	*Method (second step)*	*Data type*	*Assay variation score*
First moment	L1 distance	Raw	-136
First moment	L2 distance	Raw	-136
First moment	Pearson correlation distance	Raw	-136
First moment	L1 distance	Boolean	-25.8
First moment	L2 distance	Boolean	-22.8
First moment	Pearson correlation distance	Boolean	-8
Second moment	L1 distance	Raw	-136
Second moment	L2 distance	Raw	-136
Second moment	Pearson correlation distance	Raw	-136
Second moment	L1 distance	Boolean	-32.1
Second moment	L2 distance	Boolean	-25.8
Second moment	Pearson correlation distance	Boolean	-10.3
Equal-sized bins	L1 distance	Raw	-132
Equal-sized bins	L2 distance	Raw	-106.4
Equal-sized bins	Pearson correlation distance	Raw	-109.1
Equal-sized bins	Symmetric Kullback-Leibler divergence	Raw	NA
Equal-sized bins	L1 earth mover's distance	Raw	-98
Equal-sized bins	L2 earth mover's distance	Raw	-104.1
Equal-sized bins	L1 distance	Boolean	-38.4
Equal-sized bins	L2 distance	Boolean	-28.2
Equal-sized bins	Pearson correlation distance	Boolean	-28.3
Equal-sized bins	Symmetric Kullback-Leibler divergence	Boolean	NA
Equal-sized bins	L1 earth mover's distance	Boolean	-28.8
Equal-sized bins	L2 earth mover's distance	Boolean	-30.1
Sum of squares (10 bins)	L1 distance	Raw	-136
Sum of squares (10 bins)	L2 distance	Raw	-136
Sum of squares (10 bins)	Pearson correlation distance	Raw	-136
Sum of squares (10 bins)	Symmetric Kullback-Leibler divergence	Raw	-84.6
Sum of squares (10 bins)	L1 earth mover's distance	Raw	-136
Sum of squares (10 bins)	L2 earth mover's distance	Raw	-136
Sum of squares (10 bins)	L1 distance	Boolean	-34.7
Sum of squares (10 bins)	L2 distance	Boolean	-29.7
Sum of squares (10 bins)	Pearson correlation distance	Boolean	-21.1
Sum of squares (10 bins)	Symmetric Kullback-Leibler divergence	Boolean	-29.3
Sum of squares (10 bins)	L1 earth mover's distance	Boolean	-24.1
Sum of squares (10 bins)	L2 earth mover's distance	Boolean	-26.4
Sum of squares (50 bins)	L1 distance	Raw	-136
Sum of squares (50 bins)	L2 distance	Raw	-136

Sum of squares (50 bins)	Pearson correlation distance	Raw	-136
Sum of squares (50 bins)	Symmetric Kullback-Leibler divergence	Raw	NA
Sum of squares (50 bins)	L1 earth mover's distance	Raw	-136
Sum of squares (50 bins)	L2 earth mover's distance	Raw	-136
Sum of squares (50 bins)	L1 distance	Boolean	-38.4
Sum of squares (50 bins)	L2 distance	Boolean	-28.3
Sum of squares (50 bins)	Pearson correlation distance	Boolean	-28.4
Sum of squares (50 bins)	Symmetric Kullback-Leibler divergence	Boolean	NA
Sum of squares (50 bins)	L1 earth mover's distance	Boolean	-28.8
Sum of squares (50 bins)	L2 earth mover's distance	Boolean	-30.1
Pure L1 earth mover's distance		Raw	-134.7
Pure L2 earth mover's distance		Raw	-136
*	*	Raw	-128.4
*	*	Boolean	-27.2

The relatively high assay variation score (-84.6) for the sum of squares method (10 bins) with Kullback-Leibler divergence using raw data is entirely attributable to infinite divergences between certain samples in the same batch. The equal-sized bins method performed slightly better with raw data than other methods but was still very biased.

III.E. Conclusions

Overall Boolean gate data is preferable to raw data. Raw data is acceptable only if assay variation appears to be low. On the other hand Boolean gate data is just one possible transformation of raw data. There will be other transformations of raw data that are resistant to assay variation, and some of these transformations may be preferable to Boolean gate data in certain ways.

A probability mass function method with earth mover’s distance is the best similarity measure for most situations. Earth mover’s distance outperformed Lp distance, symmetric Kullback-Leibler divergence and Pearson correlation in terms of consistency and accuracy. Furthermore if earth mover’s distance is used with an unsupervised probability mass function method, a reference multiset is not necessary. Earth mover’s distance can compare probability mass functions with different data bins. Therefore an unsupervised binning method can be applied individually to each multiset. This approach removes one of the biggest shortcomings of probability mass function methods.

Because earth mover's distance gives extremely similar results for all unsupervised probability mass function methods, the consistency and accuracy of the unsupervised probability mass function methods is not very important. For most situations the equal-sized bins method is the best unsupervised method. It is conceptually the simplest. It is particularly attractive for Boolean gate data since Boolean gate data is naturally divided into equal-sized bins, and it is also adequate for raw data. But if the dimensionality of the dataset is very high, it may not be advisable to create s^d bins (where d is the dimensionality of the dataset and s is an integer greater than 1). In this case the sum of squares method may be preferable.

If an appropriate supervisor is available, the tree method with earth mover's distance may be a good option. The tree method appears to be more accurate than unsupervised probability mass function methods. And for certain types of data, for example very high dimensional data where many of the dimensions are unimportant, the tree method or some other supervised method may be the only acceptable option. On the other hand a similarity measure based the tree method may ignore regions that are not linked to the supervisor but nonetheless important in some way.

If a very simple similarity measure is desired, the first moment method with L1 or L2 distance is a good option. This measure outperformed several more complicated measures in terms of accuracy, consistency and other properties. Furthermore unlike other methods this method can be used even if certain variables are never observed together (see section II.B for more information).

Pure earth mover's distance performed well on the nearest-neighbor test but has some undesirable qualities. In particular it does not obey the perfect similarity property or the size property, and it can be very computationally intensive. Furthermore since pure earth mover's distance only compares the first N points of each multiset, it is not an appropriate similarity measure if the first N points are not representative of the multiset. Such a situation is likely to occur if the points are ordered chronologically and there is a chronological trend or if the data has been sorted. In situations like these, it would be preferable to randomly selected N points, but this approach would create a random similarity measure.

Pearson correlation distance and symmetric Kullback-Leibler divergence are particularly poor options. Pearson correlation distance performed terribly on the nearest-neighbor test, and similarity measures based on Pearson correlation correlated relatively poorly with other similarity measures. Symmetric Kullback-Leibler often produced infinite similarity scores and has no redeeming feature to compensate for this shortcoming. In retrospect the square root of Jensen-Shannon divergence would have been preferable to symmetric Kullback-Leibler divergence. The Jensen-Shannon divergence between two random variables X and Y is equal to KL(X,(X+Y)/2)+KL(Y,(X+Y)/2) where KL is Kullback-Leibler divergence. Jensen-Shannon divergence is guaranteed to be finite, and the square root of Jensen-Shannon divergence obeys the triangle inequality. On the other hand I doubt that Jensen-Shannon divergence would match the consistency of earth mover's distance, and unlike earth mover's distance Jensen-Shannon divergence cannot compare probability mass functions with different data bins.

IIII. Diversity measures

IIII.A. Two types of diversity measures

Diversity measures quantify the diversity of an object. Consider the following two multisets: A={1,2,3,4,5,6,7,8} and B={1,1,1,1,8,8,8,8}. Which of these two multisets is more diverse? At first glance the first multiset may appear to be more diverse since it contains eight distinct numbers whereas the second multiset contains only 1s and 8s. Now consider these two multisets: C={8,8.01,8.02,8.03,8.04,8.05,8.06,8.07} and D={0,0,0,0,1000,1000,1000,1000}. Once again the first multiset contains eight distinct numbers and the second multiset has only two values. But since all the numbers in the first multiset are very similar to each other and 0 and 1000 are not so similar, the second multiset may now seem more diverse. In fact the answer to these questions depends on the type of diversity measure that is used.

In this paper I will discuss two types of diversity measures: “nominal diversity measures” and “interval diversity measures”. Nominal diversity measures treat any elements that are not identical or in the same category as being equally different. If the elements of multiset C are all judged to fall into different categories, then according to a nominal diversity measure multiset, C is more diverse than multiset D. Interval diversity measures allow multiple (>2) levels of similarity between elements. In this paper all interval diversity measures are based on the average similarity score between elements in a multiset. The higher the average similarity score the more diverse the multiset is. Hence there is an underlying connection between similarity measures and certain diversity measures.

IIII.B. Nominal diversity measures

Shannon entropy is one of the most commonly used diversity measures. The Shannon entropy of A is $\Sigma(A(i)*log(A(i)))$ if A is a probability mass function and $\int A(x)*log(A(x))\,dx$ if A is a probability density function.

The Simpson's index of A is normally defined as $\Sigma(A(i)\hat{}2)$ if A is a probability mass function and as $\int A(x)\hat{}2\,dx$ if A is a probability density function. Because this form of Simpon's index decreases as diversity increases, two alternate forms of Simpson's index are commonly used: 1-D and 1/D where D is the normal form of Simpson's index. This paper uses the first of these alternate forms.

In order to use Shannon entropy or Simpon's index to compare two multisets of points, the multiset must first be represented as probability density functions or probability mass functions. Techniques for doing this are discussed in section are discussed in section II. Kernel density estimation can be used to create probability density functions for each multiset, and the data binning methods from section II.D can be used to create probability mass functions.

Note that the Simpson's index of a multiset A is equivalent to the mean similarity between elements in A if similarity S is defined as S(X,Y) = {0 if X and Y are in the same category, 1 otherwise}.

Because nominal diversity measures judge any elements not in the same category to be equally dissimilar, they can contradict intuitive notions of diversity. For example if all the elements in the multiset {9.1, 9.3, 9.4, 9.5, 9.53, 9.6, 9.7, 9.8} are judged to fall in different categories, then this multiset is more diverse than {4, 9.8, 11.1, 12.4, 14.6, 14.6, 17.9, 18.5}.

IIII.C. Interval diversity measures

Interval diversity measures for multisets of points can be created by calculating the average similarity between points in a multiset. In practice any similarity measure that is appropriate for points and has more than two output levels could be used, and any central tendency (e.g. mean or median) could be used to calculate the average of these similarity measures. (A similarity measure with exactly two output levels could also be used, but a nominal diversity measure would be created.)

Calculating the average similarity between all points in a multiset can be computationally intensive. If a multiset contains N points, then N*(N-1)/2 similarity scores must be calculated (assuming the similarity measure is symmetric and obeys the perfect similarity property). One alternative is to select M points from the multiset and calculate the average similarity between these points. If the first M points can reasonably be considered a random sample, these points can be used. If not, M points can be randomly selected from each multiset, although this would result in a random diversity measure. The mean Lp distance between 100-1000 points can be computed fairly quickly on a modern computer, and 100-1000 randomly selected points are likely to be an accurate representation of the multiset. Another alternative is to use a data binning method and approximate each point's location using the center of its data bin. In this case only B*(B-1)/2 similarity scores would need to be calculated where B is the number of bins.

If the mean squared L2 distance is used to measure diversity, then diversity can be calculated quite efficiently. In this case diversity is equal to twice the mean squared L2 distance from mean of the multiset. Hence only N distances need to be calculated, where N is the number of points in the multiset. (The mean squared L2 distance is also equal to the trace of the maximum-likelihood variance matrix for the multiset.

The trace is the sum of the diagonal elements of a matrix, and the trace of a variance matrix can be used as a single measure of dispersion for multidimensional data.) Unfortunately this diversity measure can contradict intuitive notions of diversity. Consider the following multisets: {(-1,-1),(-1,1),(1,-1),(1,1)} and {(-1,-1),(-1,-1),(1,1),(1,1)}. Intuitively the first dataset is more diverse since it has 4 distinct populations instead of two. Furthermore the mean Lp distance between points is greater for the first multiset than for the second. But according to mean squared L2 distance, both multisets are equally diverse.

IIII.D. Assessing diversity measures using dataset 1

The diversity of dataset 1 was calculated using the following diversity measures: Shannon entropy, Simpson's index, mean L1 distance, mean L2 distance and mean squared L2 distance. Shannon entropy, Simpson's index, mean L1 distance and mean L2 distance were calculated for each of the probability mass function methods tested in section III.C. Mean L1 distance and mean L2 distance were also calculated using the first 1024 points from each sample.

Table 12 shows the mean correlation between a diversity measure implementation and all other diversity measure implementations and the correlation between that implementation and log SIV viral load. The entire correlation matrix can be found in the supplementary tables file. Mean correlations exclude the necessarily perfect correlation between an implementation and itself.

Table 12: Comparison of diversity measures

Diversity measure	*Binning method or subset*	*Data type*	*Mean correlation with all other diversity measure implementations*	*Correlation with log viral load*
Shannon entropy	Equal-sized bins	Raw	0.9111	-0.6415
Shannon entropy	Equal-sized bins	Boolean	0.8972	-0.5861
Shannon entropy	Sum of squares (10 bins)	Raw	0.9176	-0.4650
Shannon entropy	Sum of squares (10 bins)	Boolean	0.9096	-0.5598

Shannon entropy	Sum of squares (50 bins)	Raw	0.8305	-0.1869
Shannon entropy	Sum of squares (50 bins)	Boolean	0.9008	-0.5908
Shannon entropy	Tree method	Raw	0.7792	-0.7080
Shannon entropy	Tree method	Boolean	0.8957	-0.6818
Simpson's index	Equal-sized bins	Raw	0.8884	-0.6296
Simpson's index	Equal-sized bins	Boolean	0.8739	-0.5903
Simpson's index	Sum of squares (10 bins)	Raw	0.8937	-0.4171
Simpson's index	Sum of squares (10 bins)	Boolean	0.8800	-0.5876
Simpson's index	Sum of squares (50 bins)	Raw	0.8233	-0.1791
Simpson's index	Sum of squares (50 bins)	Boolean	0.8743	-0.5905
Simpson's index	Tree method	Raw	0.7735	-0.7363
Simpson's index	Tree method	Boolean	0.9220	-0.6565
Mean L1 distance	First 1024 points	Raw	0.9015	-0.4937
Mean L1 distance	First 1024 points	Boolean	0.9349	-0.5497
Mean L1 distance	Equal-sized bins	Raw	0.9403	-0.6482
Mean L1 distance	Equal-sized bins	Boolean	0.9349	-0.5497
Mean L1 distance	Sum of squares (10 bins)	Raw	0.9124	-0.6089
Mean L1 distance	Sum of squares (10 bins)	Boolean	0.9256	-0.5306
Mean L1 distance	Sum of squares (50 bins)	Raw	0.9119	-0.5561
Mean L1 distance	Sum of squares (50 bins)	Boolean	0.9350	-0.5502
Mean L1 distance	Tree method	Raw	0.7726	-0.7347
Mean L1 distance	Tree method	Boolean	0.9280	-0.6162
Mean L2 distance	First 1024 points	Raw	0.9084	-0.5592
Mean L2 distance	First 1024 points	Boolean	0.9284	-0.5786
Mean L2 distance	Equal-sized bins	Raw	0.9434	-0.6405
Mean L2 distance	Equal-sized bins	Boolean	0.9284	-0.5786
Mean L2 distance	Sum of squares (10 bins)	Raw	0.9120	-0.5914
Mean L2 distance	Sum of squares (10 bins)	Boolean	0.9208	-0.5475
Mean L2 distance	Sum of squares (50 bins)	Raw	0.8991	-0.5588
Mean L2 distance	Sum of squares (50 bins)	Boolean	0.9288	-0.5794
Mean L2 distance	Tree method	Raw	0.7744	-0.7294
Mean L2 distance	Tree method	Boolean	0.9260	-0.6396
Mean squared L2 distance	All points	Raw	0.8978	-0.5868
Mean squared L2 distance	All points	Boolean	0.9349	-0.5497
Shannon entropy	*	*	0.8802	-0.5525
Simpson's index	*	*	0.8661	-0.5484
Mean L1 distance	*	*	0.9097	-0.5838
Mean L2 distance	*	*	0.9070	-0.6003
Mean squared L2 distance	*	*	0.9164	-0.5683
*	Equal-sized bins	*	0.9147	-0.6081
*	Sum of squares (10 bins)	*	0.9090	-0.5385
*	Sum of squares (50 bins)	*	0.8880	-0.4740
*	Tree method	*	0.8464	-0.6878
*	First 1024 points	*	0.9183	-0.5453
*	*	Raw	0.8732	-0.5617
*	*	Boolean	0.9147	-0.5849
*	*	*	0.8940	-0.5733

Overall the diversity measures are highly correlated with each other. This suggests that, even though interval and nominal diversity measures are conceptually quite different, in practice they can give similar results. In fact the mean correlation between interval diversity measures and nominal diversity measures is 0.8847, which is almost identical to the overall mean correlation of 0.8940. For most diversity measures there is a moderate negative correlation between diversity and log SIV viral load, meaning that monkeys with a more diverse lymphocyte population had lowers levels of virus. Diversity measures based on the tree method tend to correlate more strongly with log viral load, which suggests that supervised binning may result in more meaningful diversity measures.

IIII.E. Conclusions

Overall interval diversity measures are preferable to nominal diversity measures. Nominal diversity measures can produce counterintuitive results when applied to multisets of points (see section IIII.B). If a nominal diversity measures are used, the data binning strategy must be carefully chosen. For example imagine that Roederer's method [6] is used to create 10 equal-weight bins for each multiset in a dataset (without the use of a reference multiset). The Simpsons index of each multiset will equal 1-10*0.1^2=0.9, and the Shannon entropy of each multiset will equal 10*-0.1*log(0.1)=2.302585. All multiset will have equal diversity regardless of their contents.

Not all interval diversity measures are appropriate. Mean squared L2 distance is computationally efficient. But it too can produce counter-intuitive results (see section IIII.C).

Of the methods tested in section IIII.D, mean Lp distance is the best. For smaller multisets (fewer than 1000 points) the mean Lp distance between all points can be calculated in less than a minute on most modern computers. For larger multisets mean Lp distance can be approximated by calculating mean Lp distance for a subset of the multiset or by using a data binning method. A supervised data binning method, like the tree method, may even increase the meaningfulness of a diversity measure by identifying important regions and dimensions.

This paper only examines a few diversity measures, but it also provides a framework for creating new diversity measures. Any similarity measure can be combined with an appropriate central tendency to create a diversity measure. By identifying an appropriate similarity measure and central tendency, it should be possible to create a suitable diversity measure for many types of data.

V. Endnotes

1. Diaz-Romero J, Romeo S, Bovée JV, Hogendoorn PC, Heini PF, Mainil-Varlet P. “Hierarchical clustering of flow cytometry data for the study of conventional central chondrosarcoma”. Journal of cellular physiology. 2010.

2. Kaufman M, Bloch D, Zurgil N, Shafran Y, Deutsch M. "A cluster pattern algorithm for the analysis of multiparametric cell assays". Journal of computational biology. 2005.

3. Finn WG, Carter KM, Raich R, Stoolman LM, Hero AO. "Analysis of clinical flow cytometric immunophenotyping data by clustering on statistical manifolds: treating flow cytometry data as high-dimensional objects". Cytometry part b. 2008.

4. Carter KM, Raich R, Finn WG , and Hero AO. "Information preserving component analysis: data projections for flow cytometry analysis". arXiv. 2008.

5. Carter KM, Raich R, Finn WG, Hero AO. "Dimensionality reduction of flow cytometric data through information preservation". www.eecs.umich.edu/~hero/Preprints/carter_mlsp_08.pdf. 2008.

6. Roederer M, Moore W, Treister A, Hardy RR, Herzenberg LA. "Probability binning comparison: a metric for quantitating multivariate distribution differences". Cytometry. 2001.

7. Light emitted by a fluorescent dye is often detected not only by the detector of that dye but also by other detectors. This phenomenon is known as spillover. For example light emitted by FITC is often detected by the PE detector. This spillover could erroneously make it seem that a cell-stained with FITC is also stained with PE. In other to prevent this problem most flow cytometers will adjust the data to reduce the effect of spillover.

8. Rubner Y, Tomasi C, Guibas LJ. "The earth mover's distance as a metric for image retrieval". International journal of computer vision. 2000.

9. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. Springer. 2001. Chapter 14.3.6 discusses k-means clustering. Chapter 9.2 discusses trees. Chapter 8.9 discusses bumping. The version of bumping described by Hastie et al uses bootstrapping but not random cost. This book is available at www-stat.stanford.edu/~tibs/ElemStatLearn.

10. Cost refers to the cost of partitioning a dimension. If the cost for each dimension is equal, the regression tree algorithm will pick the binary partition that minimizes the sum of the squared error. If the costs are not equal, the algorithm will pick the partition for which the reduction in the sum of the squared error divided by the cost is greatest.

11. The samples were also stained with APC (allophycocyanin) Gag-CM9 tetramer, but this stain was ignored. Only Mamu-A*01 monkeys have cells that should be stainable with this dye. Including this dye would have caused diversity measures to overestimate the diversity of samples from Mamu-A*01 monkeys compared to samples from other monkeys. For consistency the APC Gag-CM9 stain was also ignored when calculating similarity.

12. The samples were also stained with APC Gag-CM9 tetramer and PE CCR5, but these stains were ignored. PE CCR5 was ignored because some samples were stained with a defective PE CCR5 dye. APC Gag-CM9 tetramer was ignored for reasons discussed in endnote 11. The decision to exclude these stains was made before any similarity or diversity scores were calculated.

VI. R Code

```
#Setup

library(rpart)
library(lpSolve)
library(clue)

#Dataset 1

vl=log(c(113353.40,2634.60,100.00,595.20,104215.60,111000.00,6206.60,5106.60))
#vl is the log viral load where vl[1] is the log viral load for the first
sample, vl[2] is the log viral load for the second sample, etc.

raw=read.csv("dataset1_raw.csv",header=T)
bg=read.csv("dataset1_Boolean.csv",header=T)
raw$wt=1 #a uniform weight is added so that the format of raw matches the
format of bg, the functions below are programmed to expect this format

#Dataset 2

RAW=read.csv("dataset2_raw.csv",header=T)
BG=read.csv("dataset2_Boolean.csv",header=T)
RAW$wt=1

#Counts cells in each data bin
```

```
fcount=function(x){
 c=ncol(x)
 ID=unique(x[,1])
 SS=unique(x[,c])
 n=length(ID)
 m=length(SS)
 y=matrix(0,n,m)
 for(i in 1:n){
  for(j in 1:m){y[i,j]=sum(subset(x,x[,1]==ID[i]&x[,c]==SS[j])[,c-1])}
 }
 for(i in 1:n){y[i,]=y[i,]/sum(y[i,])}
 return(y)
}

#Calculates the center of each data bin

fcent=function(x){
 c=ncol(x)
 SS=unique(x[,c])
 m=length(SS)
 y=matrix(0,m,c-3)
 for(i in 1:m){
  s=subset(x,x[,c]==SS[i])
  y[i,]=cov.wt(s[,-c(1,c-1,c)],s[,c-1])$center
 }
 return(y)
}

#Calculates Lp distance

flp=function(x,exp){
 n=nrow(x)
 y=matrix(1,n,n)
 for(i in 1:n){
  for(j in 1:n){y[i,j]=(sum(abs(x[i,]-x[j,])^exp))^(1/exp)}
 }
 return(y)
}

#Calculates Pearson correlation distance

fpc=function(x){
 n=nrow(x)
 y=matrix(1,n,n)
 for(i in 1:n){
  for(j in 1:n){y[i,j]=1-cov(x[i,],x[j,])/sqrt(var(x[i,])*var(x[j,]))}
 }
 return(y)
}

#Calculates symmetric Kullback-Leibler divergence

fkl=function(x){
 n=nrow(x)
 y=matrix(1,n,n)
 for(i in 1:n){
  for(j in 1:n){y[i,j]=sum(x[i,]*log(x[i,]/x[j,]),na.rm=T)}
 }
 y=y+t(y)
 return(y)
}

#Calculates earth mover's distance
#exp=1 gives L1 earth mover's distance, exp=2 gives L2 earth mover's distance,
etc.

femd=function(x,centers,exp){
 r=nrow(x)
 c=ncol(x)
 y=matrix(0,r,r)
 cost=as.matrix(dist(centers,method="minkowski",diag=T,upper=T,p=exp))
 for(i in 1:r){
  for(j in 1:r){
   a=x[i,]/sum(x[i,])
   b=x[j,]/sum(x[j,])
   source=pmax(a-b,0)
   destination=pmax(b-a,0)

y[i,j]=lp.transport(cost,direction="min",row.signs=rep("=",c),row.rhs=source,c
ol.sign=rep("=",c),col.rhs=destination,integers=NULL)$objval
  }
 }
 return(y)
}

#Tests similarity scores based on dataset 1

ftest=function(l){
 ll=l
 for(i in 1:nrow(l)){ll[i,]=rank(l[i,],ties.method="first")}
 cor2=cor((ll==2)%*%vl,vl)
 cor3=cor((ll==3)%*%vl,vl)
 cor4=cor((ll==4)%*%vl,vl)
 cor5=cor((ll==5)%*%vl,vl)
 cor6=cor((ll==6)%*%vl,vl)
 cor7=cor((ll==7)%*%vl,vl)
 cor8=cor((ll==8)%*%vl,vl)

return(list(l=l,ll=ll,cor2=cor2,cor3=cor3,cor4=cor4,cor5=cor5,cor6=cor6,cor7=c
or7,cor8=cor8))
}

#Tests susceptibility to assay variation using dataset 2

fTest=function(l){
 ll=l
 for(i in 1:nrow(l)){ll[i,]=rank(l[i,])}
 score=(sum(ll[1:9,1:9])-9)/8+(sum(ll[10:17,10:17])-8)/7-sum(ll[1:9,10:17])/8-
sum(ll[10:17,1:9])/9
 return(list(l=l,ll=ll,score=score))
}
```

```
#First moment method
#ffm calculates the first moment for each sample

ffm=function(x){
 n1=max(x[,1])
 n2=ncol(x)
 y=matrix(1,n1,n2-2)
 for(i in 1:n1){
  s=subset(x,x[,1]==i)
  y[i,]=cov.wt(s[,-c(1,n2)],wt=s[,n2])$center
 }
 return(y)
}

fr=ffm(raw)
fr1=ftest(flp(fr,1)); fr1
fr2=ftest(flp(fr,2)); fr2
frp=ftest(fpc(fr)); frp

fb=ffm(bg)
fb1=ftest(flp(fb,1)); fb1
fb2=ftest(flp(fb,2)); fb2
fbp=ftest(fpc(fb)); fbp

#Second moment method
#fsm calculates the second moment for each sample

fsm=function(x){
 n1=max(x[,1])
 n2=ncol(x)
 y=matrix(1,n1,(n2-2)^2)
 for(i in 1:n1){
  s=subset(x,x[,1]==i)
  cov=cov.wt(s[,-c(1,n2)],wt=s[,n2],method="ML")
  y[i,]=c(cov$cen%*%t(cov$cen)+cov$cov)
 }
 return(y)
}

sr=fsm(raw)
sr1=ftest(flp(sr,1)); sr1
sr2=ftest(flp(sr,2)); sr2
srp=ftest(fpc(sr)); srp

sb=fsm(bg)
sb1=ftest(flp(sb,1)); sb1
sb2=ftest(flp(sb,2)); sb2
sbp=ftest(fpc(sb)); sbp

#Equal-sized bins method
#fesb assigns rows of x to data bins based on equal-sized binning, each
dimension is divided d times

fesb=function(x,d){
 n=ncol(x)
 min1=apply(x[,-c(1,n)],2,min)
 max1=apply(x[,-c(1,n)],2,max)
 vector=d^(0:(n-3))
 m=t((t(x[,-c(1,n)])-min1)/(max1-min1))
 m=floor(d*m)
 m=ifelse(m==d,d-1,m)
 x$subset=m%*%vector
 return(x)
}

er0=fesb(raw,2)
er=fcount(er0[,c(1,9,10)])
er.c=fcent(er0)
er1=ftest(flp(er,1)); er1
er2=ftest(flp(er,2)); er2
erp=ftest(fpc(er)); erp
erk=ftest(fkl(er)); erk
ere=ftest(femd(er,er.c,1)); ere
erE=ftest(femd(er,er.c,2)); erE

eb=t(matrix(bg$wt,c(128,8))) #Boolean gate in compact form is already divided
into equal-sized bins, therefore there is no reason to use fesb, fcount or
fcent
eb.c=bg[1:128,-c(1,9)]
eb1=ftest(flp(eb,1)); eb1
eb2=ftest(flp(eb,2)); eb2
ebp=ftest(fpc(eb)); ebp
ebk=ftest(fkl(eb)); ebk
ebe=ftest(femd(eb,eb.c,1)); ebe
ebE=ftest(femd(eb,eb.c,2)); ebE

#Sum of squares method
#fss assigns rows in x to bins based on sum of squares binning, n is the
number of bins

fss=function(x,n){
 c=ncol(x)
 x$s=1
 for(i in 2:n){
  v=NULL
  for(j in 1:(i-1)){
   s=subset(x,x$s==j)
   v=c(v,diag(cov.wt(s[,-c(1,c,c+1)],s[,c],method="ML")$cov)*sum(s[,c]))
  }
  w=which(v==max(v))[1]
  ws=ceiling(w/(c-2))
  wd=w-(c-2)*(ws-1)
  s=subset(x,x$s==ws)
  mean1=cov.wt(s[,-c(1,c,c+1)],s[,c])$center[wd]
  x$s=ifelse(x$s==ws&x[,1+wd]>mean1,i,x$s)
 }
 return(x)
}

zr0=fss(raw,10)
zr=fcount(zr0[,c(1,9,10)])
zr.c=fcent(zr0)
```

```
zr1=ftest(flp(zr,1)); zr1
zr2=ftest(flp(zr,2)); zr2
zrp=ftest(fpc(zr)); zrp
zrk=ftest(fkl(zr)); zrk
zre=ftest(femd(zr,zr.c,1)); zre
zrE=ftest(femd(zr,zr.c,2)); zrE

zb0=fss(bg,10)
zb=fcount(zb0[,c(1,9,10)])
zb.c=fcent(zb0)
zb1=ftest(flp(zb,1)); zb1
zb2=ftest(flp(zb,2)); zb2
zbp=ftest(fpc(zb)); zbp
zbk=ftest(fkl(zb)); zbk
zbe=ftest(femd(zb,zb.c,1)); zbe
zbE=ftest(femd(zb,zb.c,2)); zbE

Zr0=fss(raw,50)
Zr=fcount(Zr0[,c(1,9,10)])
Zr.c=fcent(Zr0)
Zr1=ftest(flp(Zr,1)); Zr1
Zr2=ftest(flp(Zr,2)); Zr2
Zrp=ftest(fpc(Zr)); Zrp
Zrk=ftest(fkl(Zr)); Zrk
Zre=ftest(femd(Zr,Zr.c,1)); Zre
ZrE=ftest(femd(Zr,Zr.c,2)); ZrE

Zb0=fss(bg,50)
Zb=fcount(Zb0[,c(1,9,10)])
Zb.c=fcent(Zb0)
Zb1=ftest(flp(Zb,1)); Zb1
Zb2=ftest(flp(Zb,2)); Zb2
Zbp=ftest(fpc(Zb)); Zbp
Zbk=ftest(fkl(Zb)); Zbk
Zbe=ftest(femd(Zb,Zb.c,1)); Zbe
ZbE=ftest(femd(Zb,Zb.c,2)); ZbE

#Tree method
#ftree links rows of x to a supervisor and calculates the resulting tree; s is
the supervisor where s[1] is the supervisor for the first sample, s[2] is the
supervisor for the second sample, etc.; $where can be used to assign rows to
data bins

ftree=function(x,s){
 ts=NULL
 n=max(x[,1])
 m=ncol(x)
 for(i in 1:n){ts=c(ts,rep(s[i],nrow(subset(x,x[,1]==i))))}
 y=rpart(ts~.,x[,-c(1,m)],weights=x[,m],method="anova")
 return(y)
}

tr0=ftree(raw,vl)
tr=fcount(cbind(raw$id,raw$wt,tr0$where))
tr.c=fcent(cbind(raw,tr0$where))
tr1=ftest(flp(tr,1)); tr1
```

```
tr2=ftest(flp(tr,2)); tr2
trp=ftest(fpc(tr)); trp
trk=ftest(fkl(tr)); trk
tre=ftest(femd(tr,tr.c,1)); tre
trE=ftest(femd(tr,tr.c,2)); trE

tb0=ftree(bg,vl)
tb=fcount(cbind(bg$id,bg$wt,tb0$where))
tb.c=fcent(cbind(bg,tb0$where))
tb1=ftest(flp(tb,1)); tb1
tb2=ftest(flp(tb,2)); tb2
tbp=ftest(fpc(tb)); tbp
tbk=ftest(fkl(tb)); tbk
tbe=ftest(femd(tb,tb.c,1)); tbe
tbE=ftest(femd(tb,tb.c,2)); tbE

#Pure earth mover's distance
#by default the first 256 rows of each sample are used

fpemd=function(x,exp,m=256){
 x=x[,-ncol(x)]
 n=max(x[,1])
 y=matrix(0,n,n)
 for(i in 1:n){
  for(j in 1:n){
   a=subset(x,x[,1]==i)[1:m,-1]
   b=subset(x,x[,1]==j)[1:m,-1]

cost=as.matrix(dist(rbind(a,b),method="minkowski",diag=T,upper=T,p=exp))[1:m,(
m+1):(2*m)]
   sol=solve_LSAP(cost)
   y[i,j]=sum(cost[cbind(seq_along(sol),sol)])/m
  }
 }
 return(y)
}

pre=ftest(fpemd(raw,1)); pre
prE=ftest(fpemd(raw,2)); prE

#Similarity measures comparison

x=-c(1,10,19,28,37,46,55,64)
sim=cor(cbind(fr1$l[x],fr2$l[x],frp$l[x],fb1$l[x],fb2$l[x],fbp$l[x],sr1$l[x],s
r2$l[x],srp$l[x],sb1$l[x],sb2$l[x],sbp$l[x],er1$l[x],er2$l[x],erp$l[x],erk$l[x
],ere$l[x],erE$l[x],eb1$l[x],eb2$l[x],ebp$l[x],ebk$l[x],ebe$l[x],ebE$l[x],zr1$
l[x],zr2$l[x],zrp$l[x],zrk$l[x],zre$l[x],zrE$l[x],zb1$l[x],zb2$l[x],zbp$l[x],z
bk$l[x],zbe$l[x],zbE$l[x],Zr1$l[x],Zr2$l[x],Zrp$l[x],Zrk$l[x],Zre$l[x],ZrE$l[x
],Zb1$l[x],Zb2$l[x],Zbp$l[x],Zbk$l[x],Zbe$l[x],ZbE$l[x],tr1$l[x],tr2$l[x],trp$
l[x],trk$l[x],tre$l[x],trE$l[x],tb1$l[x],tb2$l[x],tbp$l[x],tbk$l[x],tbe$l[x],t
bE$l[x],pre$l[x],prE$l[x]))
names1=c("fr1","fr2","frp","fb1","fb2","fbp","sr1","sr2","srp","sb1","sb2","sb
p","er1","er2","erp","erk","ere","erE","eb1","eb2","ebp","ebk","ebe","ebE","zr
1","zr2","zrp","zrk","zre","zrE","zb1","zb2","zbp","zbk","zbe","zbE","Zr1","Zr
2","Zrp","Zrk","Zre","ZrE","Zb1","Zb2","Zbp","Zbk","Zbe","ZbE","tr1","tr2","tr
p","trk","tre","trE","tb1","tb2","tbp","tbk","tbe","tbE","pre","prE")
```

```
rownames(sim)=names1
colnames(sim)=names1
write.csv(sim,"similarityComp.csv")

#Testing sensitivity to assay variation using dataset 2

fR=ffm(RAW)
fR1=fTest(flp(fR,1)); fR1
fR2=fTest(flp(fR,2)); fR2
fRp=fTest(fpc(fR)); fRp

fB=ffm(BG)
fB1=fTest(flp(fB,1)); fB1
fB2=fTest(flp(fB,2)); fB2
fBp=fTest(fpc(fB)); fBp

sR=fsm(RAW)
sR1=fTest(flp(sR,1)); sR1
sR2=fTest(flp(sR,2)); sR2
sRp=fTest(fpc(sR)); sRp

sB=fsm(BG)
sB1=fTest(flp(sB,1)); sB1
sB2=fTest(flp(sB,2)); sB2
sBp=fTest(fpc(sB)); sBp

eR0=fesb(RAW,2)
eR=fcount(eR0[,c(1,8,9)])
eR.c=fcent(eR0)
eR1=fTest(flp(eR,1)); eR1
eR2=fTest(flp(eR,2)); eR2
eRp=fTest(fpc(eR)); eRp
eRk=fTest(fkl(eR)); eRk
eRe=fTest(femd(eR,eR.c,1)); eRe
eRE=fTest(femd(eR,eR.c,2)); eRE

eB=t(matrix(BG$wt,c(64,17)))
eB.c=BG[1:64,-c(1,8)]
eB1=fTest(flp(eB,1)); eB1
eB2=fTest(flp(eB,2)); eB2
eBp=fTest(fpc(eB)); eBp
eBk=fTest(fkl(eB)); eBk
eBe=fTest(femd(eB,eB.c,1)); eBe
eBE=fTest(femd(eB,eB.c,2)); eBE

zR0=fss(RAW,10)
zR=fcount(zR0[,c(1,8,9)])
zR.c=fcent(zR0)
zR1=fTest(flp(zR,1)); zR1
zR2=fTest(flp(zR,2)); zR2
zRp=fTest(fpc(zR)); zRp
zRk=fTest(fkl(zR)); zRk
zRe=fTest(femd(zR,zR.c,1)); zRe
zRE=fTest(femd(zR,zR.c,2)); zRE

zB0=fss(BG,10)
```

```
zB=fcount(zB0[,c(1,8,9)])
zB.c=fcent(zB0)
zB1=fTest(flp(zB,1)); zB1
zB2=fTest(flp(zB,2)); zB2
zBp=fTest(fpc(zB)); zBp
zBk=fTest(fkl(zB)); zBk
zBe=fTest(femd(zB,zB.c,1)); zBe
zBE=fTest(femd(zB,zB.c,2)); zBE

ZR0=fss(RAW,50)
ZR=fcount(ZR0[,c(1,8,9)])
ZR.c=fcent(ZR0)
ZR1=fTest(flp(ZR,1)); ZR1
ZR2=fTest(flp(ZR,2)); ZR2
ZRp=fTest(fpc(ZR)); ZRp
ZRk=fTest(fkl(ZR)); ZRk
ZRe=fTest(femd(ZR,ZR.c,1)); ZRe
ZRE=fTest(femd(ZR,ZR.c,2)); ZRE

ZB0=fss(BG,50)
ZB=fcount(ZB0[,c(1,8,9)])
ZB.c=fcent(ZB0)
ZB1=fTest(flp(ZB,1)); ZB1
ZB2=fTest(flp(ZB,2)); ZB2
ZBp=fTest(fpc(ZB)); ZBp
ZBk=fTest(fkl(ZB)); ZBk
ZBe=fTest(femd(ZB,ZB.c,1)); ZBe
ZBE=fTest(femd(ZB,ZB.c,2)); ZBE

pRe=fTest(fpemd(RAW,1)); pRe
pRE=fTest(fpemd(RAW,2)); pRE

#Permutation test for assay variation score

m=fR1$l
scores=rep(0,10000)
for(i in 1:10000){
 order=sample(1:17)
 mm=m[order,order]
 scores[i]=fTest(mm)$score
}
summary(scores)

n=fB1$l
scores2=rep(0,10000)
for(i in 1:10000){
 order=sample(1:17)
 nn=n[order,order]
 scores2[i]=fTest(nn)$score
}
summary(scores2)

#Shannon entropy

fshan=function(x){
 s=-x*log(x)
 y=0
 for(i in 1:nrow(x)){y[i]=sum(s[i,],na.rm=T)}
 return(y)
}

her=fshan(er)
heb=fshan(eb)
hzr=fshan(zr)
hzb=fshan(zb)
hZr=fshan(Zr)
hZb=fshan(Zb)
htr=fshan(tr)
htb=fshan(tb)

#Simpson's index

fsimp=function(x){
 s=x^2
 y=0
 for(i in 1:nrow(x)){y[i]=1-sum(s[i,])}
 return(y)
}

ier=fsimp(er)
ieb=fsimp(eb)
izr=fsimp(zr)
izb=fsimp(zb)
iZr=fsimp(Zr)
iZb=fsimp(Zb)
itr=fsimp(tr)
itb=fsimp(tb)

#Mean Lp distance

fmld=function(x,exp,n=1024){
 c=ncol(x)-1
 y=0
 for(i in 1:max(x[,1])){
  s=subset(x,x[,1]==i)[,-1]
  s=s[1:min(nrow(s),n),]
  m=as.matrix(dist(s[,-c],method="minkowski",diag=T,upper=T,p=exp))
  w=as.matrix(s[,c])
  y[i]=t(w)%*%m%*%w/sum(w)^2
 }
 return(y)
}

mxr=fmld(raw,1)
mxb=fmld(bg,1)
Mxr=fmld(raw,2)
Mxb=fmld(bg,2)

#Mean Lp distance using data bins

fmld2=function(x,centers,exp){
 y=0
 distance=as.matrix(dist(centers,method="minkowski",diag=T,upper=T,p=exp))
 for(i in 1:nrow(x)){
  w=as.matrix(x[i,])
  y[i]=t(w)%*%distance%*%w/sum(w)^2
 }
 return(y)
}

mer=fmld2(er,er.c,1)
meb=fmld2(eb,eb.c,1)
mzr=fmld2(zr,zr.c,1)
mzb=fmld2(zb,zb.c,1)
mZr=fmld2(Zr,Zr.c,1)
mZb=fmld2(Zb,Zb.c,1)
mtr=fmld2(tr,tr.c,1)
mtb=fmld2(tb,tb.c,1)

Mer=fmld2(er,er.c,2)
Meb=fmld2(eb,eb.c,2)
Mzr=fmld2(zr,zr.c,2)
Mzb=fmld2(zb,zb.c,2)
MZr=fmld2(Zr,Zr.c,2)
MZb=fmld2(Zb,Zb.c,2)
Mtr=fmld2(tr,tr.c,2)
Mtb=fmld2(tb,tb.c,2)

#Mean squared L2 distance

fmsld=function(x){
 y=0
 c=ncol(x)-1
 for(i in 1:max(x[,1])){
  s=subset(x,x[,1]==i)[-1]
  y[i]=sum(diag(cov.wt(s[,-c],wt=s[,c],method="ML")$cov))
 }
 return(y)
}

sxr=fmsld(raw)
sxb=fmsld(bg)

#Diversity measures comparison

div=cor(cbind(her,heb,hzr,hzb,hZr,hZb,htr,htb,ier,ieb,izr,izb,iZr,iZb,itr,itb,
mxr,mxb,mer,meb,mzr,mzb,mZr,mZb,mtr,mtb,Mxr,Mxb,Mer,Meb,Mzr,Mzb,MZr,MZb,Mtr,Mt
b,sxr,sxb,vl))
names2=c("her","heb","hzr","hzb","hZr","hZb","htr","htb","ier","ieb","izr","iz
b","iZr","iZb","itr","itb","mxr","mxb","mer","meb","mzr","mzb","mZr","mZb","mt
r","mtb","Mxr","Mxb","Mer","Meb","Mzr","Mzb","MZr","MZb","Mtr","Mtb","sxr","sx
b","vl")
rownames(div)=names2
colnames(div)=names2
write.csv(div,"diversityComp.csv")
###
```