# Mutual Information between Discrete and Continuous Data Sets

**Brian C. Ross***

Department of Physics, University of Washington, Seattle, Washington, United States of America

## Abstract

Mutual information (MI) is a powerful method for detecting relationships between data sets. There are accurate methods for estimating MI that avoid problems with "binning" when both data sets are discrete or when both data sets are continuous. We present an accurate, non-binning MI estimator for the case of one discrete data set and one continuous data set. This case applies when measuring, for example, the relationship between base sequence and gene expression level, or the effect of a cancer drug on patient survival time. We also show how our method can be adapted to calculate the Jensen–Shannon divergence of two or more data sets.

**Citation:** Ross BC (2014) Mutual Information between Discrete and Continuous Data Sets. PLoS ONE 9(2): e87357. doi:10.1371/journal.pone.0087357

**Editor:** Daniele Marinazzo, Universiteit Gent, Belgium

**Received** November 26, 2013; **Accepted** December 19, 2013; **Published** February 19, 2014

**Funding:** Funding for this work was provided by National Institutes of Health/National Human Genome Research Institute grant R01HG005115. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The author has declared that no competing interests exist.

* E-mail: heltilda@uw.edu

## Introduction

Mutual information (MI) [1] is in several ways a perfect statistic for measuring the degree of relatedness between data sets. First, MI will detect any sort of relationship between data sets whatsoever, whether it involves the mean values or the variances or higher moments. Second, MI has a straightforward interpretation as the amount of shared information between data sets (measured in, for example, bits); other statistics such as rank-ordering are harder to interpret. Since MI is grounded in information theory it has an established base of theoretical tools. Finally, MI is insensitive to the size of the data sets. Whereas a 'p-value' test for strict independence can be pushed arbitrarily low by taking a large data set if the variables are even slightly related, MI will simply converge with tight error bounds to a measure of their relatedness.

The MI between two data sets $X$ and $Y$ can be estimated from the statistics of the $(x,y)$ pairs between the two data sets. (Although MI is straightforward to calculate if the underlying probability distribution is known, that is not usually the case: our knowledge of the distribution generally comes from the sampled data itself, so MI must be estimated from the statistics of our data set.) For example, if we were to compare the day of week ($X$) with the time of breakfast ($Y$) we might find that when $x_i$ is a weekday the corresponding $y_i$ is early in the morning, and when $x_i$ is Sunday or (especially) Saturday the corresponding $y_i$ is somewhat later. MI quantifies the strength of this effect. Importantly, the procedure for estimating MI depends on whether $X$ and $Y$ take discrete values (e.g. a day of week, a nucleobase, a phenotypic category, etc.), or are real-valued continuous variables (a time of day, a gene expression level, a patient's survival time, etc.). If $X$ and $Y$ are both discrete, then we can estimate the true frequencies of all combinations of $(x,y)$ pairs by counting the number of times each pair occurs in the data, and straightforwardly use these frequencies to estimate MI. Real-valued data sets are more difficult to deal with, since they are by definition sparsely sampled: most real numbers will not be found in a data set of any size. The common workaround is to lump the continuous variables into discrete 'bins' and then apply a discrete MI estimator, but good sampling requires large bins which destroys resolution. An improved continuous-continuous MI estimator described in Ref. [2] circumvents this tradeoff by using statistics of the spacings between data points and their nearest neighbors. Crucially, their method only works when *both* variables are real-valued, as the nearest neighbor of a discrete variable is not well-defined.

This paper describes a method for estimating the MI between a discrete data set and a continuous (scalar or vector) data set, using a similar approach to that of Ref. [2]. This is an important statistic simply because so many scientific activities involve a search for significant relationships between discrete and continuous variables. For example, one might use MI to quantify the extent to which nationality (a discrete variable) determines income (continuous); to identify DNA bases (ACGT, discrete) that affect a given gene's expression level (continuous); or to find drugs (given or not: a discrete parameter) that alter cell division rates (continuous data). In the University of Washington Nanopore Physics lab we use this estimator to determine where a given DNA base must sit within the sequencing pore in order to affect the current passing through it, and to quantify the relative influence of different base positions on the current. As we will demonstrate, our nearest-neighbors method estimates MI much more reliably than does the present alternative method of 'binning' the data.

MI between a discrete and a continuous variable is equivalent to a weighted form of the Jensen-Shannon (JS) divergence [3] which is used as a measure of the dissimilarity between two or more continuous probability distributions. We can therefore apply our method to estimate the weighted JS divergence, by storing samples from each distribution to be compared in the continuous data set $Y$, and using the discrete data set $X$ to identify which distribution each sample was drawn from. To use our method to estimate the *unweighted* JS divergence, we would either draw equal numbers of samples from each distribution, or else modify our method somewhat as explained in the Analysis section.

## Methods

This section explains how to apply our nearest-neighbor method for estimating MI; the derivation is left to the Analysis section. We will also describe the binning method that we compare with our estimator.

The input to a MI estimator is a list of $(x,y)$ data points, whose underlying probability distribution $\mu(x,y)$ we can only guess at by looking at how the data points are clustered. Both $x$ and $y$ may be either scalars or vectors. Figure 1A illustrates a simple distribution between a discrete parameter $x$ that can take one of three values denoted by color, and a single scalar real-valued variable $y$ depicted along a y-axis. In this example we see that the different values of $x$ bias the sampling towards different values of $y$: for example $y$ is generally lower when $x$ is green or red than when $x$ is blue. Therefore there is a relation between $x$ and $y$, implying that MI is some positive number. The challenge is to estimate MI using only the sampled points that are known to the experimenter (Figure 1B).

### Nearest Neighbor Method

For each data point $i$ our method computes a number $I_i$ based on its nearest-neighbors in the continuous variable $y$, as illustrated for scalar $y$ in Figure 1C. We first find the $k$th-closest neighbor to point $i$ among those $N_{x_i}$ data points whose value of the discrete variable equals $x_i$ (Figure 1C, bottom line) using some distance metric of our choice. Define $d$ as the distance to this $k$th neighbor. We then count the number of neighbors $m_i$ in the full data set (top line) that lie within distance $d$ to point $i$ (including the $k$th neighbor itself). Based on $N_{x_i}$ and $m_i$ we compute

$$I_i=\psi(N)-\psi(N_{x_i})+\psi(k)-\psi(m_i) \tag{1}$$

where $\psi(\cdot)$ is the digamma function [4]. To estimate the MI from our data set, we average $I_i$ over all data points.

$$\begin{aligned} I(X,Y)&=\langle I_i\rangle \\ &=\psi(N)-\langle\psi(N_x)\rangle+\psi(k)-\langle\psi(m)\rangle \end{aligned} \tag{2}$$

In our implementation $k$ is some fixed (low) integer of the user's choice; larger $k$-values lead to lower sampling error but higher coarse-graining error.

### Binning Method

We also implemented a binning method to compare with our nearest-neighbor method. Binning methods make the data completely discrete by grouping the data points into bins in the continuous variable $y$, as shown in Figure 1D. Following established practice [2] our estimator constructs bins of different sizes so that each bin has $n$ data points inside it ($n$ is a parameter set by the user). The binned approximation to the MI is

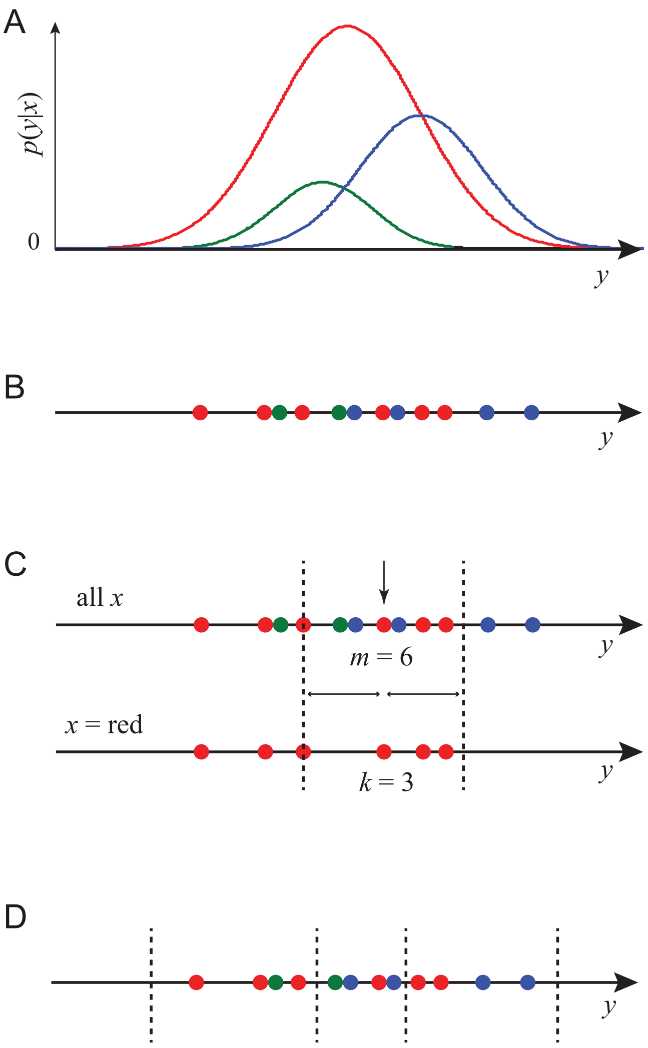

**Figure 1. Procedures for estimating MI.** (A) An example joint probability density $\mu(x,y)$ where $y$ is a real-valued scalar and $x$ can take one of three values, indicated red, blue and green. For each value of $x$ the probability density in $y$ is shown as plot of that color, whose area is proportional to $p(x)$. (B) A set of $(x,y)$ data pairs sampled from this distribution, where $x$ is represented by the color of each point and $y$ by its position on the $y$-axis. (C) The computation of $I_i$ in our nearest-neighbor method. Data point $i$ is the red dot indicated by a vertical arrow. The full data set is on the upper line, and the subset of all red data points is on the lower line. We find that the data point which is the 3rd-closest neighbor to $i$ on the bottom line is the 6th-closest neighbor on the top line. Dashed lines show the distance $d$ from point $i$ out to the 3rd neighbor. $N=12$, $k=3$, and for this point $N_{x_i}=6$ and $m_i=6$. (D) A binning of the data into equal bins containing $n=4$ data points. MI can be estimated from the numbers of points of each color in each bin.
doi:10.1371/journal.pone.0087357.g001

$$\begin{aligned} I(X,Y)&=\left\langle \log\frac{\bar{\mu}(x_i,y_i)}{p(x_i)\bar{\mu}(y_i)}\right\rangle_i \\ &=\left\langle \log\frac{p(x_i,b_i)}{p(x_i)p(b_i)}\right\rangle_i. \end{aligned} \tag{3}$$

The average is taken over all measurements $i$, not the bins. $p(x_i)$ is the fraction of all measurements whose discrete variable is $x_i$, $p(b_i)$ is the fraction of measurements whose continuous variable falls into the same bin $b_i$ as $y_i$, and $p(x_i,b_i)$ is the fraction of measurements for which $x=x_i$ and $y$ falls into bin $b_i$. The second line in Eq. 3 follows from the first because we discretize $\mu(y)$ and $\mu(x,y)$ using the same bins.

In the Supporting Information we have included two MATLAB implementations of our method: a general-purpose estimator that works with vector-valued data sets, and a faster implementation for the usual case where both data sets are scalars (simple numbers). The Supporting Information also contains our implementation of a MI estimator using the binning method, as well as the testing script that compares the three estimators and generated the plots for this paper.

## Results

To test our method, we chose two simple distributions $\mu(x,y)$: a square wave distribution in $y$ for each value in $x$, and a Gaussian distribution in $y$ for each $x$ (Figure 2A). Because we knew the exact form of the distributions, we were able to calculate MI exactly using its mathematical definition:

$$I(X,Y)=\sum_x \int \log \frac{\mu(x,y)}{p(x)\mu(y)} dy. \qquad (4)$$

Next, from each distribution, we constructed test data sets by randomly sampling a certain number $N$ of $(x,y)$ data pairs. We then independently estimated MI from those data sets using our nearest-neighbor estimator and also using our binning estimator, and compared those estimates to each other and to the exact result. We also compared the MI estimate between our vector and scalar implementations of the nearest-neighbor method. Their results in all cases are in exact agreement with each other. This is a

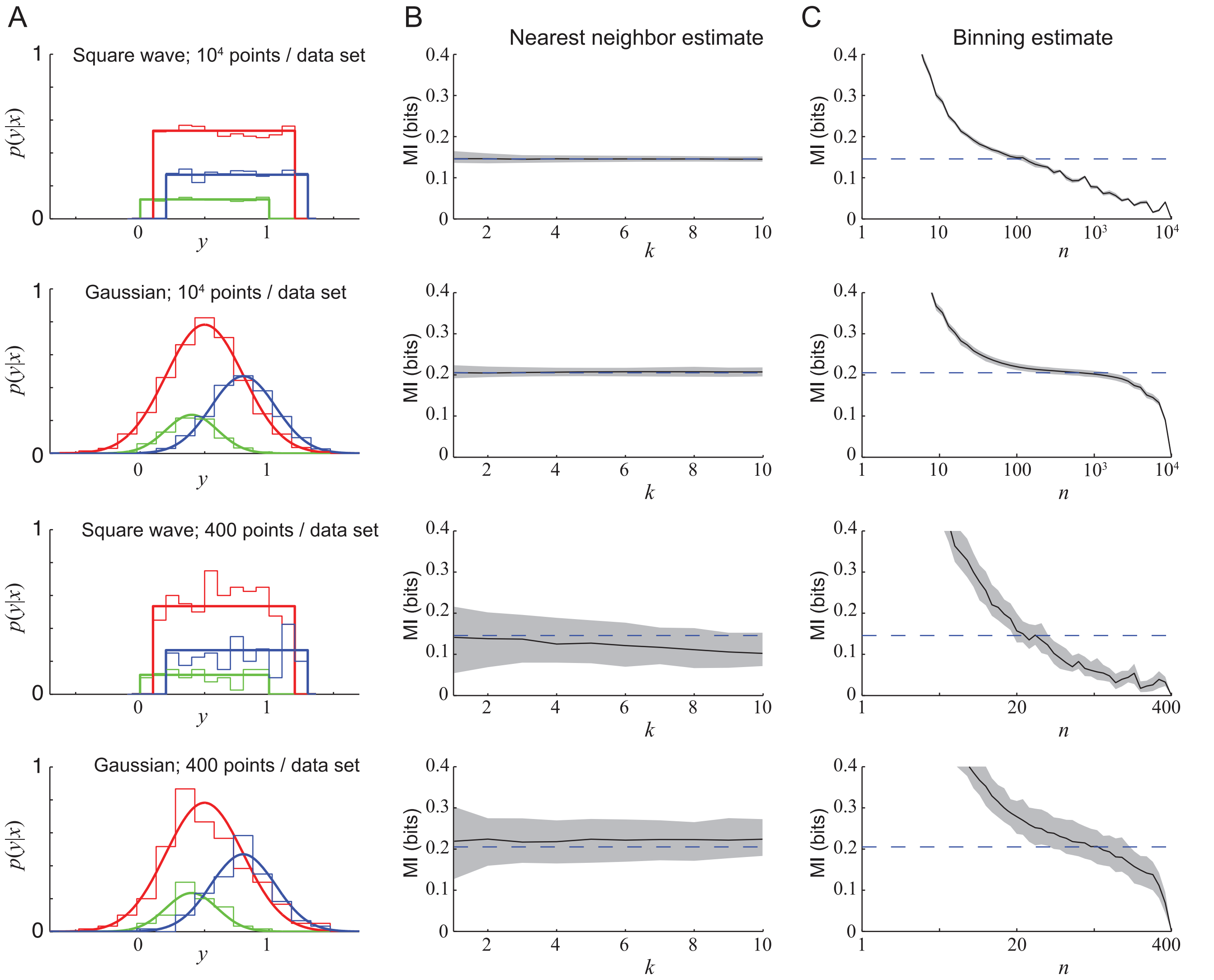

**Figure 2. MI estimated by nearest-neighbors versus binning.** (A) Sampling distributions $\mu(x,y)$ (thick lines) represented by a differently-colored graph in $y$ for each of three possible values of the discrete variable $x$ (red, blue and green). A histogram of a representative data set for each distribution is overlaid using a thinner line. (B) MI estimates as a function of $k$ using the nearest-neighbor estimator. 100 data sets were constructed for each distribution, and the MI of each data set was estimated separately for different values of $k$. The median MI estimate of the 100 data sets for each $k$-value is shown with a black line; the shaded region indicates the range (lowest 10% to highest 10%) of MI estimates. (C) MI estimates plotted as a function of bin size $n$ using the binning method (right panel), using the same 100 data sets for each distribution. The black line shows the median MI estimate of the 100 data sets for each $n$-value; the shaded region indicates the 10%–90% range
doi:10.1371/journal.pone.0087357.g002

strong check that the scripts were written correctly, since the two estimators were coded quite differently.

Both the nearest-neighbor method and the binning method involve a somewhat arbitrary parameter that must be set by the user. The nearest neighbor method requires that the user specify $k$ (the $k$th neighbor). $k$ should be some low integer, much less than the number of data points $N$, so Figure 2B plots MI estimated by nearest neighbors over the range $1 \leq k \leq 10$. Likewise, the binning method requires that the user specify the number of data points $n$ per bin. It is less obvious what the best value of $n$ should be; Figure 2C plots MI estimated by binning over all possible values $1 \leq n \leq N$.

Our first conclusion is that there is a much simpler prescription for setting the $k$ parameter of the nearest-neighbor estimator than the $n$ parameter of the binning method. The nearest-neighbor estimator consistently gives good results when $k$ is set to a low integer. Reference [2] suggests using $k=3$, and that choice works well with our estimator too. By contrast, the binning estimator overestimates MI when $n$ is low and underestimates MI when $n$ is high, and although there is guaranteed to be a crossing point where the method is accurate it is hard to guess where that point might be. (In the limit $n=1$ the binning method estimates MI to be the entropy of the discrete variable. The actual MI only attains this maximum limit if the sub-distributions $\mu(y|x)$ are all completely separated in $y$. In the limit $n=N$ the binning method estimates MI to be zero.).

Our second conclusion is that there is *no* simple way to calculate the optimal binning parameter $n$ based on simple statistics of the data, such as the total number of data points $N$ or frequencies with which different discrete symbols occur. For example, the large Gaussian data sets and the large square-wave data sets each have 10000 data points per set, with twice as many red points as blue points on average, and five times more reds than greens. But the best value of $n$ is ~100 for the square-wave data set and ~600 for the Gaussian data sets. This is easiest to see in Figure 3A, which plots the ratio of the median binning error using given $n$ to the median nearest-neighbors error using $k=3$. We find that there is no choice of $n$ for which binning is better than nearest-neighbors for both the square wave and Gaussian data sets. Figure 3B shows roughly the same result for the 400-point data sets, which again are statistically similar except in the shape of their distributions in $y$.

We conclude that MI estimation by the nearest neighbor method is far more accurate than binning-based MI estimates, barring a lucky guess of the unknowable best value of $n$. Furthermore, our nearest-neighbor method is computationally cheap: both computation time and memory usage are proportional to $N$ for the scalar estimator. Therefore nearest neighbors should be the method of choice for estimating MI in the discrete-continuous case.

## Analysis

Here we derive the formula for our nearest-neighbor MI estimator.

Consider a discrete variable $X$ and the continuous variable $Y$, drawn from probability density $\mu(x,y)$. Both $X$ and $Y$ may be either univariate (composed of scalars) or multivariate (vectors). We will write discrete probability functions as $p(\cdot)$ and continuous densities using the symbol $\mu(\cdot)$: therefore $p(x) = \int \mu(x,y)dy$ and $\mu(y) = \sum_x \mu(x,y)$. The mutual information is:

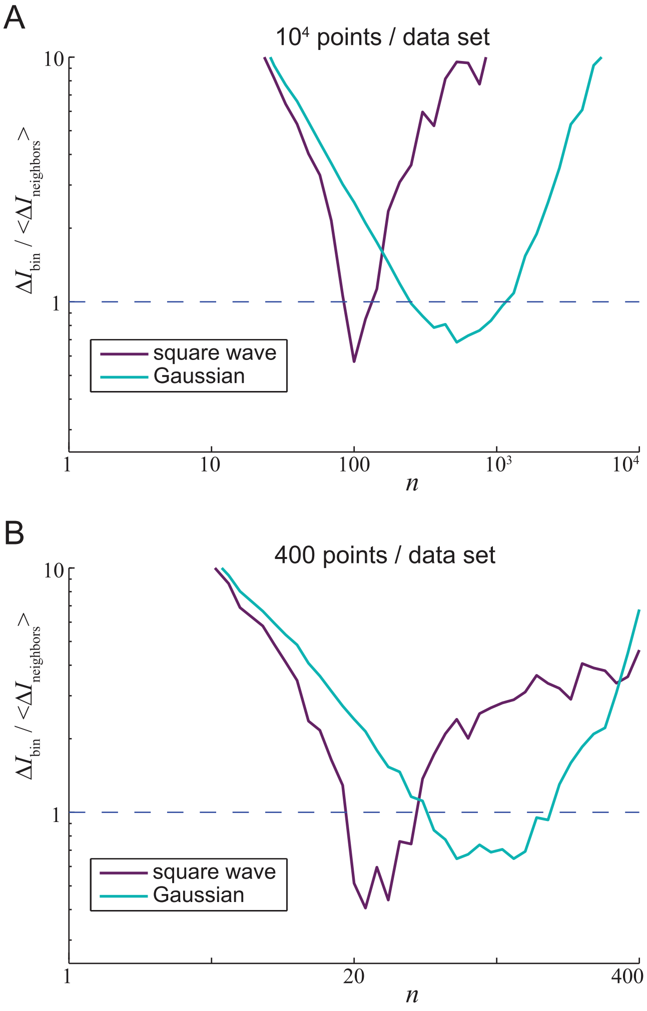

**Figure 3. Binning error relative to nearest-neighbors error.** (A) Error from the binning method divided by error from the nearest-neighbor method. Errors in MI were calculated for each of the 100 data sets of the square-wave (light blue) and Gaussian (purple) 10,000-length data sets (see Figure 2). Each line shows the ratio of the median MI for a given number of neighbors $n$ estimated using binning, as a function of $n$, to the median (over all data sets and all values of $k$) of all MI estimates using nearest neighbors. The binning method gives superior results for values of $n$ for which this ratio is less than one. Evidently, there is no optimal value of $n$ that works for all distributions: $n = N^{0.5}$ works well for the square wave distribution but $n = N^{0.7}$ is better for a Gaussian distribution. (B) MI error using nearest-neigbor method versus binning method for the 400-data point sets.
doi:10.1371/journal.pone.0087357.g003

$$
\begin{aligned}
I(X,Y) &= H(X) + H(Y) - H(X,Y) \\
&= -\sum_x p(x)\log p(x) - \int \mu(y)\log\mu(y)dy \\
&\quad + \sum_x \int \mu(x,y)\log\mu(x,y)dy \qquad (5) \\
&= -\int \mu(y)\log\mu(y)dy + \sum_x \int \mu(x,y)\log\mu(y|x)dy \\
&= -\langle \log\mu(y)\rangle + \langle \log\mu(y|x)\rangle .
\end{aligned}
$$

Here $H$ denotes an entropy, $\mu(y)$ is the probability density for sampling $y$ irrespective of the value of $x$, and $\mu(y|x)=\mu(x,y)/p(x)$ is the probability density for sampling $y$ given a particular value of $x$. The averages are taken over the full distribution and weighted by $\mu(x,y)$, and they would be straightforward to calculate if we knew the underlying density functions. Alternatively, each average can be taken over a representative set from $(x,y)$ pairs sampled from the distribution; using this latter interpretation we estimate the MI from the mean of $\log \mu(y)$ and $\log \mu(y|x)$ at each of our sampled data points. The more points we have, the greater the accuracy.

The remaining task is to estimate the logarithm of two continuous distributions evaluated at given data points. For this we use a nearest-neighbor entropy estimator originally developed by Kozachenko and Leonenko [5] whose proof we will briefly outline. Given a point $i$, we define $V$ as the volume of points centered about $i$ that are closer to point $i$ than its $k$th neighbor. The estimator uses Bayesian arguments to identify $\sigma(\mu_y|V)$ with $\sigma(V|\mu_y)$ ($\sigma$ denotes a probability density that is not to be confused with $\mu$). Approximating the density function $\mu$ as being constant throughout the neighborhood of point $i$, we find:

$$\begin{aligned}\langle \log \mu V \rangle &\approx \frac{\int_0^1 (\mu V)^{k-1}(1-\mu V)^{N-k-1} \log(\mu V) d(\mu V)}{\int_0^1 (\mu V)^{k-1}(1-\mu V)^{N-k-1} d(\mu V)} \\ &= \frac{1}{B(k,N-k)}\frac{dB(k,N-k)}{dk} \\ &= \psi(k)-\psi(N) \\ \rightarrow \langle \log \mu \rangle &\approx \psi(k)-\psi(N)-\log V\end{aligned} \tag{6}$$

where $B(\cdot)$ is the beta function [4] and $\psi(\cdot)$ is the digamma function. We can now estimate the entropy using the full data set:

$$\langle \log \mu(y) \rangle \approx \psi(k)-\psi(N)-\langle \log V \rangle \tag{7}$$

where the average is taken over all sampled data points.

For each sampled data point $i$ we employ the Kozachenko-Leonenko (KL) entropy estimator twice: once to estimate $\mu(y)$ by finding a neighbor from the full set of data points, and once to estimate $\mu(y|x)$ by finding a neighbor in the subset of data points $j$ for which $x_j=x_i$. Notice that we can independently choose the neighbors of the two points: we will pick the $k$th neighbor in the reduced distribution and the $m$th neighbor from the full distribution. The result is

$$\begin{aligned}I(X,Y) \approx \psi(N)-\psi(m)+\langle \log V_{m;y} \rangle \\ -\psi(N_x)+\psi(k)-\langle \log V_{k;y|x} \rangle.\end{aligned} \tag{8}$$

There is a systematic averaging error that comes from the fact that the $k$th-neighbor KL entropy estimator applied to point $i$ necessarily computes the average of $\log \mu(x_i,y_i)$ over the volume $V_k$, rather than evaluated exactly at point $(x_i,y_i)$. Following Ref. [2], we attempt to minimize this error by choosing $k$ and $m$ so that both uses of the KL entropy estimator use the same neighbor $j$. Therefore $V_{m;y}=V_{k;y|x}$ for each data point, and we obtain Eq. 2. The cancellation is only partial; but because the averaging error scales with the number of data pairs as $N^{-2}$ whereas the counting error scales as $N^{-1/2}$, averaging error is generally insignificant except for very small data sets (as we have verified in our tests).

As mentioned before, the mutual information between discrete and continuous data is equivalent to a weighted Jensen-Shannon (JS) divergence between the conditional distributions $\mu(y|x)$, where the frequencies $p(x)$ of the discrete symbols $x$ are the weighting factors. To compute an *unweighted* JS divergence we need to place all the conditional distributions on equal footing irrespective of their frequencies in the data, by weighting each term in the averages in Eq. 5 by the factor $N/N_x c_X$ where $c_X$ is the number of distinct values that $x$ can take. The result is

$$JSD=\psi(N)+\psi(k)-\frac{1}{c_X}\sum_i \frac{\psi(N_{x_i})+\psi(m_i)}{N_{x_i}}. \tag{9}$$

## Supporting Information

**Script S1 Slow (vector) MI calculator.** Estimates MI between two vector or scalar data sets using the nearest-neighbor method. (M)

**Script S2 Fast (scalar) MI calculator.** Estimates MI between two *scalar* data sets using the nearest-neighbor method. (M)

**Script S3 Binning MI calculator.** Estimates MI between two scalar data sets using the binning method. (M)

**Script S4 Testing script.** Compares the methods using sampled data drawn from user-defined distributions. This script was used to generate the plots in this paper. (M)

## Acknowledgments

The author wishes to acknowledge Vikram Agarwal and Walter Ruzzo for helpful discussions, and Andrew Laszlo, Henry Brinkerhoff, Jens Gunlach and Jenny Mae Samson for valuable comments on this paper.

## Author Contributions

Conceived and designed the experiments: BCR. Performed the experiments: BCR. Analyzed the data: BCR. Contributed reagents/materials/analysis tools: BCR. Wrote the paper: BCR.

## References

1. Cover T, Thomas J (1991) Elements of information theory. New York: John Wiley & Sons.
2. Kraskov A, Stögbauer H, Grassberger P (2004) Estimating mutual information. Physical Review E 69: 066138.
3. Grosse I, Bernaola-Galván P, Carpena P, Román-Roldán R, Oliver J, et al. (2002) Analysis of symbolic sequences using the jensen-shannon divergence. Physical Review E 65: 041905.
4. Abramowitz M, Stegun I (1970) Handbook of mathematical functions. New York: Dover Publishing Inc.
5. Kozachenko L, Leonenko NN (1987) Sample estimate of the entropy of a random vector. Problemy Peredachi Informatsii 23: 9–16.