An efficient not-only-linear correlation coefficient based on machine learning

A DOI-citable version of this manuscript is available at https://doi.org/10.1101/2022.06.15.496326.

Authors

- **Milton Pividori**
  0000-0002-3035-4403 · miltondp · miltondp
  Department of Genetics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA · Funded by The Gordon and Betty Moore Foundation GBMF 4552; The National Human Genome Research Institute (R01 HG010067)

- **Marylyn D. Ritchie**
  0000-0002-1208-1720 · MarylynRitchie
  Department of Genetics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA

- **Diego H. Milone**
  0000-0003-2182-4351 · dmilone · d1001
  Research Institute for Signals, Systems and Computational Intelligence (sinc(i)), Universidad Nacional del Litoral, Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Santa Fe CP3000, Argentina

- **Casey S. Greene**
  0000-0001-8713-9213 · cgreene · GreeneScientist
  Center for Health AI, University of Colorado School of Medicine, Aurora, CO 80045, USA; Department of Biochemistry and Molecular Genetics, University of Colorado School of Medicine, Aurora, CO 80045, USA · Funded by The Gordon and Betty Moore Foundation (GBMF 4552); The National Human Genome Research Institute (R01 HG010067); The National Cancer Institute (R01 CA237170)

Abstract

The Clustermatch Correlation Coefficient (CCC) is an efficient, easy-to-use tool for identifying linear and nonlinear patterns in data. CCC is based on machine learning models and reveals biologically meaningful patterns that are missed by standard, linear-only correlation coefficients. Compared to state-of-the-art coefficients such as the Maximal Information Coefficient, CCC is much faster while capturing general patterns in data. Applying CCC to human gene expression data reveals robust linear relationships and nonlinear patterns associated with sex differences that are not detected by linear-only methods. Gene pairs highly ranked by CCC were enriched for interactions in integrated networks, suggesting that CCC can detect functional relationships that linear-only methods miss. CCC is a powerful, next-generation not-only-linear correlation coefficient that can be applied to genome-scale data and other domains across different data types.

Introduction

Data collection technology has advanced significantly, creating immense amounts of data across various disciplines. This data provides new opportunities to answer scientific questions, as long as we have effective tools to detect different types of patterns. Correlation analysis is an important statistical technique for identifying relationships between variables [pmid:21310971?]. Correlation coefficients are used in exploratory data mining techniques, such as clustering or community detection algorithms, to compute the similarity between two objects, such as genes [pmid:27479844?] or lifestyle factors [1] related to diseases. Correlation methods are also used in supervised tasks, for example, to select features and improve prediction accuracy [pmid:27006077?,pmid:33729976?]. The Pearson correlation coefficient is widely used in various application domains and scientific areas. Therefore, even small and significant improvements in these techniques could have a great impact on industry and research.

Recent advances in gene expression analysis have revealed the importance of gene-gene relationships in understanding complex traits and human diseases. To uncover these relationships, correlation coefficients are often used to measure associations between genes. However, traditional correlation coefficients are not able to capture nonlinear relationships, which are believed to be an important part of highly-interconnected, disease-relevant regulatory networks [pmid:21241896?,pmid:25915600?,pmid:21606319?,pmid:16968540?]. Analysis of large RNA-seq datasets [pmid:32913098?,pmid:34844637?] has also revealed molecular mechanisms underlying human diseases [pmid:27479844?,pmid:31121115?,pmid:30668570?,pmid:32424349?,pmid:34475573?], and with the introduction of the omnigenic model of complex traits [pmid:28622505?,pmid:31051098?], gene-gene relationships are playing an increasingly important role in genetic studies of human diseases [2,3,4,pmid:34845454?], including the use of polygenic risk scores [5]. Combining disease-associated genes from genome-wide association studies (GWAS) with gene co-expression networks can further prioritize “core” genes directly affecting diseases [2,3,6], which cannot be detected by traditional statistical methods. Therefore, an efficient correlation coefficient that can capture nonlinear relationships could be beneficial for many areas of biology, such as the identification of candidate drug targets in the precision medicine field.

Widely used metrics such as Pearson and Spearman correlation coefficients are efficient for capturing linear or monotonic patterns, but may fail to detect complex yet critical nonlinear relationships. To address this, several novel coefficients have been proposed, such as Maximal Information Coefficient (MIC) [pmid:22174245?] and Distance Correlation (DC) [7], which have been successfully applied across multiple domains [pmid:33972855?,pmid:33001806?,pmid:27006077?]. However, their computational complexity makes them impractical for moderately sized datasets

[pmid:33972855?,pmid:27333001?]. We previously developed a clustering method that outperformed Pearson, Spearman, DC and MIC in detecting clusters of simulated linear and nonlinear relationships with varying noise levels [8]. Here, we introduce the Clustermatch Correlation Coefficient (CCC), an efficient not-only-linear coefficient that works for both quantitative and qualitative variables. CCC has a single parameter that limits the maximum complexity of relationships found (from linear to more general patterns) and computation time. We provide an efficient CCC implementation that is highly parallelizable, allowing for faster computation across variable pairs with millions of objects or conditions. To assess its performance, we applied our method to gene expression data from the Genotype-Tissue Expression v8 (GTEx) project across different tissues [9]. CCC successfully captured both strong linear relationships and novel nonlinear patterns, which were entirely missed by standard coefficients. Additionally, gene pairs detected in expression data by CCC had higher interaction probabilities in tissue-specific gene networks from the Genome-wide Analysis of gene Networks in Tissues (GIANT) [10]. The ability of CCC to efficiently handle diverse data types (including numerical and categorical features) reduces preprocessing steps and makes it appealing for analyzing large and heterogeneous repositories.

Results

A robust and efficient not-only-linear dependence coefficient

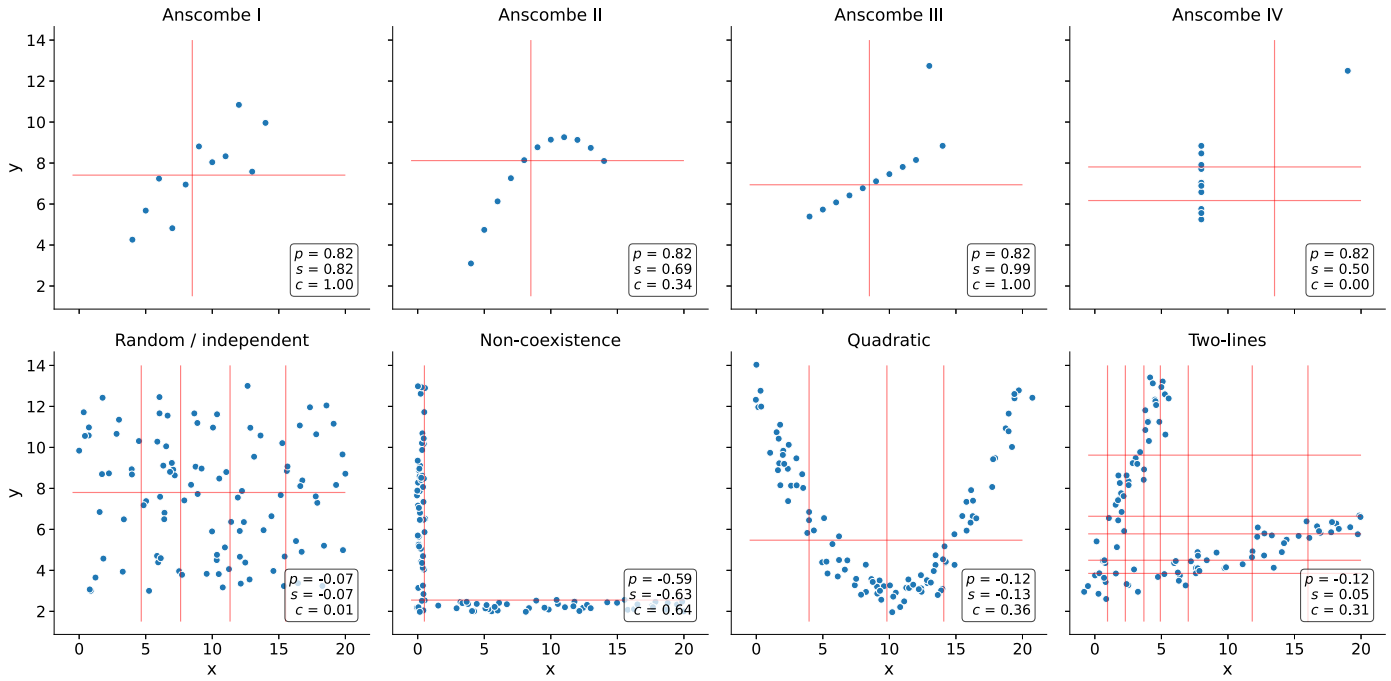

Figure 1: Different types of relationships in data. Each panel contains a set of simulated data points described by two generic variables: x and y. The first row shows Anscombe's quartet with four different datasets (from Anscombe I to IV) and 11 data points each. The second row contains a set of general patterns with 100 data points each. Each panel shows the correlation value using Pearson (p), Spearman (s) and CCC (c). Vertical and horizontal red lines show how CCC clustered data points using x and y.

Figure 1 shows the performance of CCC in comparison with other correlation coefficients for a nonlinear gene expression dataset.

The CCC provides a measure of similarity between any pair of variables, with either numerical or categorical values. Our method assumes that if two variables/features have a relationship, then the clustering of the data points/objects using each variable should be similar. For numerical values, CCC uses quantiles to separate them into clusters (e.g., the median divides numerical data into two clusters). We define the CCC as the maximum adjusted Rand index (ARI) [11] between the clusterings,

ranging from 0 to 1. Further details of the CCC algorithm can be found in Methods. Figure 1 compares the performance of CCC with other correlation coefficients for a nonlinear gene expression dataset.

We examined the behavior of Pearson (p), Spearman (s) and CCC (c) correlation coefficients on different simulated data patterns. Figure 1 shows the classic Anscombe's quartet [12], which comprises four synthetic datasets with different patterns but the same data statistics (mean, standard deviation and Pearson's correlation). The "Datasaurus" [13,14,15] is a reminder of the importance of going beyond simple statistics. This is because patterns such as outliers and nonlinear relationships, which are biologically meaningful, can be masked by summary statistics alone.

Figure ??? shows the Anscombe's examples and their CCC values.

The Anscombe's examples (Figure ???) demonstrate the effectiveness of CCC in detecting nonlinear relationships and outliers. Anscombe I and Anscombe III contain noisy and perfect linear patterns, respectively. CCC separates the data points in two clusters (one red line for each variable x and y), yielding 1.0 in both cases and thus indicating a strong relationship. Anscombe II follows a partially quadratic relationship interpreted as linear by Pearson and Spearman. CCC yields a lower yet non-zero value of 0.34, reflecting a more complex relationship than a linear pattern. Anscombe IV shows a vertical line of data points where x values are almost constant except for one outlier. This outlier does not influence CCC, which yields a value of 0.00, correctly indicating no association for this variable pair. Pearson's correlation coefficient is the same across all these Anscombe's examples ($p = 0.82$), whereas Spearman is 0.50 or greater. These simulated datasets show that both Pearson and Spearman are powerful in detecting linear patterns, but any deviation from this assumption affects their robustness.

We simulated different types of relationships, including some previously described from gene expression data [16,17,18] (see Figure 1, second row). For the random/independent pair of variables, all correlation coefficients correctly agreed with a value close to zero. The non-coexistence pattern was captured by all coefficients, which indicates that one gene (x) might be expressed while the other one (y) is inhibited, highlighting a potentially strong biological relationship (such as a microRNA negatively regulating another gene). Pearson and Spearman coefficients, however, failed to capture the nonlinear pattern between variables x and y in the quadratic and two-lines patterns. CCC, on the other hand, used different degrees of complexity to capture these relationships. For the quadratic pattern, it separated x into four clusters to reach the maximum ARI. In the two-lines example, it increased the complexity of the model by using eight clusters for x and six for y, resulting in a c coefficient of 0.31.

The CCC reveals linear and nonlinear patterns in human transcriptomic data

Figure 1 shows the distributions of the correlation coefficients for Pearson, Spearman and CCC.

We then examined the correlation coefficients of gene expression data from GTEx v8 across different tissues. We selected the top 5,000 genes with the largest variance and computed the correlation matrix between them using Pearson, Spearman and CCC (see Methods). Figure 1 shows the distributions of the correlation coefficients for the three methods.

We examined the absolute values of each coefficient's distribution in GTEx (Figure 2). CCC (mean=0.14, median=0.08, sd=0.15) had a much more skewed distribution than Pearson (mean=0.31, median=0.24, sd=0.24) and Spearman (mean=0.39, median=0.37, sd=0.26). 70% of gene pairs reached cumulative sets at different values (Figure 2 b), $c = 0.18$, $p = 0.44$ and $s = 0.56$, suggesting that for this type of data, the coefficients cannot be directly compared by magnitude, so we used ranks for further

comparisons. In GTEx v8, CCC values were closer to Spearman and vice versa than either was to Pearson (Figure 2 c). We also compared the Maximal Information Coefficient (MIC) in this data (see Supplementary Note 1). We found that CCC behaved similarly to MIC, although CCC was up to two orders of magnitude faster to run (see Supplementary Note 2). MIC is an advanced correlation coefficient which can capture general patterns beyond linear relationships and has been used in various application domains [pmid:33972855?,pmid:33001806?,pmid:27006077?]. These results suggest that our findings for CCC generalize to MIC. Therefore, in the subsequent analyses, we focus on CCC and linear-only coefficients.

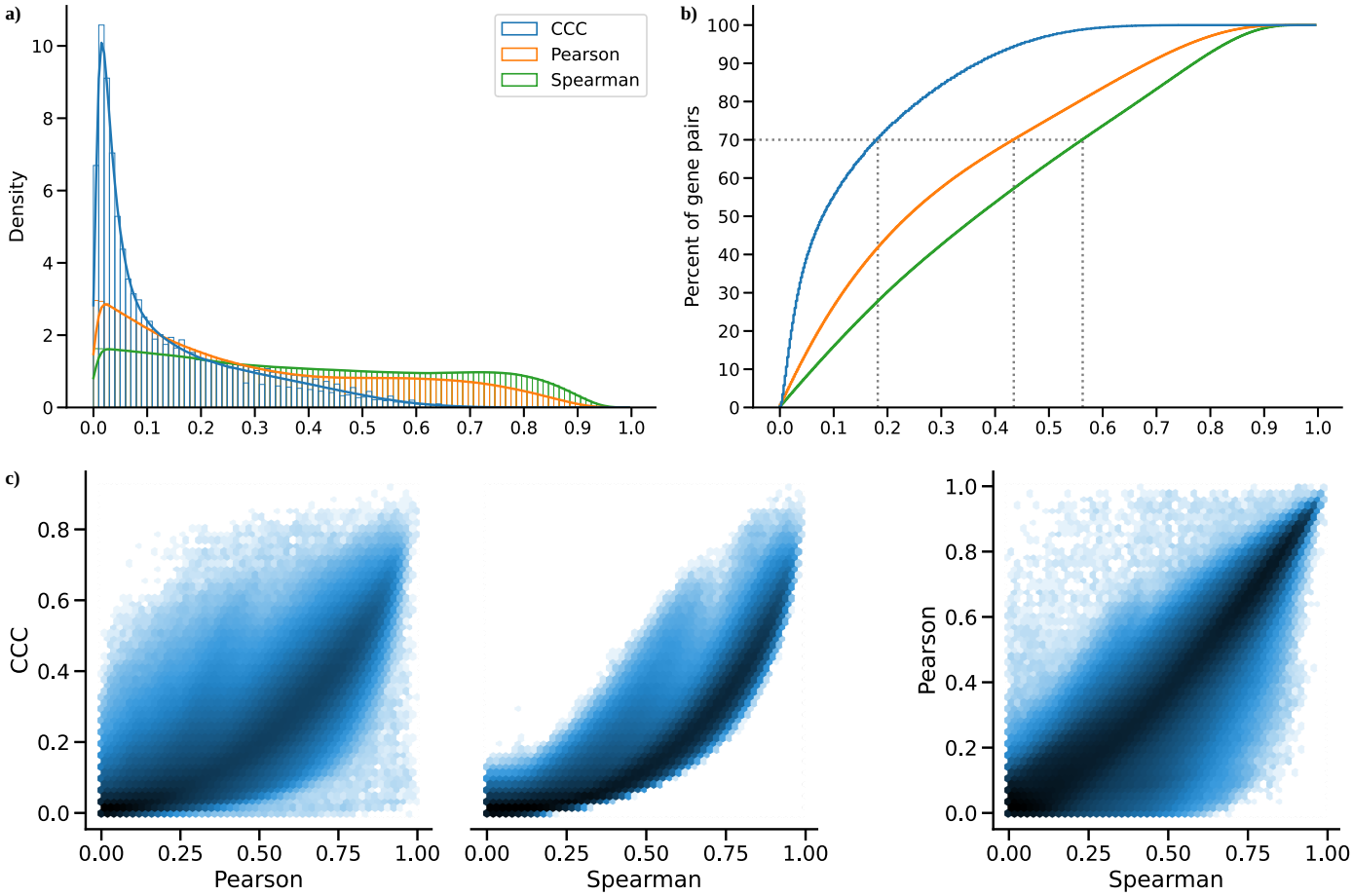

Figure 2: Distribution of coefficient values on gene expression (GTEx v8, whole blood). a) Histogram of coefficient values. **b)** Corresponding cumulative histogram. The dotted line maps the coefficient value that accumulates 70% of gene pairs. **c)** 2D histogram plot with hexagonal bins between all coefficients, where a logarithmic scale was used to color each hexagon.

We analyzed the agreements and disagreements between the three correlation coefficients by obtaining the top 30% of gene pairs with the largest correlation values ("high" set) and the bottom 30% ("low" set). An UpSet analysis [19] (Figure 3 a) showed that the three coefficients agreed on whether there is a strong correlation (42.1%) or no relationship (34.3%) in 76.4% of cases. We found that CCC and Spearman agreed more on either highly or poorly correlated pairs (4.0% in "high", and 7.0% in "low") than either with Pearson (all between 0.3%-3.5% for "high", and 2.8%-5.5% for "low"). In summary, CCC agreed with either Pearson or Spearman in 90.5% of gene pairs by assigning a high or a low correlation value.

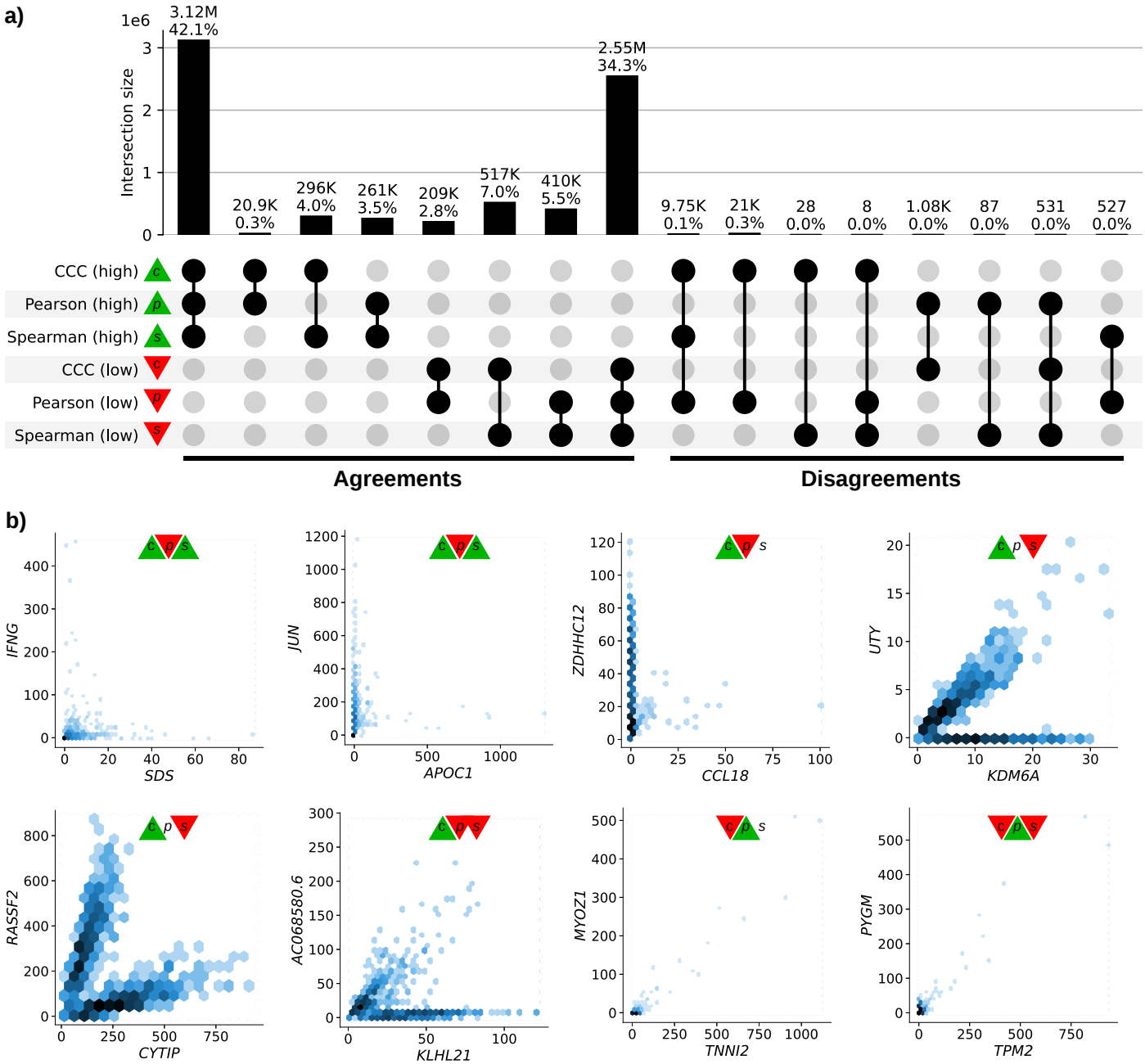

Figure 3: Intersection of gene pairs with high and low correlation coefficient values (GTEx v8, whole blood). a) UpSet plot with six categories (rows) grouping the 30% of the highest (green triangle) and lowest (red triangle) values for each coefficient. Columns show different intersections of categories grouped by agreements and disagreements. **b)** Hexagonal binning plots with examples of gene pairs where CCC (c) disagrees with Pearson (p) and Spearman (s). For each method, colors in the triangles indicate if the gene pair is among the top (green) or bottom (red) 30% of coefficient values. No triangle means that the correlation value for the gene pair is between the 30th and 70th percentiles (neither low nor high). A logarithmic scale was used to color each hexagon.

We found that for more than 20,000 gene pairs, the CCC value was higher than the values of other correlation coefficients (Figure 3 a, right). There were also 1,075 gene pairs with a high Pearson value and either low CCC or low Spearman values, and 531 gene pairs with both low CCC and low Spearman values. However, further analysis revealed that many of these cases were likely due to potential outliers (Figure 3 b). We examined the top five gene pairs of each intersection in the “Disagreements” group (Figure 3 a, right) where CCC disagreed with Pearson, Spearman or both.

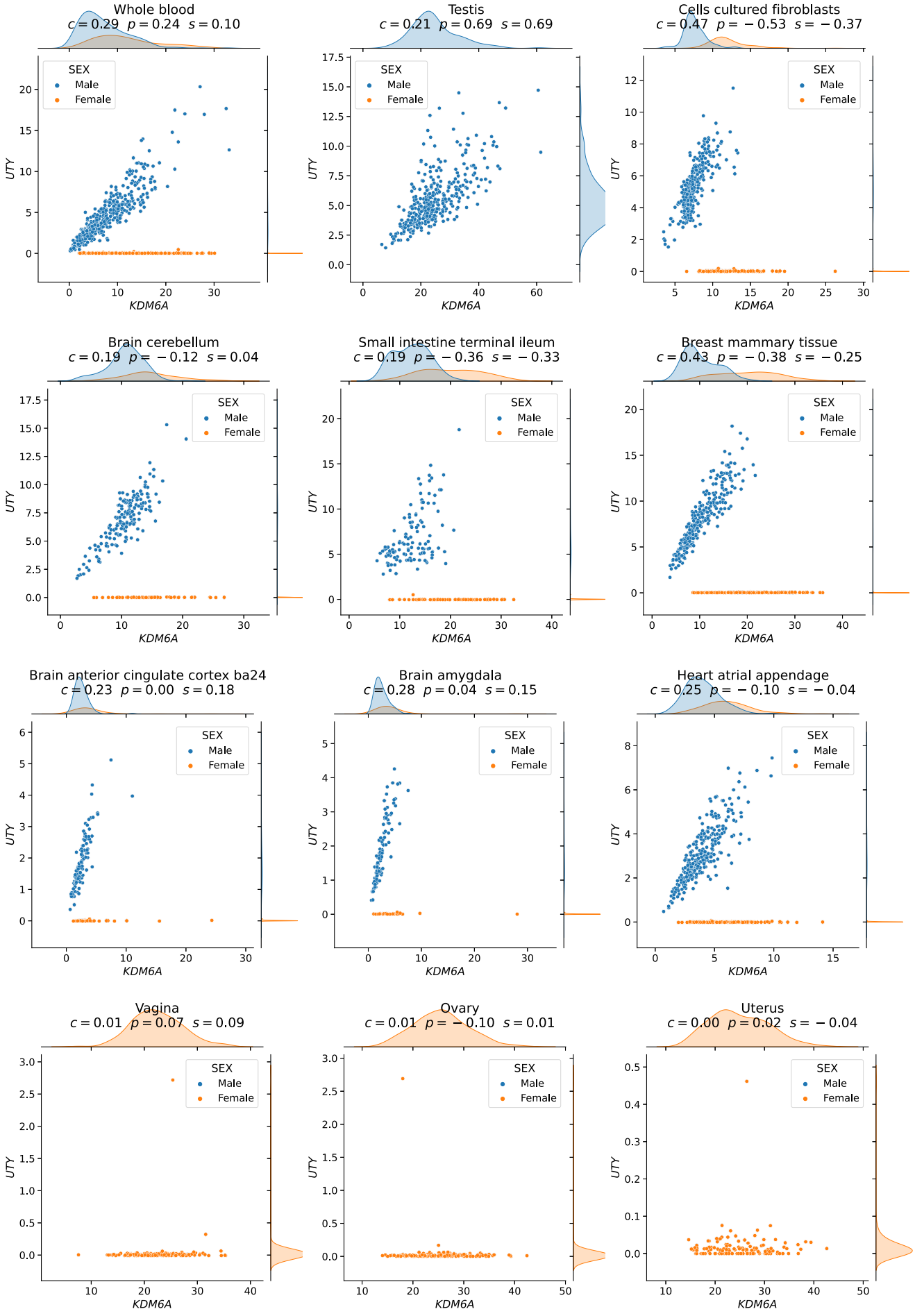

Figure 4: The expression levels of *KDM6A* and *UTY* display sex-specific associations across GTEx tissues. CCC captures this nonlinear relationship in all GTEx tissues (nine examples are shown in the first three rows), except in female-specific organs (last row).

The top three gene pairs (*IFNG* - *SDS*, *JUN* - *APOC1*, and *ZDHHC12* - *CCL18*) have high CCC and low Pearson values, suggesting a non-coexistence relationship between them. In samples where one of the genes is highly expressed, the other is slightly activated, and vice versa, which may indicate an inhibiting effect. The next three gene pairs (*UTY* - *KDM6A*, *RASSF2* - *CYTIP*, and *AC068580.6* - *KLHL21*) show patterns combining either two linear or one linear and one independent relationships. For example, *UTY* and *KDM6A* (paralogs) have a nonlinear relationship, where a subset of samples follows a linear pattern and another subset has a constant expression of one gene. This is because *UTY* is located on the Y chromosome (Yq11) and *KDM6A* on the X chromosome (Xp11). Males have a linear pattern, while females show no expression for *UTY*. This combination of linear and independent patterns is captured by CCC ($c = 0.29$, above the 80th percentile) but not by Pearson ($p = 0.24$, below the 55th percentile) or Spearman ($s = 0.10$, below the 15th percentile). Moreover, the same gene pair pattern is highly ranked by CCC in all other tissues in GTEx, except for female-specific organs (Figure 4).

Replication of gene associations using tissue-specific gene networks from GIANT

We sought to systematically analyze discrepant scores to assess whether associations were replicated in other datasets besides GTEx. This is challenging and prone to bias because linear-only correlation coefficients are usually used in gene co-expression analyses. We used 144 tissue-specific gene networks from the Genome-wide Analysis of gene Networks in Tissues (GIANT) [20,21], where nodes represent genes and each edge a functional relationship weighted with a probability of interaction between two genes (see Methods). Importantly, the version of GIANT used in this study did not include GTEx samples [22], making it an ideal case for replication. These networks were built from expression and different interaction measurements, including protein-interaction, transcription factor regulation, chemical/genetic perturbations and microRNA target profiles from the Molecular Signatures Database (MSigDB [**pmid:16199517?**]). We reasoned that highly-ranked gene pairs using three different coefficients in a single tissue (whole blood in GTEx, Figure 3) that represented real patterns should often replicate in a corresponding tissue or related cell lineage using the multi-cell type functional interaction networks in GIANT. In addition to predicting a network with interactions for a pair of genes, the GIANT web application can also automatically detect a relevant tissue or cell type where genes are predicted to be specifically expressed (the approach uses a machine learning method introduced in [23] and described in Methods). For example, we obtained the networks in blood and the automatically-predicted cell type for gene pairs *RASSF2* - *CYTIP* (CCC high, Figure 5 a) and *MYOZ1* - *TNNI2* (Pearson high, Figure 5 b). In addition to the gene pair, the networks include other genes connected according to their probability of interaction (up to 15 additional genes are shown), which allows estimating whether genes are part of the same tissue-specific biological process. Two large black nodes in each network's top-left and bottom-right corners represent our gene pairs. A green edge means a close-to-zero probability of interaction, whereas a red edge represents a strong predicted relationship between the two genes. In this example, genes *RASSF2* and *CYTIP* (Figure 5 a), with a high CCC value ($c = 0.20$, above the 73th percentile) and low Pearson and Spearman ($p = 0.16$ and $s = 0.11$, below the 38th and 17th percentiles, respectively), were both strongly connected to the blood network, with interaction scores of at least 0.63 and an average of 0.75 and 0.84, respectively (Supplementary Table ??). The autodetected cell type for this pair was leukocytes, and interaction scores were similar to the blood network (Supplementary Table ??). However, genes *MYOZ1* and *TNNI2*, with a very high Pearson value ($p = 0.97$), moderate Spearman ($s = 0.28$) and very low CCC ($c = 0.03$), were predicted to belong to much less cohesive networks (Figure 5 b), with average interaction scores of 0.17 and 0.22 with the rest of the genes, respectively. Additionally, the autodetected cell type (skeletal muscle) is not related to blood or one of its cell lineages. These preliminary results suggested that CCC might be capturing blood-specific patterns missed by the other coefficients.

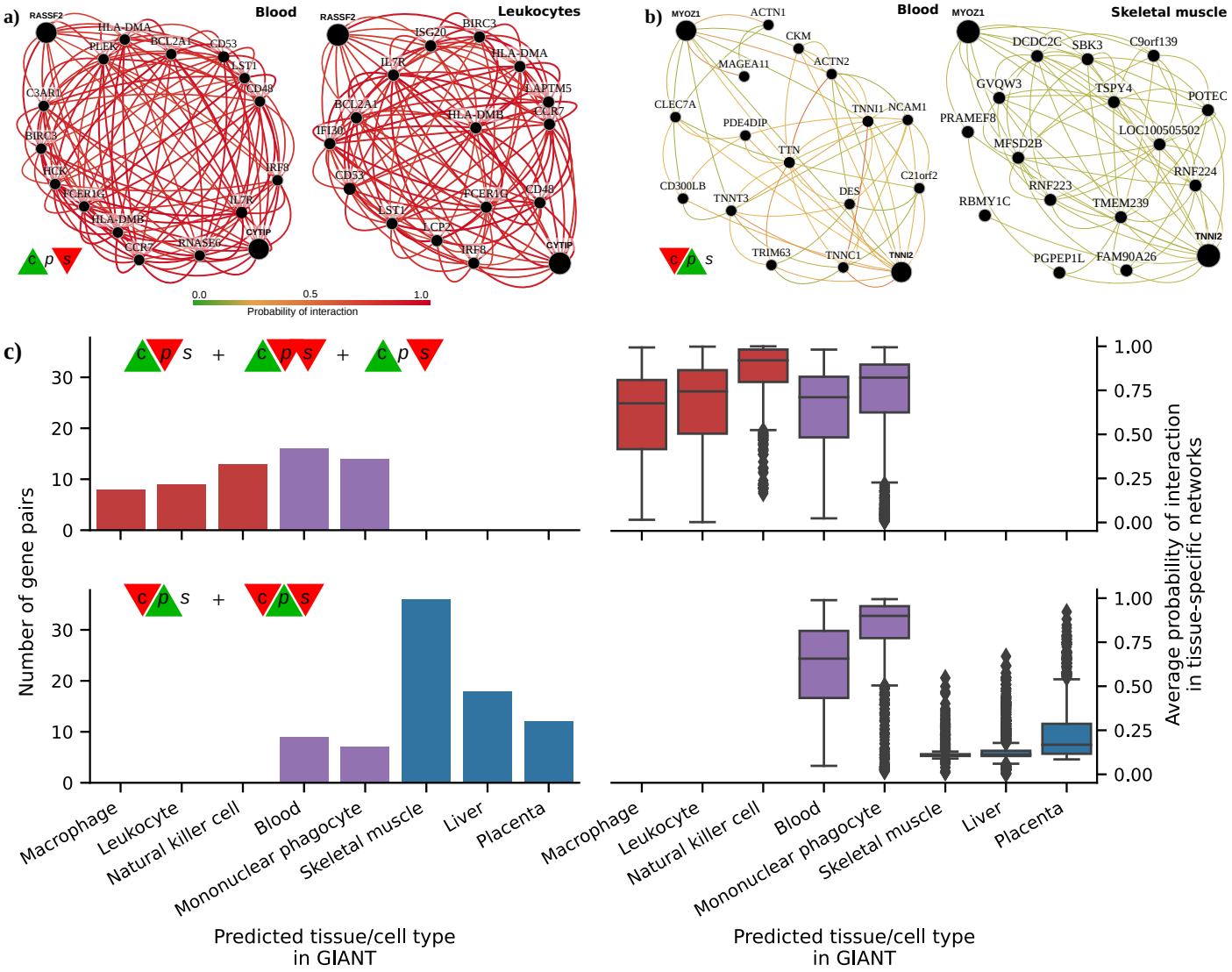

Figure 5: Analysis of GIANT tissue-specific predicted networks for gene pairs prioritized by correlation coefficients. a-b) Two gene pairs prioritized by correlation coefficients (from Figure 3 b) with their predicted networks in blood (left) and an automatically selected tissue/cell type (right) using the method described in [23]. A node represents a gene and an edge the probability that two genes are part of the same biological process in a specific cell type. A maximum of 15 genes are shown for each network. The GIANT web application automatically determined a minimum interaction confidence (edges' weights) to be shown. These networks can be analyzed online using the following links: *RASSF2 - CYTIP* [24], *MYOZ1 - TNNI2* [25]. **c)** Summary of predicted tissue/cell type networks for gene pairs exclusively prioritized by CCC and Pearson. The first row combines all gene pairs where CCC is high and Pearson or Spearman are low. The second row combines all gene pairs where Pearson is high and CCC or Spearman are low. Bar plots (left) show the number of gene pairs for each predicted tissue/cell type. Box plots (right) show the average probability of interaction between genes in these predicted tissue-specific networks. Red indicates CCC-only tissues/cell types, blue are Pearson-only, and purple are shared.

We next evaluated the top 100 gene pairs with the highest discrepancy between CCC and the other two coefficients. To assess whether genes were predicted to be specifically expressed in a blood-relevant cell lineage, we used the GIANT software to autodetect the relevant cell type for each gene pair (see Methods). The top five most commonly autodetected cell types for CCC-ranked gene pairs were all blood-specific (Figure 5 c, top left), including macrophage, leukocyte, natural killer cell, blood, and mononuclear phagocyte. The average probability of interaction between genes in these CCC-ranked networks was significantly higher than the other coefficients (Figure 5 c, top right), with all medians larger than 67% and first quartiles above 41% across predicted cell types. In contrast, most Pearson-ranked gene pairs were predicted to be specific to tissues unrelated to blood (Figure 5 c, bottom left), with skeletal muscle being the most commonly predicted tissue. The interaction probabilities in these Pearson-ranked networks were also generally lower than in CCC, except for blood-specific gene pairs (Figure 5 c, bottom right). These results suggest that the associations

exclusively detected by CCC in whole blood from GTEx were more strongly replicated in these independent networks that incorporated multiple data modalities. CCC-ranked gene pairs not only had high probabilities of belonging to the same biological process, but were also predicted to be specifically expressed in blood cell lineages. Conversely, most Pearson-ranked gene pairs were not predicted to be blood-specific, and their interaction probabilities were relatively low. This lack of replication in GIANT is consistent with our earlier observations of outlier-driven associations (Figure 3 b).

Discussion

We introduce the Clustermatch Correlation Coefficient (CCC), an efficient machine learning-based statistic that can capture nonlinear relationships. Applying CCC to the GTEx v8 dataset revealed that it was robust to outliers and detected both linear and complex patterns that standard coefficients missed. For example, CCC alone detected gene pairs with nonlinear patterns on sex chromosomes, demonstrating its capacity to capture sex-specific differences. This ability extends beyond sex differences, as CCC can detect complex relationships where a subset of samples or conditions are explained by other factors, such as health and disease. We found that the top CCC-ranked gene pairs in whole blood from GTEx were replicated in independent tissue-specific networks trained from multiple data types and attributed to cell lineages from blood, even though CCC had no access to any cell lineage-specific information. This suggests that CCC can accurately identify intricate cell lineage-specific transcriptional patterns that linear-only coefficients cannot. Furthermore, CCC was more similar to Spearman than Pearson in terms of robustness to outliers. Moreover, CCC results were concordant with MIC [**pmid:27683765?**], but much faster to compute, making it practical for large datasets. In addition, CCC can also process categorical variables together with numerical values. It is conceptually easy to interpret and has a single parameter that controls the maximum complexity of the detected relationships while also balancing compute time.

Visualization of datasets such as Anscombe or “Datasaurus” can be helpful in understanding relationships between variables. However, when examining many datasets, it is often infeasible to examine each possible relationship. In such cases, more sophisticated and robust correlation coefficients are necessary. The CCC coefficient is an example of an advanced yet interpretable coefficient that can help focus human interpretation on patterns that are more likely to reflect real biology. For example, CCC detected a strong linear relationship between genes *UTY* and *KDM6A* (from sex chromosomes), but only in a subset of samples (males). This highlights the importance of considering sex as a biological variable (SABV) [26] to avoid overlooking differences between men and women, for instance, in disease manifestations [27,28]. Furthermore, a not-only-linear correlation coefficient like CCC could identify significant differences between variables that are explained by a third factor, which would be missed by linear-only coefficients.

It is well-known that biomedical research focuses on a small fraction of human genes [**pmid:17620606?**,**pmid:17472739?**]. This is seen in the CCC-ranked pairs (Figure 3 b), which include genes such as *SDS* (12q24) and *ZDHHC12* (9q34). These genes have been found to receive fewer publications than expected [**pmid:30226837?**]. This could be because linear correlation coefficients are widely used, which could bias researchers away from genes with complex coexpression patterns. To further investigate this, a beyond-linear gene co-expression analysis of large datasets could be conducted. This could shed light on the function of understudied genes, such as *KLHL21* (1p36) and *AC068580.6* (*ENSG00000235027*, in 11p15). These genes have a high CCC value and are missed by other coefficients. For example, *KLHL21* has been suggested as a potential therapeutic target for hepatocellular carcinoma [**pmid:27769251?**] and other cancers [**pmid:29574153?**,**pmid:35084622?**]. Its nonlinear correlation with *AC068580.6* may reveal other important players in cancer initiation or progression, possibly in subsets of samples with specific characteristics (as suggested in Figure 3 b).

Not-only-linear correlation coefficients like CCC could be beneficial in genetic studies, such as genome-wide association studies (GWAS). GWAS have been successful in understanding the molecular basis of common diseases, but the estimated effect sizes of the genes identified are generally modest, and they explain only a fraction of the phenotype variance [29,30]. The omnigenic model for complex traits [**pmid:28622505?**,**pmid:31051098?**] suggests that highly-interconnected gene regulatory networks, with some core genes having a more direct effect on the phenotype than others, could explain this. We and others [2,3,6] have demonstrated that integrating gene co-expression networks in genetic studies could uncover core genes that are missed by linear-only models like GWAS. Our results suggest that more efficient correlation coefficients could better estimate gene co-expression profiles and therefore more accurately identify these core genes. Such approaches could be useful for precision medicine by providing the computational tools to focus on more promising genes, which could be better candidate drug targets.

Our study has some limitations. We used a sample of the top variable genes to keep computation time feasible. Although our correlation coefficient (CCC) is much faster than other methods such as Maximum Information Coefficient (MIC), Pearson, and Spearman, these are still the most computationally efficient since they rely on simple data statistics. However, our results reveal the advantages of using more advanced coefficients like CCC for detecting and studying more intricate molecular mechanisms that replicate in independent datasets. We suggest that applying CCC to larger compendia, such as recount3 [**pmid:34844637?**] with thousands of heterogeneous samples across different conditions, can reveal other potentially meaningful gene interactions. The single parameter of CCC, $k_{\max}$, controls the maximum complexity of patterns found and also affects computation time. Our analysis found that $k_{\max} = 10$ was sufficient to identify both linear and more complex patterns in gene expression. A more comprehensive analysis of optimal values for this parameter could provide insights to adjust it for different applications or data types.

[31]

Linear and rank-based correlation coefficients are fast to calculate, but they cannot capture nonlinear patterns in biological datasets. For example, patterns associated with sex are not revealed by linear-only coefficients, but can be identified with not-only-linear methods. Furthermore, not-only-linear coefficients can disentangle intricate patterns from expression data alone, which can be replicated in models integrating different data modalities. The CCC correlation coefficient, in particular, is highly parallelizable, and could potentially be implemented on a GPU to make it even faster. It is an efficient, next-generation correlation coefficient that is highly effective in transcriptome analyses and has potential uses in a broad range of other domains. [31]

Methods

The proposed correlation coefficient (CCC) was implemented in Python and uses scikit-learn [**pmid:25431818?**] to fit the nonlinear models.

The code to reproduce our analyses and generate the figures is available at https://github.com/greenelab/ccc. We provide scripts to download the required data and run all the steps. A Docker image is also available to use the same runtime environment. The proposed correlation coefficient (CCC) was implemented in Python and uses scikit-learn [**pmid:25431818?**] to fit the nonlinear models.

The CCC algorithm

The Clustermatch Correlation Coefficient (CCC) calculates a similarity value $c \in [0, 1]$ between any two numerical or categorical features/variables $\mathbf{x}$ and $\mathbf{y}$ measured on n objects. CCC assumes that when two features $\mathbf{x}$ and $\mathbf{y}$ are similar, the clustering of the n objects using each feature would match. For example, given $\mathbf{x} = (11, 27, 32, 40)$ and $\mathbf{y} = 10x = (110, 270, 320, 400)$, with $n = 4$, clustering each variable into two clusters ($k = 2$) using their medians (29.5 for $\mathbf{x}$ and 295 for $\mathbf{y}$) would result in partitions $\Omega^{\mathbf{x}}_{k=2} = (1, 1, 2, 2)$ for $\mathbf{x}$, and $\Omega^{\mathbf{y}}_{k=2} = (1, 1, 2, 2)$ for $\mathbf{y}$. The agreement between $\Omega^{\mathbf{x}}_{k=2}$ and $\Omega^{\mathbf{y}}_{k=2}$ can be computed using any measure of similarity between partitions, such as the adjusted Rand index (ARI) [11], which returns the maximum value (1.0 in the case of ARI). Note that the same value of k may not always be the right one to find a relationship between any two features. For instance, in the quadratic example in Figure 1, CCC returns a value of 0.36 (grouping objects in four clusters using one feature and two using the other). If we used only two clusters instead, CCC would return a similarity value of 0.02. Therefore, the CCC algorithm (shown below) searches for the optimal number of clusters given a maximum k, which is its single parameter $k_{\max}$.

Algorithm 1: CCC algorithm

```
1  Function get_partitions(v, k_max):
       Output:
           Ω_r: clustering with r clusters over n objects
2      if v ∈ ℝ^n then
3          for r ← 2 to min{k_max, |v| − 1} do
4              ρ ← (ρ_ℓ | Pr(v_i < ρ_ℓ) ≤ (ℓ − 1)/r), ∀ℓ ∈ [1, r + 1]
5              Ω_rℓ ← {i | ρ_ℓ < v_i ≤ ρ_{ℓ+1}}, ∀ℓ ∈ [1, r]
6      else
7          C ← ∪_j {v_i}
8          r ← |C|
9          Ω_rc ← {i | v_i = C_c}, ∀c ∈ [1, r]
10     Ω ← {Ω_r | |Ω_r| > 1}, ∀r
11     return Ω
12
13 Function ccc(x, y, k_max):
       Input:
           x: feature values on n objects
           y: feature values on n objects
           k_max: maximum number of internal clusters
       Output:
           c: similarity value for x and y (c ∈ [0, 1])
14     Ω^x = get_partitions(x, k_max)
15     Ω^y = get_partitions(y, k_max)
16     c ← max{A(Ω^x_p, Ω^y_q)}, ∀p, q
17     return max(c, 0)
```

The algorithm `ccc` starts by partitioning each of the features $\mathbf{x}$ and $\mathbf{y}$ into $\Omega^{\mathbf{x}}$ and $\Omega^{\mathbf{y}}$ (lines 14 and 15). Then, it computes the Adjusted Rand Index (ARI) between each pair of partitions in $\Omega^{\mathbf{x}}$ and $\Omega^{\mathbf{y}}$ (line 16) and keeps the pair that generates the maximum ARI. Since ARI does not have a lower bound (it could return negative values, which are not meaningful in our case), the algorithm returns only values between 0 and 1 (line 17) [doi:10.1145/3104482.3104525?].

This allows CCC to detect nonlinear relationships [pmid:15861768?], which is a major limitation of linear correlation coefficients [32].

CCC only requires two partitions to calculate a similarity value, so it supports any feature that can be used for clustering. If the feature is numerical (lines 2 to 5 in the `get_partitions` function), it uses quantiles to group objects into clusters (e.g. the median produces $k = 2$ clusters). This can be done from $k = 2$ to $k = k_{\max}$. Categorical features (lines 7 to 9) are grouped into categories. This means numerical and categorical variables can be combined since the clusters do not need to be ordered. This allows CCC to detect nonlinear relationships, which is a major advantage over linear correlation coefficients [32].

We used $k_{\max} = 10$ in our analyses. This means that for each gene pair, 18 partitions were generated (9 for each gene, from $k = 2$ to $k = 10$) and 81 ARI comparisons were performed. Smaller values of $k_{\max}$ can reduce computation time, but may overlook more complex/general relationships. Our examples in Figure 1 suggest that using $k_{\max} = 2$ would force CCC to find only linear patterns, which could be a valid use case scenario where only this kind of relationships are desired. In addition, $k_{\max} = 2$ implies that only two partitions are generated, and only one ARI comparison is performed. Our Python implementation of CCC provides flexibility in specifying $k_{\max}$. For instance, instead of the maximum k (an integer), the parameter could be a custom list of integers, such as `[2, 5, 10]` which would partition the data into two, five and ten clusters.

We used three CPU cores to speed up the computation of our correlation coefficient (CCC). This allowed us to parallelize the process of generating partitions and computing similarity for each pair of features (genes in our study). To further increase the speed of computation, an improved implementation of CCC could make use of graphical processing units (GPUs) [pmid:338467?].

We have created a Python implementation of the CCC, which is optimized with the `numba` library [33]. This implementation is available in our Github repository [34], as well as a package published in the Python Package Index (PyPI), which can be easily installed.

Gene expression data and preprocessing

We used GTEx v8 data for all tissues and normalized it using transcripts per million (TPM). We focused our primary analysis on whole blood, which had 755 samples. To avoid a bias towards highly-expressed genes, we standardized the data with $log(x + 1)$ and selected the top 5,000 genes with the largest variance. We then computed Pearson, Spearman, maximal information coefficient (MIC) and an efficient not-only-linear correlation coefficient (CCC) on the 5,000 genes across all 755 samples on the TPM-normalized data, resulting in a pairwise similarity matrix of size 5,000 x 5,000 [pmid:30207583?].

Tissue-specific network analyses using GIANT

We used the GIANT database [21] to access tissue-specific gene networks. Our version of GIANT included 987 genome-scale datasets with around 38,000 conditions from 14,000 publications. The details on how these networks were built are described in [10]. We used gene expression data from the NCBI Gene Expression Omnibus (GEO) [35], protein-protein interactions from BioGRID [36], IntAct [37], MINT [38] and MIPS [39], transcription factor regulation from JASPAR [40], and chemical and genetic perturbations from MSigDB [41]. We log-transformed the gene expression data, computed the Pearson correlation for each gene pair, normalized it using the Fisher's z transform, and discretized the z-scores into different bins. Gold standards for tissue-specific functional relationships were built using expert curation and experimentally derived gene annotations from the Gene Ontology. We then trained a naive Bayesian classifier (using C++ implementations from the Sleipnir library

[pmid:18499696?]) for each of the 144 tissues, using these gold standards. Finally, these classifiers were used to estimate the probability of tissue-specific interactions for each gene pair.

The average probability of interaction between the genes was computed as $2 \times \frac{\sum_{i=1}^{n} P_i}{n(n-1)}$, where P_i is the probability of interaction between the genes and n is the number of genes in the network.

For each pair of genes prioritized in our study using GTEx, we used GIANT through HumanBase to obtain a predicted gene network for blood (manually selected to match whole blood in GTEx) and a gene network with an automatically predicted tissue. This tissue prediction approach is described in [23], and it uses a machine learning model trained with comprehensive transcriptional data and gold standards of different cell lineages (e.g., macrophages). This model is then used to predict other cell lineage-specific genes. We included the top 15 genes with the highest probability of interaction with the queried gene pair for each network. The average probability of interaction between the genes was computed as $2 \times \frac{\sum_{i=1}^{n} P_i}{n(n-1)}$, where P_i is the probability of interaction between the genes and n is the number of genes in the network.

Maximal Information Coefficient (MIC)

We used the Python package `minepy` [42,43] (version 1.2.5) to estimate the MIC coefficient in GTEx v8 (whole blood) with the default parameters `alpha=0.6`, `c=15` and `estimator='mic_e'` (an improved implementation of the original MIC introduced in [44]). For parallelization, we used the `pairwise_distances` function from `scikit-learn` [45]. Additionally, for our computational complexity analyses (see Supplementary Material), we ran the original MIC (using parameter `estimator='mic_approx'`) and MIC_e (`estimator='mic_e'`).

References

1. **Using distance correlation and SS-ANOVA to assess associations of familial relationships, lifestyle factors, diseases, and mortality**
Jing Kong, Barbara EK Klein, Ronald Klein, Kristine E Lee, Grace Wahba
Proceedings of the National Academy of Sciences (2012-11-21) https://doi.org/f4htm9
DOI: 10.1073/pnas.1217269109 · PMID: 23175793 · PMCID: PMC3528609

2. **Projecting genetic associations through gene expression patterns highlights disease etiology and drug mechanisms**
Milton Pividori, Sumei Lu, Binglan Li, Chun Su, Matthew E Johnson, Wei-Qi Wei, Qiping Feng, Bahram Namjou, Krzysztof Kiryluk, Iftikhar Kullo, … Casey S Greene
Cold Spring Harbor Laboratory (2021-07-06) https://doi.org/gk9g25
DOI: 10.1101/2021.07.05.450786

3. **Linking common and rare disease genetics through gene regulatory networks**
Olivier B Bakker, Annique Claringbould, Harm-Jan Westra, Henry Wiersma, Floranne Boulogne, Urmo Võsa, Sophie Mulcahy Symmons, Iris H Jonkers, Lude Franke, Patrick Deelen
Cold Spring Harbor Laboratory (2021-10-26) https://doi.org/gpdftn
DOI: 10.1101/2021.10.21.21265342

4. **Large-scale cis- and trans-eQTL analyses identify thousands of genetic loci and polygenic scores that regulate blood gene expression**
Urmo Võsa, Annique Claringbould, Harm-Jan Westra, Marc Jan Bonder, Patrick Deelen, Biao Zeng, Holger Kirsten, Ashis Saha, Roman Kreuzhuber, Seyhan Yazar, …
Nature Genetics (2021-09) https://doi.org/gmpj66
DOI: 10.1038/s41588-021-00913-z · PMID: 34475573 · PMCID: PMC8432599

5. **The omnigenic model and polygenic prediction of complex traits**
Iain Mathieson
The American Journal of Human Genetics (2021-09) https://doi.org/gmv9s5
DOI: 10.1016/j.ajhg.2021.07.003 · PMID: 34331855 · PMCID: PMC8456163

6. **Identification of therapeutic targets from genetic association studies using hierarchical component analysis**
Hao-Chih Lee, Osamu Ichikawa, Benjamin S Glicksberg, Aparna A Divaraniya, Christine E Becker, Pankaj Agarwal, Joel T Dudley
BioData Mining (2020-06-17) https://doi.org/gjp5pf
DOI: 10.1186/s13040-020-00216-9 · PMID: 32565911 · PMCID: PMC7301559

7. **Measuring and testing dependence by correlation of distances**
Gábor J Székely, Maria L Rizzo, Nail K Bakirov
The Annals of Statistics (2007-12-01) https://doi.org/dkgjb4
DOI: 10.1214/009053607000000505

8. **Clustermatch: discovering hidden relations in highly diverse kinds of qualitative and quantitative data without standardization**
Milton Pividori, Andres Cernadas, Luis A de Haro, Fernando Carrari, Georgina Stegmayer, Diego H Milone
Bioinformatics (2018-10-24) https://doi.org/gfg4bt
DOI: 10.1093/bioinformatics/bty899 · PMID: 30357313

9. **The GTEx Consortium atlas of genetic regulatory effects across human tissues**

, François Aguet, Shankara Anand, Kristin G Ardlie, Stacey Gabriel, Gad A Getz, Aaron Graubert, Kane Hadley, Robert E Handsaker, Katherine H Huang, … Simona Volpi
Science (2020-09-11) https://doi.org/ghbnhr
DOI: 10.1126/science.aaz1776 · PMID: 32913098 · PMCID: PMC7737656

10. **Understanding multicellular function and disease with human tissue-specific networks**
Casey S Greene, Arjun Krishnan, Aaron K Wong, Emanuela Ricciotti, Rene A Zelaya, Daniel S Himmelstein, Ran Zhang, Boris M Hartmann, Elena Zaslavsky, Stuart C Sealfon, … Olga G Troyanskaya
Nature Genetics (2015-04-27) https://doi.org/f7dvkv
DOI: 10.1038/ng.3259 · PMID: 25915600 · PMCID: PMC4828725

11. **Comparing partitions**
Lawrence Hubert, Phipps Arabie
Journal of Classification (1985-12) https://doi.org/bphmzh
DOI: 10.1007/bf01908075

12. **Graphs in Statistical Analysis**
FJ Anscombe
The American Statistician (1973-02) https://doi.org/gfpn48
DOI: 10.1080/00031305.1973.10478966

13. **Download the Datasaurus: Never trust summary statistics alone; always visualize your data**
Alberto Cairo
http://www.thefunctionalart.com/2016/08/download-datasaurus-never-trust-summary.html

14. **Same Stats, Different Graphs**
Justin Matejka, George Fitzmaurice
Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (2017-05-02) https://doi.org/gdtg2w
DOI: 10.1145/3025453.3025912

15. **Generating data sets for teaching the importance of regression analysis**
Lori L Murray, John G Wilson
Decision Sciences Journal of Innovative Education (2021-03-31) https://doi.org/gjmgqt
DOI: 10.1111/dsji.12233

16. **Detecting Novel Associations in Large Data Sets**
David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, Pardis C Sabeti
Science (2011-12-16) https://doi.org/bzn5c3
DOI: 10.1126/science.1205438 · PMID: 22174245 · PMCID: PMC3325791

17. **A Novel Method to Efficiently Highlight Nonlinearly Expressed Genes**
Qifei Wang, Haojian Zhang, Yuqing Liang, Heling Jiang, Siqiao Tan, Feng Luo, Zheming Yuan, Yuan Chen
Frontiers in Genetics (2020-01-31) https://doi.org/gnr5k7
DOI: 10.3389/fgene.2019.01410 · PMID: 32082366 · PMCID: PMC7006292

18. **Comprehensive Identification of Cell Cycle–regulated Genes of the YeastSaccharomyces cerevisiaeby Microarray Hybridization**
Paul T Spellman, Gavin Sherlock, Michael Q Zhang, Vishwanath R Iyer, Kirk Anders, Michael B Eisen, Patrick O Brown, David Botstein, Bruce Futcher
Molecular Biology of the Cell (1998-12) https://doi.org/gnr5k5

DOI: 10.1091/mbc.9.12.3273 · PMID: 9843569 · PMCID: PMC25624

19. **UpSet: Visualization of Intersecting Sets**
Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, Hanspeter Pfister
IEEE Transactions on Visualization and Computer Graphics (2014-12-31) https://doi.org/f3ssr5
DOI: 10.1109/tvcg.2014.2346248 · PMID: 26356912 · PMCID: PMC4720993

20. **Understanding multicellular function and disease with human tissue-specific networks**
Casey S Greene, Arjun Krishnan, Aaron K Wong, Emanuela Ricciotti, Rene A Zelaya, Daniel S Himmelstein, Ran Zhang, Boris M Hartmann, Elena Zaslavsky, Stuart C Sealfon, … Olga G Troyanskaya
Nature genetics (2015-06) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4828725/
DOI: 10.1038/ng.3259 · PMID: 25915600 · PMCID: PMC4828725

21. **HumanBase: data-driven predictions of gene function and interactions**
https://hb.flatironinstitute.org/

22. **Data sources** https://hb.flatironinstitute.org/data

23. **Defining cell-type specificity at the transcriptional level in human disease**
Wenjun Ju, Casey S Greene, Felix Eichinger, Viji Nair, Jeffrey B Hodgin, Markus Bitzer, Young-suk Lee, Qian Zhu, Masami Kehata, Min Li, … Matthias Kretzler
Genome Research (2013-08-15) https://doi.org/f5g4hm
DOI: 10.1101/gr.155697.113 · PMID: 23950145 · PMCID: PMC3814886

24. **RASSF2, CYTIP - HumanBase** https://hb.flatironinstitute.org/gene/9770+9595

25. **MYOZ1, TNNI2 - HumanBase** https://hb.flatironinstitute.org/gene/58529+7136

26. **Policy: NIH to balance sex in cell and animal studies**
Janine A Clayton, Francis S Collins
Nature (2014-05) https://doi.org/gfzc82
DOI: 10.1038/509282a · PMID: 24834516 · PMCID: PMC5101948

27. **Considering Sex as a Biological Variable in Basic and Clinical Studies: An Endocrine Society Scientific Statement**
Aditi Bhargava, Arthur P Arnold, Debra A Bangasser, Kate M Denton, Arpana Gupta, Lucinda M Hilliard Krause, Emeran A Mayer, Margaret McCarthy, Walter L Miller, Armin Raznahan, Ragini Verma
Endocrine Reviews (2021-03-11) https://doi.org/gm642r
DOI: 10.1210/endrev/bnaa034 · PMID: 33704446 · PMCID: PMC8348944

28. **Considering sex as a biological variable will require a global shift in science culture**
Rebecca M Shansky, Anne Z Murphy
Nature Neuroscience (2021-03-01) https://doi.org/gjhkx8
DOI: 10.1038/s41593-021-00806-8 · PMID: 33649507

29. **10 Years of GWAS Discovery: Biology, Function, and Translation**
Peter M Visscher, Naomi R Wray, Qian Zhang, Pamela Sklar, Mark I McCarthy, Matthew A Brown, Jian Yang
The American Journal of Human Genetics (2017-07) https://doi.org/gcsmnm
DOI: 10.1016/j.ajhg.2017.06.005 · PMID: 28686856 · PMCID: PMC5501872

30. **Benefits and limitations of genome-wide association studies**
Vivian Tam, Nikunj Patel, Michelle Turcotte, Yohan Bossé, Guillaume Paré, David Meyre
Nature Reviews Genetics (2019-05-08) https://doi.org/ggcxxb

DOI: 10.1038/s41576-019-0127-1 · PMID: 31068683

31. **Comment on: 'Empirical comparison of web-based antimicrobial peptide prediction tools'**
Boris Vishnepolsky, Malak Pirtskhalava
Bioinformatics (2018-12-18) https://doi.org/grjmbb
DOI: 10.1093/bioinformatics/bty1023 · PMID: 30561507

32. **IGG3: a tool to rapidly integrate large genotype datasets for whole-genome imputation and individual-level meta-analysis**
Miao-Xin Li, Lin Jiang, Patrick Yu-Ping Kao, Pak-C Sham, You-Qiang Song
Bioinformatics (2009-04-03) https://doi.org/chkd9k
DOI: 10.1093/bioinformatics/btp183 · PMID: 19346322

33. **Numba**
Siu Kwan Lam, Antoine Pitrou, Stanley Seibert
Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15 (2015) https://doi.org/gf3nks
DOI: 10.1145/2833157.2833162

34. **Clustermatch Correlation Coefficient (CCC)**
Greene Laboratory
(2022-12-08) https://github.com/greenelab/ccc

35. **NCBI GEO: archive for functional genomics data sets—update**
Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, … Alexandra Soboleva
Nucleic Acids Research (2012-11-26) https://doi.org/f3mn62
DOI: 10.1093/nar/gks1193 · PMID: 23193258 · PMCID: PMC3531084

36. **The BioGRID interaction database: 2013 update**
Andrew Chatr-Aryamontri, Bobby-Joe Breitkreutz, Sven Heinicke, Lorrie Boucher, Andrew Winter, Chris Stark, Julie Nixon, Lindsay Ramage, Nadine Kolas, Lara O'Donnell, … Mike Tyers
Nucleic acids research (2013-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3531226/
DOI: 10.1093/nar/gks1158 · PMID: 23203989 · PMCID: PMC3531226

37. **The IntAct molecular interaction database in 2012**
S Kerrien, B Aranda, L Breuza, A Bridge, F Broackes-Carter, C Chen, M Duesbury, M Dumousseau, M Feuermann, U Hinz, … H Hermjakob
Nucleic Acids Research (2011-11-24) https://doi.org/bpmrdk
DOI: 10.1093/nar/gkr1088 · PMID: 22121220 · PMCID: PMC3245075

38. **MINT, the molecular interaction database: 2012 update**
Luana Licata, Leonardo Briganti, Daniele Peluso, Livia Perfetto, Marta Iannuccelli, Eugenia Galeota, Francesca Sacco, Anita Palma, Aurelio Pio Nardozza, Elena Santonico, … Gianni Cesareni
Nucleic Acids Research (2011-11-16) https://doi.org/cqvx3b
DOI: 10.1093/nar/gkr930 · PMID: 22096227 · PMCID: PMC3244991

39. **MIPS: a database for genomes and protein sequences**
HW Mewes, K Heumann, A Kaps, K Mayer, F Pfeiffer, S Stocker, D Frishman
Nucleic acids research (1999-01-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC148093/
DOI: 10.1093/nar/27.1.44 · PMID: 9847138 · PMCID: PMC148093

40. **JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles**

Elodie Portales-Casamar, Supat Thongjuea, Andrew T Kwon, David Arenillas, Xiaobei Zhao, Eivind Valen, Dimas Yusuf, Boris Lenhard, Wyeth W Wasserman, Albin Sandelin
Nucleic Acids Research (2009-11-10) https://doi.org/ddwfqp
DOI: 10.1093/nar/gkp950 · PMID: 19906716 · PMCID: PMC2808906

41. **Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles**
Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, Jill P Mesirov
Proceedings of the National Academy of Sciences (2005-09-30) https://doi.org/d4qbh8
DOI: 10.1073/pnas.0506580102 · PMID: 16199517 · PMCID: PMC1239896

42. **minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers**
Davide Albanese, Michele Filosi, Roberto Visintainer, Samantha Riccadonna, Giuseppe Jurman, Cesare Furlanello
Bioinformatics (2012-12-14) https://doi.org/f4nxg6
DOI: 10.1093/bioinformatics/bts707 · PMID: 23242262

43. **minepy - Maximal Information-based Nonparametric Exploration**
minepy - Maximal Information-based Nonparametric Exploration (MINE) in C and Python (2022-11-26) https://github.com/minepy/minepy

44. **Measuring Dependence Powerfully and Equitably**
Yakir Reshef, David Reshef, Hilary Finucane, Pardis Sabeti, Michael Mitzenmacher
Journal of Machine Learning Research (2010) https://jmlr.org/papers/v17/15-308.html

45. **Scikit-learn: Machine Learning in Python**
Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, … Edouard Duchesnay
Journal of Machine Learning Research (2011)
https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

46. **An improved algorithm for the maximal information coefficient and its application**
Dan Cao, Yuan Chen, Jin Chen, Hongyan Zhang, Zheming Yuan
Royal Society Open Science (2021-02) https://doi.org/gpcwkd
DOI: 10.1098/rsos.201424 · PMID: 33972855 · PMCID: PMC8074658

47. **A New Algorithm to Optimize Maximal Information Coefficient**
Yuan Chen, Ying Zeng, Feng Luo, Zheming Yuan
PLOS ONE (2016-06-22) https://doi.org/gbpjt7
DOI: 10.1371/journal.pone.0157567 · PMID: 27333001 · PMCID: PMC4917098

48. **RapidMic: Rapid Computation of the Maximal Information Coefficient**
Dongming Tang, Mingwen Wang, Weifan Zheng, Hongjun Wang
Evolutionary Bioinformatics (2014-01) https://doi.org/gpt7c8
DOI: 10.4137/ebo.s13121 · PMID: 24526831 · PMCID: PMC3921152

49. **A Novel Algorithm for the Precise Calculation of the Maximal Information Coefficient**
Yi Zhang, Shili Jia, Haiyun Huang, Jiqing Qiu, Changjie Zhou
Scientific Reports (2014-10-17) https://doi.org/gpt7c7
DOI: 10.1038/srep06662 · PMID: 25322794 · PMCID: PMC4200418

Supplementary material

Supplementary Note 1: Comparison with the Maximal Information Coefficient (MIC) on gene expression data

We compared all the coefficients in this study with MIC [pmid:22174245?], a popular nonlinear method that can find complex relationships in data. This method is, however, very computationally intensive [46]. To further investigate this, we ran MIC_e (see Methods) on all possible pairwise comparisons of our 5,000 highly variable genes from whole blood in GTEx v8. This took 4 days and 19 hours to finish (compared with 9 hours for CCC). We then analyzed the distribution of coefficients (the same as in the main text), shown in Figure 6. We observed that CCC and MIC behave similarly in this dataset, with essentially the same distribution but only shifted. Figure 6 c shows that these two coefficients have a nearly linear relationship, and both compare very similarly with Pearson and Spearman.

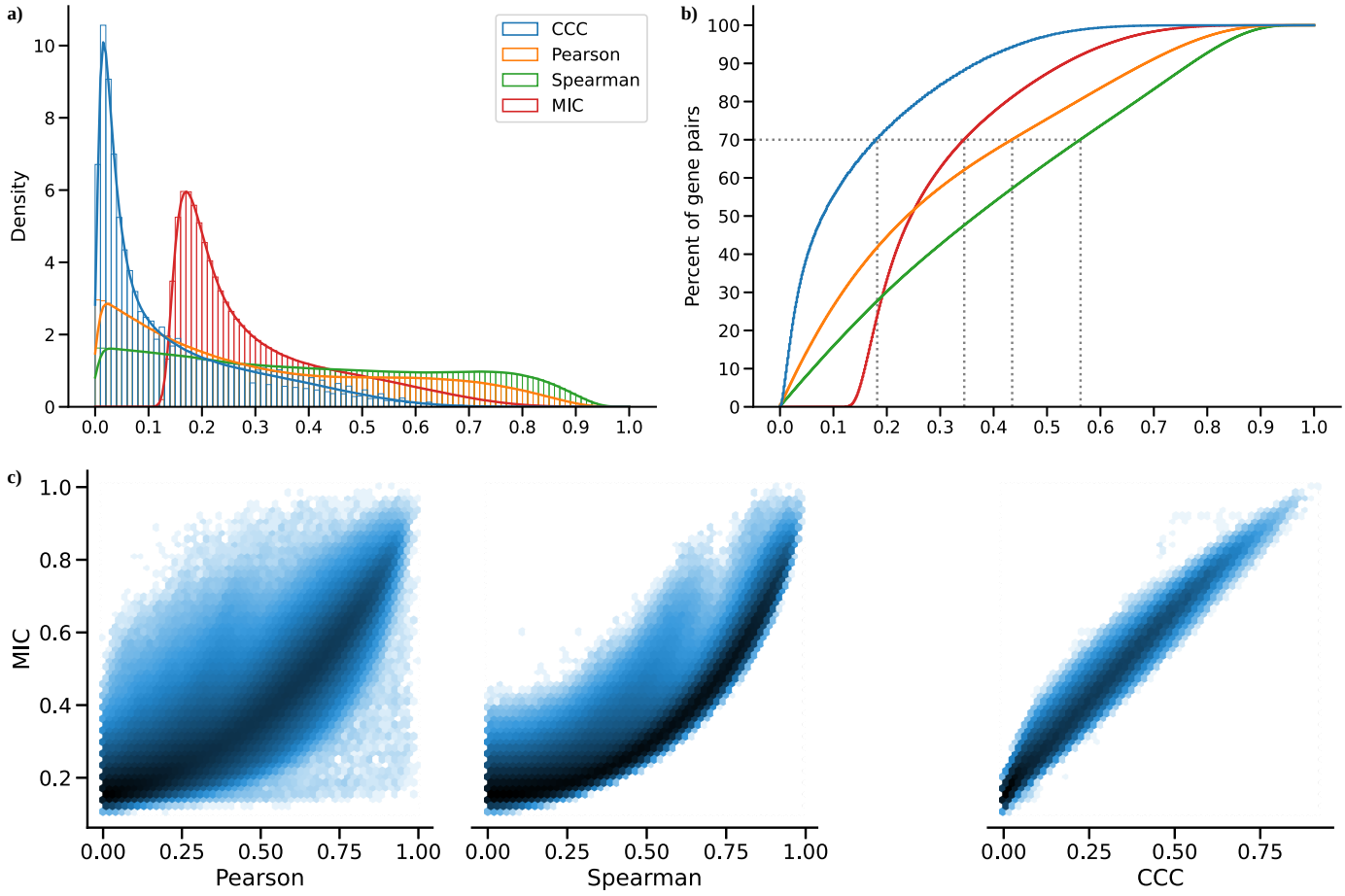

Figure 6: Distribution of MIC values on gene expression (GTEx v8, whole blood) and comparison with other methods. a) Histogram of coefficient values. **b)** Corresponding cumulative histogram. The dotted line maps the coefficient value that accumulates 70% of gene pairs. **c)** 2D histogram plot with hexagonal bins between all coefficients, where a logarithmic scale was used to color each hexagon.

Supplementary Note 2: Computational complexity of coefficients

We compared our CCC with other coefficients in terms of computational complexity. We found that, while CCC and MIC might identify similar gene pairs in gene expression data (see here), the use of MIC in large datasets remains limited due to its very long computation time, despite some methodological/implementation improvements [42,46,47,48,49]. The original MIC implementation uses ApproxMaxMI, a computationally demanding heuristic estimator [16]. Recently, a more efficient implementation called MIC_e was proposed [44]. These two MIC estimators are provided by the `minepy` package [42], a C implementation available for Python. To compare the computation time of all these coefficients, we used randomly generated variables of different sizes, which simulates a gene expression data scenario with different numbers of conditions. CCC allows for easy parallelization of the computation of a single gene pair (see Methods), so we tested the cases using 1 and 3 CPU cores. The results are shown in Figure 7, which displays the time in seconds in log scale.

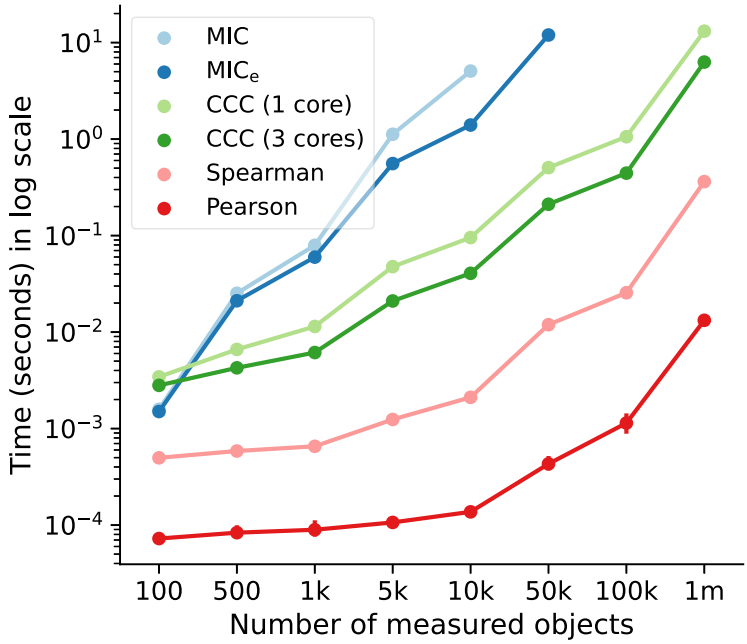

Figure 7: Computational complexity of all correlation coefficients on simulated data. We simulated variables/features with varying data sizes (from 100 to a million, x-axis). The plot shows the average time in seconds (log-scale) taken for each coefficient on ten repetitions (1000 repetitions were performed for data size 100). CCC was run using 1 and 3 CPU cores. MIC and MIC_e did not finish running in a reasonable amount of time for data sizes of 10,000 and 100,000, respectively.

Figure ??? shows the average time performance of the different correlation coefficients.

Pearson and Spearman correlation coefficients are the fastest, since they only require computing basic summary statistics from the data. CCC was faster than the two MIC variations, except in very small data sizes, where it was up to two orders of magnitude slower. This is because CCC was implemented in Python (optimized with `numba`) while the MIC variants were implemented in C [42], a high-performance programming language. With a data size of a million, the multi-core CCC was twice as fast as the single-core CCC. This suggests that using more advanced processing units (such as GPUs) could make CCC reach speeds closer to Pearson. Figure ??? shows the average time performance of the different correlation coefficients.

Tissue-specific gene networks with GIANT

	Interaction confidence						
	Blood			Predicted cell type			
Gene	**Min.**	**Avg.**	**Max.**	**Cell type**	**Min.**	**Avg.**	**Max.**
IFNG	0.19	0.42	0.54	Natural killer cell	0.74	0.90	0.99
SDS	0.18	0.29	0.41		0.65	0.81	0.94
JUN	0.26	0.68	0.97	Mononuclear phagocyte	0.36	0.73	0.94
APOC1	0.22	0.47	0.77		0.29	0.50	0.80
ZDHHC12	0.05	0.07	0.10	Macrophage	0.03	0.12	0.33
CCL18	0.74	0.79	0.86		0.36	0.70	0.90
RASSF2	0.69	0.77	0.90	Leukocyte	0.66	0.74	0.88
CYTIP	0.74	0.85	0.91		0.76	0.84	0.91
MYOZ1	0.09	0.17	0.37	Skeletal muscle	0.11	0.11	0.12
TNNI2	0.10	0.22	0.44		0.10	0.11	0.12
PYGM	0.02	0.04	0.14	Skeletal muscle	0.01	0.02	0.04
TPM2	0.05	0.56	0.80		0.01	0.28	0.47

We present in Table ?? the network statistics of six gene pairs shown in Figure 3 b for blood and predicted cell types. The table lists only the gene pairs included in the GIANT models. For each gene in the pair (first column), the table provides the minimum, average and maximum interaction coefficients with the other genes in the network.