RESEARCH

Supplementary information for: A new set of cluster driven composite development indicators

Anshul Verma[1*†], Orazio Angelini[1†] and Tiziana Di Matteo[1,2,3]

[*] Correspondence:
anshul.verma@kcl.ac.uk
[1] Department of Mathematics,
King's College London, Strand,
WC2R 2LS London, UK
Full list of author information is
available at the end of the article
[†] Equal contributor

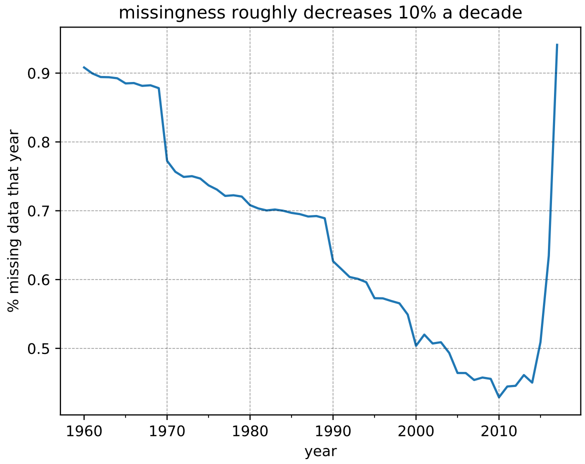


Figure 1: Missing data percentage per year, all years (1963-2017), all countries.

1 Cleaning procedure

1.1 Imputation

The WDI dataset suffers from high levels of missing data. We solved this problem with a combination of removal and imputation of datapoints. For starters, the amount of missing data decreases in time, as can be seen in Fig. 1. We decided to use the last 20 years of data, which have the least amount of missing datapoints in the dataset, so to not have to deal with missingness values above 50%.

We considered the possible bias of the dataset due to the fact that data is not missing at random. In fact, it can be seen from Fig. 2 that the amount of missing data a country has is correlated, sometimes strongly, with the values of some of its indicators. It seems that the dataset is biased towards industrialized and more developed countries. While this might cause problems when one tries to make predictions out of the data, we believe the results about the existence of a correlation structure in the data are affected little by this.

The remaining data still has a high amount of missingness. We therefore proceded to impute it. We tested several algorithms on the dataset, readily available from the Fancyimpute python package [1]. They cover mostly matrix factorization approaches to imputation: SoftImpute [2], IterativeSVD [3] and MatrixFactorization [1] are all based on this principle. SimpleFill consists in replacing missing entries with the median, and KNN is K-Nearest Neighbours [4]. In Table 1 we report the Mean Average Error (MAE) and Mean Square Error (MSE) for the techniques adopted (obtained by holding out 0.5% of the data to test the quality of the results). Interestingly, the best performing technique is K-Nearest Neighbours (KNN). This is in line with the result of [5], which predicts GDP change over time for a country by averaging the past GDP changes of similar countries, where similarity is measured as an euclidean distance on a space defined by two macroeconomic indicators. This agreement might point to the fact that the most reliable way to model a country is by its similarity to similar countries already observed. The only metaparameter for the KNN algorithm (D, the number of neighbours to average)

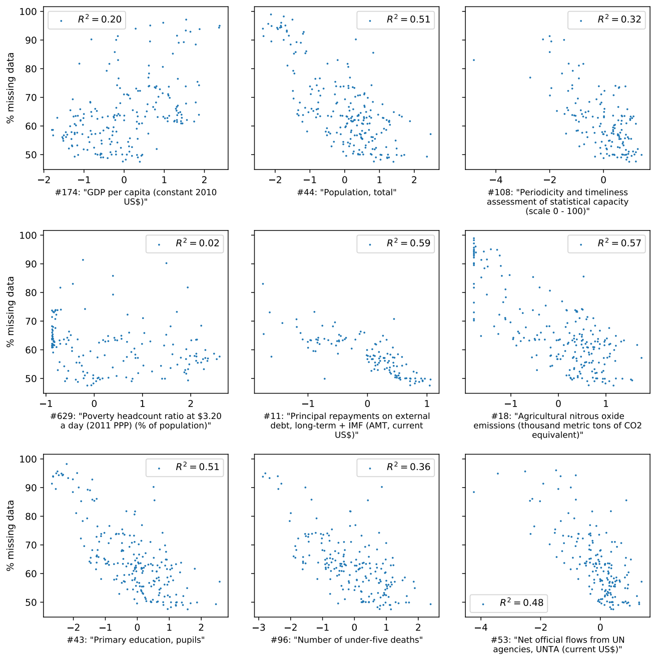


Figure 2: Correlation between percentage of missing points for a country and the value of an indicator, all years (1963-2017), all countries.

has been chosen by means of grid searching on logarithmically separated values of D and testing on a holdout set of size 0.5%. Table 2 shows that the best value for D is either 2 or 4, depending on whether one minimizes MAE or MSE. We chose the average, $D = 3$. We have checked that the results do not change qualitatively if $D = 2$ or $D = 4$ is chosen.

imputer	MAE	MSE
KNN	0.032636	0.088496
SoftImpute	0.061094	0.112322
IterativeSVD	0.112888	0.181256
MatrixFactorization	0.130820	0.200824
SimpleFill	0.238268	0.321349

Table 1: The Mean Average Error (MAE) and Mean Square Error (MSE) in the second and third columns for the different imputation schemes in the first column.

We have investigated the influence of missing data on the results by adding a random white noise, with the value of the variance given by the Mean Square Error (MSE) when $K = 3$ (the parameter of the KNN which optimises the MSE). We then recalculated the correlation matrix and reapplied the Directed Bubble Hierarchical Tree (DBHT) algorithm. Comparing the two different set of clusters tests whether the value imputed by the KNN is accurate. Practically, this comparison is achieved by applying a similar procedure that is detailed in section 4.2 for comparing the DBHT clustering to the topics to see if the clusters in the imputed data are overexpressed. This overexpression indicates statistically whether our imputed data clusters are indeed present in the random data clusters. With a p-value of 1.80×10^{-6}, which is the p-value of 0.01 modified by Bonferoni correction, that all

D	MAE	MSE
1	0.033483	0.102603
2	0.030785	0.091696
3	0.031161	0.090172
4	0.031588	0.087745
5	0.032636	0.088496
6	0.033698	0.089346
8	0.036122	0.091546
11	0.039479	0.094866
14	0.042721	0.097449
18	0.046539	0.100678
23	0.050710	0.104088
29	0.055189	0.108290
37	0.060167	0.113266
48	0.065573	0.119146
61	0.070659	0.124860
78	0.075911	0.131326
100	0.081211	0.137930

Table 2: Mean Average Error (MAE) (second column) and Mean Squared Error (MSE) (last column) when varying the number of neighbours to average D of the KNN algorithm (first column) using a 0.5% holdout set size.

of the clusters in the imputed data are overexpressed and thus also present into the random data. We also tested whether the KNN algorithm for imputation is appropriate and does not significantly affect the results. This is accomplished by instead replacing the missing data with random white noise with mean 0 and variance 1 (since the indicator data are standardised before being imputed). Like before, we recalculate the correlation matrix, apply the DBHT algorithm and compare the two sets of clusters through the same procedure with a p-value of 1.68×10^{-6}. Here, we find 96 of the original 102 clusters are overexpressed and thus present in this new random data. This is quite high considering that the assumption tested and total replacement of missing data (rather than just adding noise with a lower standard deviation) here is stronger than before.

Since the formation of the CDCIs relies on the clusters present in the data, we can therefore safely conclude that missing data does not change the final results significantly.

1.2 Distribution regularisation

Another characteristic of the WDI dataset is the heterogeneity of the value distributions across different indicators. For example, many indicators are percentages, and as such are bounded between the values 0 and 100. Long-tailed distributions are very common, as well as some that might remind Gaussian distributions. A sample of these distributions can be seen in Fig. 3. We applied mathematical transformations to some of the indicators, in order to change their distribution and have a more homogeneous and tractable dataset.

We applied one of three possible transformations to each indicator. The first possibility is the identity function, i.e. we left the values unchanged. The second consists in taking the base-10 logarithm of the modulus of each indicator's value. The third is the *bisymmetric log transformation* [6].

$$\mathrm{logbisymmetric}_b(x) = \mathrm{sign}(x) * \log_b(1 + |x|) \tag{1}$$

Given the high number of indicators and the need to avoid arbitrary decisions, the decision of what transformations to apply to each indicator has been made through an algorithm. To understand the criteria used, we will introduce first the definition of *span* of a set of numbers X. We define span as:

$$\text{span}(X) = \max_{x \in X}(\log_{10}(|x|)) - \min_{x \in X}(\log_{10}(|x|)) \tag{2}$$

In order to decide what transformation to apply to each indicator, we consider the set of all values for that indicator found in the dataset, X. We then define two quantities. The first we will call *in-span*, which is the span for the subset of values x found in X such that $-1 < x < 1$. The second is the *out-span*, i.e. the span for all values of X that are outside the $[-1, 1]$ interval:

$$\begin{aligned} \text{inspan}(X) &= \text{span}(x | x \in (X \cap (-1, 1)) \\ \text{outspan}(X) &= \text{span}(x | x \in X \setminus [-1, 1])) \end{aligned}$$

Then, the algorithm for assigning the transformation is this:

```
Result: What kind of transformation to apply to the indicator.
given a set of numbers $X$;
compute bothsigns($X$) = whether $X$ contains both numbers $> 0$ and $< 0$;
compute haszeros($X$) = whether $X$ contains the value 0;
compute inspan($X$), outspan($X$);
if $outspan(X) > 2$ then
    if $inspan(X) > 2$ then
        if not haszero($X$) and not bothsigns($X$) then
            apply $\log_{10}(|X|)$;
        else
            apply identity($X$);
        end
    else
        apply logbisymmetric$_{10}(X)$;
    end
else
    apply identity($X$);
end
```

Algorithm 1: The algorithm used to choose which transformation to apply to each individual indicator, given the set of values X from its empirical distribution

The rationale behind this algorithm is that frequently the values in X span a large number of orders of magnitude, and in this case we want to transform them so that their distribution is easier to manage with linear techniques such as PCA or factor models. If the numbers are all of the same sign and there is no zero in X, one can directly take the logarithm; otherwise we apply the log-bisymmetric transformation, which has no singularity on the zero and is defined for negative numbers.

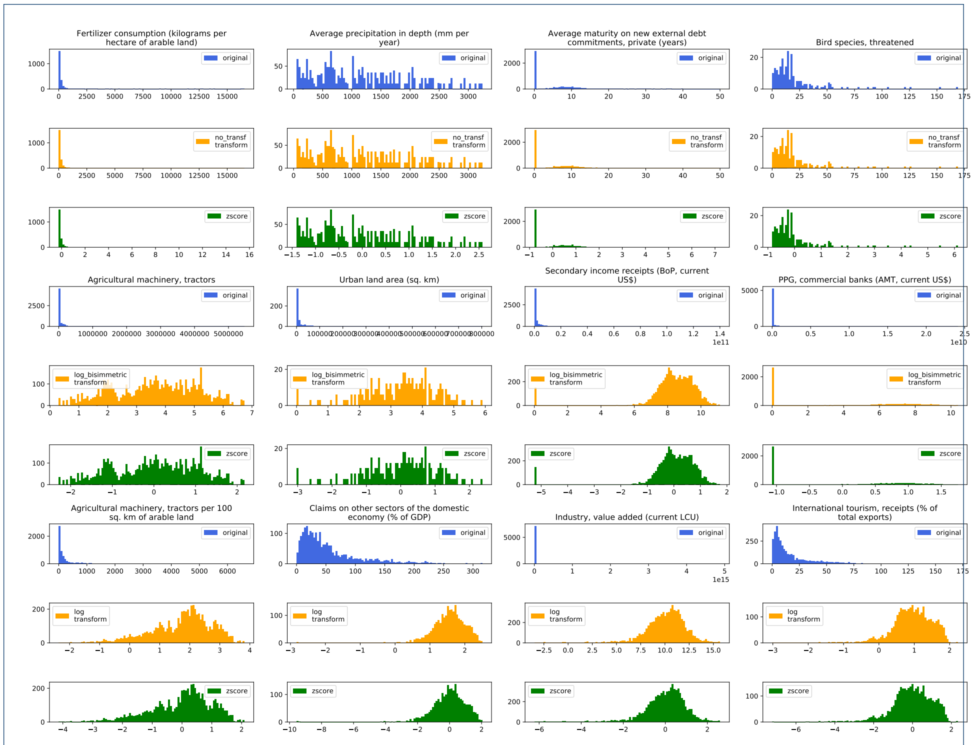


Figure 3: Examples of the transformations applied to indicators and how they transform their distribution.

After transforming the dataset with this algorithm, we z-score each indicator individually, so to set the mean to zero and the standard deviation to one. A sample of the results of this procedure can be seen in Fig. 3.

2 Eigenvalue Spectrum

Firstly, we should only extract components of $\boldsymbol{E}$ that describe relevant interactions between indicators. The question then arises about how many principal components we keep [7]. This directly controls the size of the reduced correlation matrix - which we would like to be as small as possible - versus the fraction of the total variance of the indicator system that the reduced matrix can explain. It would also help us in identifying what economic indicators are responsible in driving the indicator system by analysing which are the main contributing indicators to the top eigenvalues.

The eigenvalues, however, could also be affected by noise from taking a finite sample [8]. We should therefore first study the empirical distribution of the eigenvalues, identifying those eigenvalues which are just noise and discarding them. To identify noisy eigenvalues, we will need a null distribution, produced from a Gaussian white noise process. The answer is provided by the well-known Marčenko-Pastur (MP) distribution [9], given by

$$p(\lambda) = \frac{1}{2\pi q \sigma^2} \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\lambda} \; , \tag{3}$$

where $p(\lambda)$ is the probability density of eigenvalues having support in $\lambda_- < \lambda < \lambda_+$. The edge points $\lambda_\pm = \sigma\left(1 \pm \sqrt{q}\right)^2$, $q = N/Y$ and σ is the standard deviation over all indicators. If we compare the distribution in Eq. (3) to the empirical eigenvalue distribution of $\mathbf{E}$, we will be able to see how many components are indistinguishable from noise, often called the 'bulk' eigenvalues. These are then discarded. In practice, this is achieved by fitting Eq. Eq. (3) to the eigenvalues of $\mathbf{E}$, with q and σ acting as free parameters. The results are shown in Fig. 4a which compares the empirical histogram of eigenvalues of $\mathbf{E}$ and the best fit MP distribution in red, giving 216 components beyond the upper limit of the MP distribution. Whilst this number appears large, it still means that we can reduce the size of the correlation matrix by 85% before we start to include components which statistically can be seen as noise. Further methods can be used to reduce the number of components further e.g. cross validation or cumulative variance [7], and also [10].

However, by comparing the best fit MP distribution for our dataset in red in Fig. 4a we see that in fact there seems to be a noticeable deviation of the bulk eigenvalues from the MP distribution, so we can infer that the MP distribution may not be suitable in identifying noisy eigenvalues. We also notice that the best value of q is noticeably different than the theoretical value for this dataset of 0.35 indicating a significant difference in the predicted properties of the bulk using Eq. (3). Indeed, the use of the MP distribution in this respect has been questioned more recently [11, 12, 13] for financial data at least. Moreover, it would also indicate that there could actually be some structure hidden within the bulk eigenvalues. We test this by shuffling our differenced indicator data, recalculating the correlation matrix and again finding the best MP fit to the eigenvalue distribution of this new correlation matrix. In doing so, we destroy the correlations between indicators, therefore testing whether these are the cause of the differences seen in the bulk in Fig. 4a. The results are reported in Fig. 4b, where the histogram of the eigenvalues coming from the new correlation matrix is in blue bars and the best MP fit for this given in red. We can clearly see an almost perfect fit in this case of the MP fit and q much closer to theoretical value which Eq. (3) predicts, which suggests that indeed the earlier bulk eigenvalues are a result of non-trivial strucutre within $\mathbf{E}$, and are not just random fluctuations in the data. Overall, these two results together suggest that there is no natural way to a select a subset of principal components without loosing non-trivial information, which may make PCA an unsuitable method of dimensionality reduction for this dataset.

Nevertheless, as the inset plot in Fig. 4a shows, there are some eigenvalues whose magnitude is 2 times greater than that of some of the smaller eigenvalues e.g. the first principal component has an eigenvalue of 94. These eigenvalues from the perspective of PCA are the most important eigenvalues since they make the biggest contributions to the overall variance of the system. They are also well separated from the bulk, which means that they are less affected by noise and will have a clearer, more discernible interpretation [14].

3 Procedure for calculating the p-values of ρ_g

Here we detail the procedure used to calculate the p-values used to produce Table 1. Under the null hypothesis that $\boldsymbol{\rho}_i$ is random, the entries of $\Delta\boldsymbol{X}$ will be i.i.d normally distributed with mean 0 and standard deviation of 1. We can therefore use the exact same definition given in Eq. (3) but with a randomly generated $\Delta \boldsymbol{X}$ to produce an instance of $\boldsymbol{\rho}_i$ under the null hypothesis. One can then estimate the empirical cumulative distribution function [15] of each entry $\rho_{i,g}$ by repeating this process many times and aggregating the results with the same g. For Table 1, we repeat the process 1000 times.

4 PMFG network

Here, we report the visualisation of the PMFG network computed on $\mathbf{E}$ in Fig. 5. From the PMFG, we can observe that there a few hubs of nodes which are connected to other less connected nodes, which is consistent with the observations from other complex networks in different contexts.

5 Elastic net regression

Elastic net regression is used to find the values of β_{ik} from Eq. (5). Further details of the use of this method is provided in this section. Elastic net regression [16] is a hybrid version of ridge regularisation and lasso regression, thus providing a way of dealing with correlated explanatory variables (in our case $I_k(t)$ and $I_{k'}(t)$) and also performing feature selection, which takes into account non-interacting clusters $I_{k'}(t)$ that ridge regularisation would ignore. Elastic net regression solves the constrained minimisation problem

$$\min_{\boldsymbol{\beta}_i} \frac{1}{Y} \sum_{y=1}^{Y} \left(\Delta \boldsymbol{X}(y,i) - \boldsymbol{I}^\dagger \boldsymbol{\beta}_i \right)^2 + \lambda P_a(\boldsymbol{\beta}_i) \ , \tag{4}$$

where $\boldsymbol{\beta}_i$ is the vector of loadings given by $(\beta_{i1}, \beta_{i2}, \ldots, \beta_{iK})^\dagger$, $\boldsymbol{I}$ is the matrix consisting of columns $(I_1(t), I_2(t), \ldots, I_K$ and λ and a are hyperparameters. $P_a(\boldsymbol{\beta}_i)$ is defined as

$$P_a(\boldsymbol{\beta}_i) = \sum_{k=1}^{K} \left((1-a)\frac{\beta_{ik}^2}{2} + a|\beta_{ik}| \right) \ . \tag{5}$$

The first term in the sum of Eq. (5) is the L_2 penalty for the ridge regularisation and the second term in the sum is the L_1 penalty for the lasso regression. Hence if $a = 0$ then elastic net reduces to ridge regression and if $a = 1$ then elastic net becomes lasso, with a value between the two controlling the extent which one is preferred to the other. The determination of the a hyperparameter, controlling the extent of lasso vs ridge, and λ, for the ridge, is done using 10 cross validated fits [16], picking the pair of (a, λ) that give the minimum prediction error.

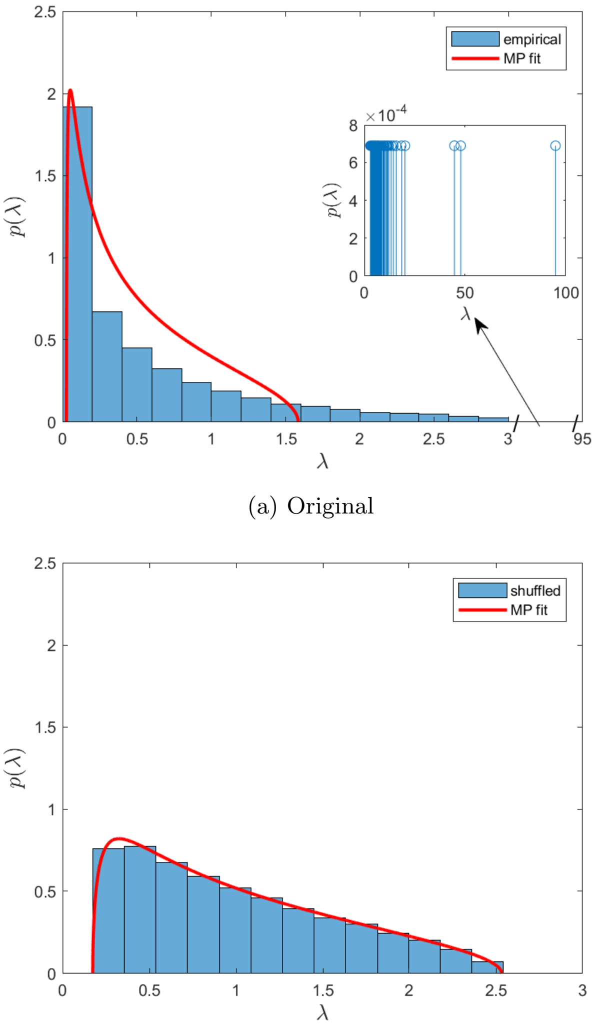
Author details

[1]Department of Mathematics, King's College London, Strand, WC2R 2LS London, UK. [2]Department of Computer Science, University College London, Gower Street, WC1E 6BT London, UK. [3]Complexity Science Hub Vienna, Josefstadter Strasse 39, A 1080 Vienna, Austria.

References

1. Rubinsteyn, A., Feldman, S., O'Donnell, T., Beaulieu-Jones, B.: hammerlab/fancyimpute: Version 0.2.0 (2017). doi:10.5281/zenodo.886614. http://dx.doi.org/10.5281/zenodo.886614
2. Mazumder, R., Hastie, T., Tibshirani, R.: Spectral regularization algorithms for learning large incomplete matrices. Journal of machine learning research **11**(Aug), 2287–2322 (2010)

3. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for dna microarrays. Bioinformatics **17**(6), 520–525 (2001)
4. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer **27**(2), 83–85 (2005)
5. Tacchella, A., Mazzilli, D., Pietronero, L.: A dynamical systems approach to gross domestic product forecasting. Nature Physics **14**(8), 861 (2018)
6. Webber, J.B.W.: A bi-symmetric log transformation for wide-range data. Measurement Science and Technology **24**(2), 027001 (2012)
7. Jolliffe, I.: Principal Component Analysis. Wiley Online Library, Hoboken (2002)
8. Plerou, V., Gopikrishnan, P., Rosenow, B., Amaral, L.A.N., Guhr, T., Stanley, H.E.: Random matrix approach to cross correlations in financial data. Physical Review E **65**(6), 066126 (2002)
9. Marčenko, V.A., Pastur, L.A.: Distribution of eigenvalues for some sets of random matrices. Sbornik: Mathematics **1**(4), 457–483 (1967)
10. Verma, A., Vivo, P., Di Matteo, T.: A memory-based method to select the number of relevant components in principal component analysis. Journal of Statistical Mechanics: Theory and Experiment **2019**(9), 093408 (2019)
11. Guhr, T., Kälber, B.: A new method to estimate the noise in financial correlation matrices. Journal of Physics A: Mathematical and General **36**(12), 3009 (2003)
12. Livan, G., Alfarano, S., Scalas, E.: Fine structure of spectral properties for random correlation matrices: An application to financial markets. Physical Review E **84**(1), 016113 (2011)
13. Wilinski, M., Ikeda, Y., Aoyama, H.: Complex correlation approach for high frequency financial data. Journal of Statistical Mechanics: Theory and Experiment **2018**(2), 023405 (2018)
14. Bun, J., Bouchaud, J.-P., Potters, M.: Cleaning large correlation matrices: tools from random matrix theory. Physics Reports **666**, 1–109 (2017)
15. Van der Vaart, A.W.: Asymptotic Statistics vol. 3. Cambridge university press, ??? (2000)
16. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(2), 301–320 (2005)



(a) Original

(b) Shuffled

Figure 4: (Top) a is the histogram of the empirical eigenvalue distribution for $\mathbf{E}$ in blue bars, with the best MP fit in red. The best MP fit has values $q = 0.59 \pm 0.025$ and $\sigma = 0.71 \pm 0.006$. The inset plot are the top 100 eigenvalues. (Bottom) b is the histogram of the eigenvalue distribution but of the correlation matrix when we shuffle the data. The corresponding MP fit is also given in red, for which $q = 0.34 \pm 0.027$ and $\sigma = 1.00 \pm 0.017$.

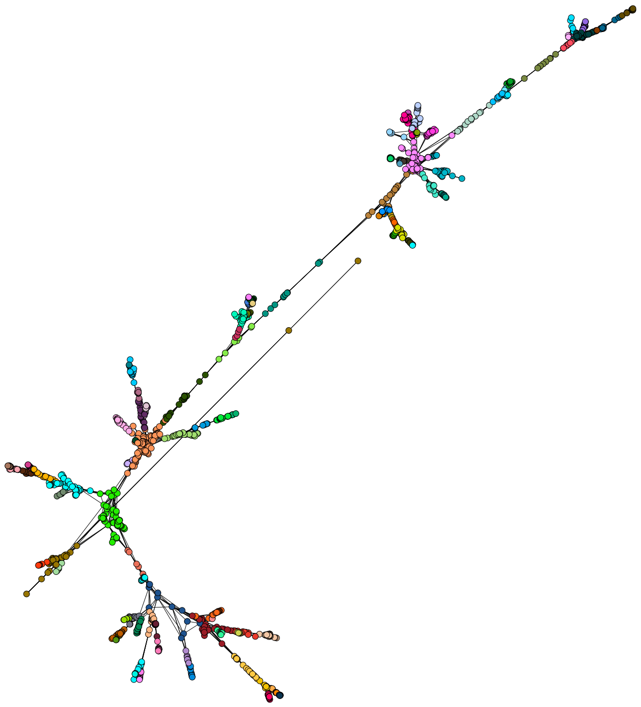


Figure 5: The PMFG of **E**, with the colour of each node representing cluster membership according the DBHT algorithm.