A Measure of the Robustness of Clusters in a Network with No Ground Truth - A Chronic Lower Back Pain Case Study

Iris Ho
Computer Science and Software Engineering
California Polytechnic State University
San Luis Obispo, CA, USA
iwho@calpoly.edu

Paul Anderson
Computer Science and Software Engineering
California Polytechnic State University
San Luis Obispo, CA, USA
pander14@calpoly.edu

Jean Davidson
Department of Biological Sciences
California Polytechnic State University
San Luis Obispo, CA, USA
jdavid06@calpoly.edu

Jeffrey Lotz
Department of Orthopaedic Surgery
School of Medicine, University of California San Francisco
San Francisco, CA, USA
jeffrey.lotz@ucsf.edu

Theresa Migler
Computer Science and Software Engineering
California Polytechnic State University
San Luis Obispo, CA, USA
tmigler@calpoly.edu

***Abstract*—We present a metric for evaluating the robustness of clusters as information is removed. Utilizing the edit distance between two clusterings as a measure of stability, we test this metric on an Erdos-Renyi random graph and on four different networks representing the connections between patients who experience chronic lower back pain. On all graphs, we utilize both k-means and the Louvain algorithm for cluster identification to test the metric.**

***Index Terms*—clustering, network metrics, cluster robustness, cluster stability, edit distance**

I. Introduction

Clustering algorithms have many different metrics associated with them. However, these metrics often describe the current state of the cluster and how well individuals are clustered. In a complex domain, such as chronic pain, where there is no ground truth for clustering, it is not only valuable to know how distinct the clusters are, but also important to understand how robust a given cluster is.

This paper proposes a simple method of evaluating the robustness of graphical clustering approaches. Given the graph with all the available data (the "true" graph) and a clustering algorithm, we treat the clusters produced by the clustering algorithm on the complete graph as the "right" clusters. Then, we remove some of the edges and using the same clustering technique, we determine how many nodes have changed clusters. We analyze this method on a case study of the network of patients with chronic lower back pain.

NIH Heal Initiative, BACPAC

II. Background

Given a graph, the nodes can be partitioned, or *clustered*, based on their properties and connections.

For example, we might have a graph where the nodes are patients and two patients are connected if they experience the same symptoms. A clustering algorithm might cluster these patients based on the severity of their symptoms, but an equally valid grouping would be by their response to treatment.

There are various methods to approach clustering nodes in a graph, here we discuss k-means and Louvain clustering.

A. Clustering Algorithms

The k-means clustering algorithm, also referred to as the Lloyd–Forgy algorithm, aims to divide the data set into k clusters, where k is a parameter for the algorithm [6], [16], [17]. In addition to k, the algorithm also takes in a set of vectors. The input vectors can be any dimension but all vectors must be the same dimension. Thus to run k-means clustering on a graph, each node must be represented as a vector. This is accomplished with the help of node embedding algorithms, such as node2vec and GRAPHsage [10], [12].

One way to measure the quality of the clustering is through the average *silhouette score* of every point, which is a measure of a how well clustered a node is [22]. The silhouette score has a range of -1 to 1, where a higher value is associated with the node being closer to values in their cluster as opposed to neighboring clusters. Simply put, a silhouette score of -1 for a node implies that the node is likely in the wrong cluster

and a score of 1 means that the node is in the best cluster it could possibly be in. When utilizing k-means, it is common to try a range of values for k and pick the best k value based on the silhouette score and/or the clusters that make the most sense based on the domain of objects that are being clustered, especially if there is a notion of what the right clusters are.

Another clustering method is Louvain, which is a hierarchical clustering algorithm that aims to maximize the *modularity score* for each community [2]. The modularity score measures how densely connected the nodes within a community are, which is similar to the silhouette score. However, the modularity score is focused on the internal structure of the cluster whereas the silhouette score is a measure of how homogenous elements in a cluster are. Modularity is on a scale of 0 to 1 with 1 being the most modular.

B. Evaluating and Comparing Clusters

In general, cluster validation falls into three categories: relative cluster validation, external cluster validation, and internal cluster validation. Relative cluster validation is when an algorithm is executed with varying parameters and their results are compared to each other, such as trying different k values for the k-means clustering algorithm. External cluster validation is when we compare a given clustering to an external source of clustering. This is mainly used when there are known "true" clusters or there are otherwise external criteria known about the clusters. External cluster validation includes methods such as the Rand Index [20], Adjusted Rand Index [20], Jaccard Index [14], and Fowlkes-Mallows Index [9]. Internal cluster validation is then useful when there are no known clusters and utilizes internal clustering metrics to determine the goodness of the clusters. Methods of internal cluster validation include silhouette score [22], Davies-Bouldin index [5], Dunn index [7], [8], and Calinski-Harabasz index [3].

Related to cluster validation, but distinctly different is cluster stability or *robustness*, which quantifies the consistency of cluster assignments across multiple runs of the clustering algorithm or variations in the dataset. Such methods include Jaccard Index [14], Adjusted Rand Index [20], and Cophenetic Correlation Coefficient [23].

Given two different partitions, A and B, of the nodes in a graph, we can calculate the *edit distance*, which is the minimum number of edits needed to transform from one partitioning to the other. Note that $edit(A, B) = edit(B, A)$. This edit distance would then represent the number of nodes that are in different clusters in the two partitions. Thus, the number of nodes that are in different clusters is the same as all the nodes minus the nodes that are in the same clusters. With that in mind, we can transform this edit distance problem into an assignment problem where the goal is to find the maximum cost assignment of clusters in A to clusters in B. The cost of assigning cluster A_i to cluster B_j is then $|A_j \cap B_j|$. Thus, solving the assignment problem would produce the number of nodes that are in the same cluster.

III. Related Work

Clustering in biological networks is desired for many reasons. In the context of chronic pain, many others have utilized clustering to identify treatment groups.

Tagliaferri et al. used data from the UKBioBank and found five groups of chronic back pain patients using machine learning mainly characterized by a varying combination of social isolation and depressive symptoms [24]. Another study by Larsson et al. identified four groups of chronic pain among Swedish older adults distinguished by varing degrees of pain and psychological symptoms [15]. Larsson et al. utilized two-step cluster analysis (TSCA), which consisted of pre-clustering and then hierarchical methods on the basis of best fit. Lastly a 2018 study by Bäckryd et al. using psychometric data from the Swedish quality registry for pain rehabilitation (SQRP), identified that chronic pain patients belong to one of four groups [1]. In a past research paper, we proposed graphical clustering methods with some success [13]. To our knowledge, we are the first to address the *stability* of the proposed clusters.

We acknowledge that our method for measuring cluster stability is simple, but we find value in the simplicity given the complexity of our domain.

IV. Methods

A. Data

The data utilized in the case study for this study is based on The Longitudinal Clinical Cohort for Comprehensive Deep Phenotyping of Chronic Low-Back Pain (cLBP) Adults Study (comeBACK). This longitudinal multicenter observational study was designed to perform comprehensive deep phenotyping in patients with cLBP and was conducted at 4 clinical sites in the United States (U.S.) with a coordinating center at UCSF. The comeBACK clinical sites are located at four of the University of California campuses, including UC San Francisco (UCSF), UC Davis, UC Irvine, and UC San Diego. Recruitment for the study commenced in March 2021 and was completed in June 2023 with a total of 450 participants enrolled and to be followed for up to 2 years. Participants attend in-clinic baseline and annual visits (on month 12 and 24). Remote (via online surveys with a link sent by email and/or phone) visits occur at Months 1, 2, 3, 4, 5, 6, 18, and at months 7-8, if necessary.

The comeBACK dataset is not publicly available yet. Thus, we used CTGAN (Conditional Tabular Generative Adversarial Network) to generate synthetic data that mimics the characteristics of the original dataset. Specifically, data for 1,000 patients was synthesized.

Given the focus of this paper, we decided to use a small subset of the data collected in comeBACK. More specifically, we decided to use age, sex, BMI, pain duration (on a scale of 1-5), pain frequency (on a scale of 1-3), and pain intensity (on a scale of 0-10).

B. Graph Construction

Given this data, we constructed a graph such that nodes represented patients and two patients are connected by an edge

TABLE I: Edge Types

Edge Type	Possible Values	Similar If…
Age	1 0-95	± 5
Sex	1 (Male), 2 (Female), 3 (Unknown)	± 0
BMI	11-44	± 2
Pain Duration	1-5	± 0
Pain Frequency	1-3	± 0
Pain Intensity	0-10	± 1

of type $type$ if they shared a similar value for the data $type$ variable. Since each metric's range of values varies, "similar" will mean different things for each metric. Table I specifies what 'similar' means for each metric in the column 'Similar If'.

For example, if patient a had a pain duration value of 1 and patient b had a pain duration value of 2, they would not share an edge since their similarity is within ± 1 and not ± 0. In other words a ± 0 value means an edge will only be added between two patients if they have the exact same value. However, if patient a had an intensity value of 3 and patient b had an intensity value of 4, then they would be connected by an intensity edge since they are within 1 numerical value from each other.

Allowing "similar" numerical scores in a category to share an edge accounts for varying interpretations of potentially the same levels of pain. For example, a pain intensity score of 1 is likely similar to a pain intensity of 2.

We experimented with two variations of graph construction: one, G_e, where each data variable was an edge type (for a total of six different edge types) and another, G_m, where some variables (age, sex, and bmi) were node properties (leaving pain duration, frequency, and intensity to be edge types).

Due to the nature of the different graphs, we ran Louvain community detection and node2vec followed by k-means clustering on the first graph, G_e, where all data was represented in the edges; and we ran node2vec and graphSAGE both followed by k-means clustering on the second graph, G_m, where the data was represented as a mix of node properties and edges. Since node2vec is unable to accommodate for node properties, this means that when it is run on G_m, node attributes are being ignored and therefore unused.

We chose to implement our graphs in Python using in NetworkX [11], and we utilized node2vec, scikit-learn [19], and stellargraphs [4] for node embedding and clustering algorithm implementations.

C. Using Similarity to Measure Robustness

Given two potential clusterings, A and B, the edit distance between the A and B can be interpreted as a measure of similarity – where the smaller the edit distance is, the less differences there are between the two clusterings, and thus the more similar A and B are.

We propose that a clustering algorithm can be considered robust if clusterings are relatively similar when a majority of the information is still available. Thus, we can repeatedly remove some information from a graph, re-cluster, and compute the edit distance between the new clustering and the original clustering. We define the *similarity score*, or *similarity percentage* as the maximum number of individuals that stayed in the same clustering, based on the edit distance, divided by the total number of individuals.

This simple, and perhaps naive, definition of cluster similarity is helpful in the context of a complex domain where transparency is not only valuable but also necessary. We acknowledge that the previously mentioned metrics will likely show similar results compared to our metric.

Below, we clearly define our process for evaluating the robustness of clusters given our edit distance metric.

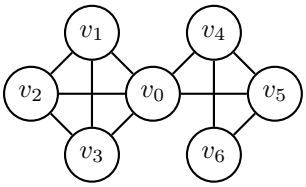
D. Experimental Setup

Given the data and a clustering algorithm, we perform the following:

1) Build the graph G
2) Run the (node embedding and) clustering algorithm on the graph to obtain clustering A – we chose node2vec followed by k-means clsutering, GRAPHsage followed by k-means clustering, and louvain with k values of $3 \dots 6$.
3) For each percent $p \in \{2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$, do the following $i = 0 \dots 20$ times:
   a) Remove p percent of the edges to to obtain G'_{p_i}
   b) Run the same clustering algorithm on the graph to obtain clustering B
   c) Build the cost matrix for A and B, a $n \times n$ matrix where $n = \max(|A|, |B|)$ and the cost of assigning cluster A_i to cluster B_j is then $|A_i \cap B_j|$. If $|A| \neq |B|$, pad the remaining entries with 0.
   d) Compute edit distance between A and B using cost matrix, to obtain $dist_{p_i}$
4) Average all $dist_{p_i}$'s for a given percentage p to see how consistent an algorithm is on a graph given that percent of the information.

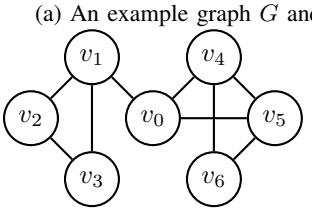
For example, Figure 1a shows a graph G and a potential initial clustering A_1 and A_2. This would be the result of step 1 and 2 from above. Then, following step 3, Figure 1b has removed some edges from G resulting in graph G' and thus a new clustering, B_1 and B_2. The cost matrix between A and B is then shown in Figure 1c. The maximum cost assignment would then be to assign A_1 with B_1 and A_2 with B_2 leading to a total cost of 6, meaning that 6 vertices did not change clustering. Given that there are 7 vertices in total, that means that the similarity score for this would be $\frac{6}{7}$.

To compute the maximum similarity between two clusters, we use the Python package `munkres` which implements the Hungarian algorithm, or the Kuhn-Munkres algorithm, to solve the assignment problem [18].



Cluster A_1	$\{v_0, v_1, v_2, v_3\}$
Cluster A_2	$\{v_4, v_5, v_6\}$

(a) An example graph G and a potential initial clustering.



Cluster B_1	$\{v_1, v_2, v_3\}$
Cluster B_2	$\{v_0, v_4, v_5, v_6\}$

(b) An example graph G' where some edges are removed from graph G and a potential clustering.

	A_1	A_2
B_1	3	0
B_2	1	3

(c) Cost Matrix associated with clustering A and B.

Fig. 1: An example of how the cost matrix would be built based on a graph and a given clustering.

V. Results

Using the methods described above, we acquired a series of scores that represented how stable the clustering algorithm was given a percentage of the total information.

We first did this on an Erdos-Renyi random graph [21] consisting of 200 nodes and 10,000 edges. Since there was only one edge type, we only tried node2vec embeddings with $k = 3..6$ and Louvain clustering. Figure 2 shows the average similarity scores as more information is retained.

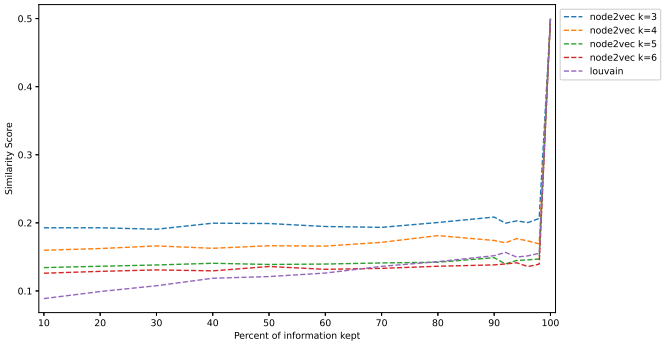


Fig. 2: Erdos-Renyi Random Graph with 200 nodes and 10,000 edges.

Next, the same methods were applied on graphs G_e and G_m generated with the synthetic data. Figures 3, 4, 5 and 6 show the comparison of stability scores for a given graph produced by the data and graph clustering methods. Each of those four graphs shows a different group of 200 patients from the dataset. In general, one can see that having more information leads to the clusters being more similar to the clusters produced with all the information. Across all four graphs, we can see that G_m, a graph where half the data is represented as edges and the other half is node attributes, using the node2vec node embedding algorithm and $k = 3$ for k-means shows to be generally the most stable.

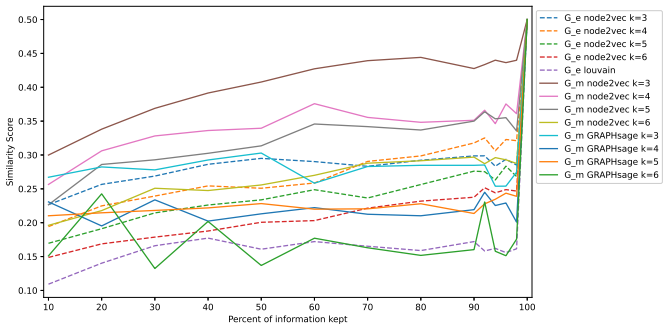


Fig. 3: Graph with 200 nodes. G_e has 16,289 edges and G_m has 11,871 edges.

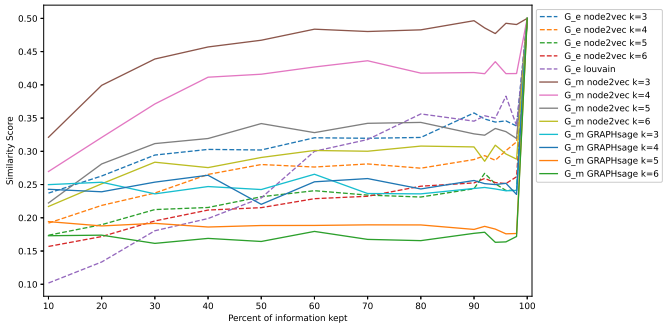


Fig. 4: Graph with 200 nodes. G_e has 16,273 edges and G_m has 12,002 edges.

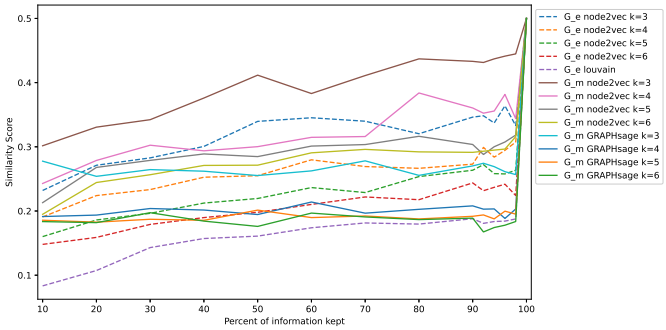


Fig. 5: Graph with 200 nodes. G_e has 16,619 edges and G_m has 12,375 edges.

As a final point of analysis, we will compare our proposed metric to existing methods such as the Adjusted Rand Index and the Jaccard Index. Similar to before, we treat the clusters generated with all the information as the "true" clusters and use the index to compare the clusters generated with less information. Figures 7 and 8 show the same graph from Figure 6 but compared using the Adjusted Rand Index and

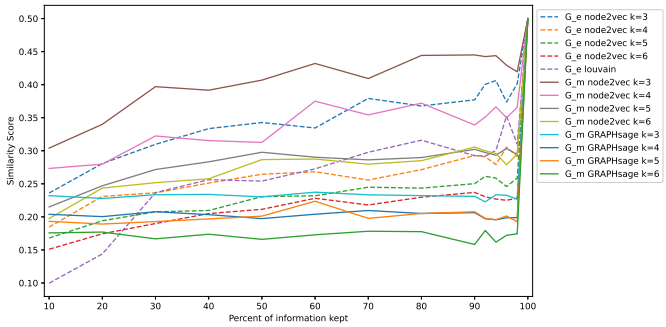


Fig. 6: Graph with 200 nodes. G_e has 16,083 edges and G_m has 12,033 edges.

Jaccard Index respectively. We can see how there are some similarities between the three similarity metrics, but there are also differences.

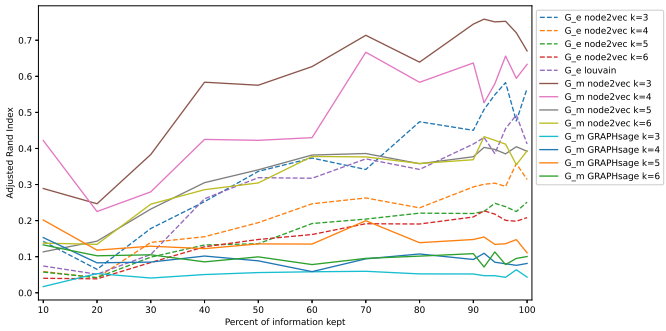


Fig. 7: Graph with 200 nodes. G_e has 16,083 edges and G_m has 12,033 edges. Evaluated using the Adjusted Rand Index.

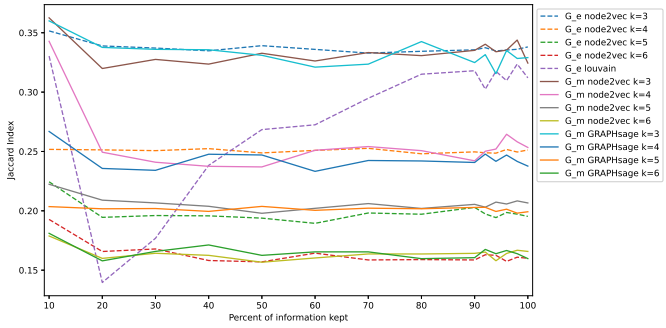


Fig. 8: Graph with 200 nodes. G_e has 16,083 edges and G_m has 12,033 edges. Evaluated using the Jaccard Index.

VI. Conclusions and Future Work

In this paper, we proposed a simple method of evaluating a clustering algorithm on data. This metric is especially useful for measuring how consistent a given clustering algorithm is on the data. We acknowledge that this metric is not entirely novel, nor groundbreaking, but the use of this metric is useful in the context of exploring chronic lower back pain patient clustering. This metric can be further refined and evaluated by testing on a broader spectrum of clustering algorithms and more graphs.

Acknowledgment

We extend our gratitude to members of the Cal Poly Bioinformatics Research Group for their invaluable support throughout this study. This research was supported by "UCSF Core Center for Patient-centric Mechanistic Phenotyping in Chronic Low Back Pain (UCSF REACH)" funded by NIH. Support was provided for the Computational Molecular Sciences Center by the Frost Fund at the Cal Poly Bailey College of Science and Math.

References

[1] Emmanuel Bäckryd, Elisabeth Persson, Annelie Inghilesi Larsson, Marcelo Rivano Fischer, and Björn Gerdle. Chronic pain patients can be classified into four groups: Clustering-based discriminant analysis of psychometric data from 4665 patients referred to a multidisciplinary pain centre (a SQRP study). *PLOS ONE*, 2018.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[3] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[4] CSIRO's Data61. Stellargraph machine learning library. https://github.com/stellargraph/stellargraph, 2018.

[5] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[6] Geert De Soete and J. Douglas Carroll. K-means clustering in a low-dimensional Euclidean space. In Edwin Diday, Yves Lechevallier, Martin Schader, Patrice Bertrand, and Bernard Burtschy, editors, *New Approaches in Classification and Data Analysis*, pages 212–219, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[7] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[8] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[9] Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.

[10] Aditya Grover and Jure Leskovec. Node2vec: Scalable Feature Learning for Networks. pages 855–864, 2016.

[11] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[13] Iris Ho, Paul Anderson, Jean Davidson, Jeffrey Lotz, and Theresa Migler. An evaluation of graph based approaches for clustering: a case study in chronic pain categories. In *Applied Network Science*. Springer Nature, 2024.

[14] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

[15] Björn Larsson, Björn Gerdle, Lars Bernfort, Lars-Åke Levin, and Elena Dragioti. Distinctive subgroups derived by cluster analysis based on pain and psychological symptoms in Swedish older adults with chronic pain - a population study (PainS65+). *BMC Geriatrics*, 2017.

[16] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[17] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[18] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[19] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[20] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[21] Erdos RENYI. On random graph. *Publicationes Mathematicate*, 6:290–297, 1959.

[22] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[23] Robert R Sokal and F James Rohlf. The comparison of dendrograms by objective methods. *Taxon*, pages 33–40, 1962.

[24] Scott D. Tagliaferri, Tim Wilkin, Maia Angelova, Bernadette M. Fitzgibbon, Patrick J. Owen, Clint T. Miller, and Daniel L. Belavy. Chronic back pain sub-grouped via psychosocial, brain and physical factors using machine learning. *Scientific Reports*, 2022.