Annales Univ. Sci. Budapest., Sect. Comp. 51 (2020) 131–146

GENOME CLASSIFICATION USING OVERLAP GRAPH CENTRALITIES

Péter Lehotay-Kéry and Attila Kiss

(Budapest, Hungary)

Dedicated to the 70th birthday of Professor Antal Járai

Communicated by András Benczúr

(Received December 16, 2019; accepted April 15, 2020)

Abstract. Genetics is a fast developing field and lot of its development relies on bioinformatics and solving computing problems. The genetic data are huge, for example the human reference genome is about 3 GB and for other species they can be even greater. It is not a trivial task to process them efficiently, recovering useful data for biological and medical sciences. Researchers have already developed different models and representations of genomes to provide deeper knowledge and explore hidden context in these data. Recent years a lot of publications have been made about how to represent genomes in graphs and examining the graph features of genomes like graph centrality.

The aim of this paper is comparing and examining the graph centrality of viral genomes that could help in the study of these data. We use a number of concepts of genetics and bioinformatics, mostly in meaningful context. Their exact individual definition would place too much burden on the article; the interested readers may turn to the references provided.

Key words and phrases: Betweenness centrality, closeness centrality, degree centrality, edit distance, eigenvector centrality, graphs, harmonic centrality, load centrality, string algorithms, overlap graph.

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

https://doi.org/10.71352/ac.51.131

1. Introduction

Deoxyribonucleic acid (DNA) is a complex molecule containing the genetic information, built up by nucleotides. Nucleotides are built up by three components: nucleobases, a sugar called deoxyribose and a phosphate group. There are four kind of nucleobases: adenine, cytosine, guanine and thymine.

A genome sequence contains the complete list of the nucleobases found in the chromosomes of an individual or a species. In bioinformatics, we store the genomes as strings, composed by the characters for the nucleobases, in case of a DNA genome 'A', 'C', 'G' and 'T'. Similarly, we can store RNA and protein genome sequence data too.

The common technology to sequence the DNA genome of an organism, can get only short reads (about the length of 50-250 base pairs) from the DNA. These reads are not in order, they can overlap and even repeats are allowed.

It is common to use graphs to assemble short reads of genomes. In [8] authors reviewed the most important types of genetic graphs, together with the algorithmic challenges and open issues related to their use.

Mainly, the following types of graphs are used to represent genomes: overlap graphs, string graphs, de Bruijn graphs, genome alignment graphs, tiled graphs, sequence graphs and variation graphs.

In this paper our aim was to analyze and compare different features of the graphs built from the genomes of viruses. We are going to focus mainly on whether we are able to classify these viral genomes into their virus families and genuses using the graph centrality values and the similarity of subsequences in the nodes with the maximal centrality values in their overlap graphs.

2. Related works

This section starts with the overview on the state of the art of different graph representations of genomes.

In the case of overlap graphs, in [18] authors proposed a de novo assembler software to process short reads produced by the Illumina sequencing platform. Based on a classical overlap graph representation and on the detection of potentially spurious reads, their software generates a set of accurate contigs of several kilobases (abbreviated by 'kb') that cover most of the bacterial genome.

Also, in [17] the Omega (overlap-graph meta genome assembler) was developed by the authors for assembling and scaffolding Illumina sequencing data of microbial communities. In comparison with three de Bruijn graph assemblers (SOAPdenovo, IDBA-UD and MetaVelvet), Omega provided comparable overall performance on a HiSeq 100-bp dataset and superior performance on a MiSeq 300-bp dataset.

For example of using de Bruijn graphs, in [27] authors developed a new set of algorithms, collectively called "Velvet" to manipulate de Bruijn graphs for genomic sequence assembly. When applied to real Solexa data sets, Velvet generated contigs of about 8 kb in a prokaryote and 2 kb in a mammalian BAC.

Authors in [22] present the concept and formalism of the string graph, which represents all that is inferable about a DNA sequence from a collection of shotgun sequencing reads collected from it. They also demonstrate that the decomposition of reads into k-mers employed in the de Bruijn graph approach described earlier is not essential.

In [25] authors introduce some tile assembly models (aTAM, kTAM, 2HAM) based on tile graphs. They also discuss and define a wide array of more recently developed models and discuss their various trade-offs in comparison to the previous models and to each other.

Authors of [7] provide an initial, theoretical solution to the challenge of de novo assembly from whole-genome shotgun microreads. Assemblies are presented in sequence graph that retains intrinsic ambiguities such as those arising from polymorphism, thereby providing information that has been absent from previous genome assemblies.

In [14] authors present vg, a toolkit of computational methods for creating, manipulating, and utilizing variation graph structures as references at the scale of the human genome. They found that using variation graphs as references for DNA sequencing practical at gigabase scale, or at the topological complexity of de novo assemblies.

There are some works discussing how else graphs can be used for genomic calculations. Now, we are mostly interested about graph and network centrality.

As first example, in [26] authors propose a method to integrate different breast cancer gene signatures by using graph centrality in a context-constrained protein interaction network (PIN). The genes which are well-known breast cancer genes, such as TP53 and BRCA1, are ranked extremely high in their results.

In [20] authors used the topological centrality in protein networks of complex trait genes for implications in genetics, personal genomics, and therapy.

Furthermore, in [21] authors identified potential drug targets of Mycobacterium tuberculosis H37Rv through systematically integrated comparative genome and network centrality analysis.

3. Background

This section is about the theoretical background of the algorithms we used. Here we define the overlap graph of strings, the graph centrality measures that we calculated for genomes in graph representation and a method to calculate the similarity of strings.

3.1. Overlap graph of strings

Definition 3.1 (see [2])**.** Let s and t be strings over an alphabet. If there exists a partition of s and t with the properties:

$$s = xy,\ t = yz,\ x \neq e,\ z \neq e$$

where the length of y is maximal and e is empty string, then y is the overlap of s and t, denoted by $Ov(s,t)$ and $ov(s,t)$ is the length of $Ov(s,t)$.

Definition 3.2 (see [2])**.** Let $S = (s_1, ..., s_n)$ be a set of strings. The overlap graph of S is the complete edge-weighted directed graph:

$$G_{ov}(S) = (V, E, c),$$

where

$$V = S,\ E = V^2,\ c : E \to \mathbb{N}$$

with

$$\forall s_i, s_j \in V : c(s_i, s_j) = ov(s_i, s_j).$$

We can use a threshold value indicating a minimal number of overlapping characters betweenness two words. When building the graph, only those overlaps will be considered that satisfy this threshold value. Figure 1 shows an example overlap graph, of strings 'CCT', 'CTT', 'TGC', 'TGG', 'GAT', 'ATT' with threshold of 1.

3.2. Graph centrality measures

Definition 3.3. The degree centrality for a node is the fraction of nodes it is connected to.

Definition 3.4 (see [4])**.** Eigenvector centrality computes the centrality for a node based on the centrality of its neighbors. The eigenvector centrality for node i is the i-th element of the eigenvector e defined by the following equation:

$$Re = le,$$

where R is the adjacency matrix of the graph G with eigenvalue l. There is a unique solution e, all of whose entries are positive, if l is the largest eigenvalue of the adjacency matrix R.

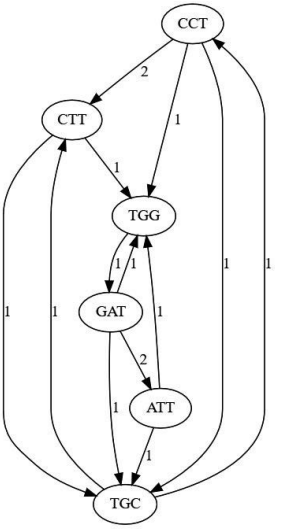

Figure 1. Example overlap graph

Definition 3.5 (see [13])**.** Closeness centrality of a node u is the reciprocal of the average shortest path distance to u over all $n-1$ reachable nodes.

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)},$$

where $d(v,u)$ is the shortest-path distance between v and u, and n is the number of nodes that can reach u.

Definition 3.6 (see [12][5])**.** betweenness centrality of a node v is the sum of the fraction of all-pairs shortest paths that pass through v:

$$c_b(v) = \sum_{s,t \in V} \frac{o(s,t|v)}{o(s,t)}$$

where V is the set of nodes, $o(s,t)$ is the number of shortest (s,t)-paths, and $o(s,t|v)$ is the number of those paths passing through some node v other than s,t. It is also called shortest-path betweenness centrality.

Definition 3.7 (see [5][6])**.** betweenness centrality of an edge e is the sum of the fraction of all-pairs shortest paths that pass through e:

$$c_B(e) = \sum_{s,t \in V} \frac{o(s,t|e)}{o(s,t)},$$

where V is the set of nodes, $o(s,t)$ is the number of shortest (s,t)-paths, and $o(s,t|e)$ is the number of those paths passing through edge e.

Definition 3.8 (see [15][24])**.** The load centrality of a node is the fraction of all shortest paths that pass through that node.

Definition 3.9 (see [3])**.** Harmonic centrality of a node x is the sum of the reciprocal of the shortest path distances from all other nodes to x:

$$C(x) = \sum_{y \neq x} \frac{1}{d(y,x)}$$

where $d(y,x)$ is the shortest-path distance between y and x.

Table 1 summarizes the centrality statistics of the example graph presented on Figure 1.

Centrality measure	Maximal value	Nodes, edges with maximal value
Degree centrality	1.2	TGC
Eigenvector centrality	0.6	TGC, TGG
Closeness centrality	0.8	TGC, TGG
Betweenness centrality	0.35	TGC, TGG, GAT
Edge betweenness centrality	0.4	TGG-GAT
Load centrality	0.35	TGC, TGG, GAT
Harmonic centrality	4.5	TGC,TGG

Table 1. Centrality values of example graph

3.3. Similarity of strings

Definition 3.10 (see [2])**.** Let $s = s_1...s_m$ and $t = t_1...t_n$ be two strings over an alphabet E. Let $-$ be a gap symbol, E' another alphabet, h homomorphism:

$$- \notin E,\ E' = E \cup \{-\},\ h : (E')^* \to E^*,\ h(a) = a,\ h(-) = e,$$

where e is empty string.

An alignment of s and t is a pair of strings of length l over alphabet E':

$$(s', t') : l \geq max\{m, n\}$$

such that the following conditions hold:

$$|s'| = |t'| \geq max\{|s|, |t|\},\ h(s') = s, h(t') = t$$

and there is no position containing a gap symbol in s' as well as in t', i.e.,

$$\forall i \in 1, .., l : (s'_i \neq - \vee t'_i \neq -)$$

Definition 3.11 (see [2]). Let s and t be two strings over an alphabet E.

$$\forall a, b \in E : p(a, b) \in \mathbb{Q}, g \in \mathbb{Q}.$$

The score d of an alignment (s', t') of length l is first defined column-wise:

$$\forall x, y \in E : d(x, y) = p(x, y).$$

Furthermore, let $d(-, y) = d(x, -) = g$. The score of an alignment is then defined as the sum of the values over all columns, i.e.,

$$d(s', t') = \sum_{i=1}^{l} d(s'_i, t'_i).$$

For an alignment score d, we furthermore define an optimization goal

$$goal_d \in \{min, max\}$$

Definition 3.12 (see [2]). Let s and t be two strings, and let d be an alignment scoring function. The similarity $sim_d(s, t)$ of s and t according to d is the score of an optimal alignment of s and t, i.e.,

$$sim_d(s, t) = goal_d\{d(s', t')|(s', t')\}$$

where (s', t') is an alignment of s and t. If the alignment scoring function is clear from the context, we also write sim instead of sim_d.

We chose the parameters as $p(a, a) = 1$, $p(a, b) = -1$, if a not equals b and $g = -2$. So the similarity of two subsequence will be the maximum score of an alignment.

In the above calculations we defined linear penalty for gaps. For the measurements we calculated affine penalty.[1] An affine gap penalty is written as $a + b(L - 1)$, where L is the length of the gap, a is a constant called the gap opening penalty, and b is a constant called the gap extension penalty. We defined a as -10 and b as -2.

In Table 2 we present the similarity values of some example strings.

4. Centrality results

The genomes processed in this article are from the NCBI [23] and the Ensembl Genome databases [11]. We collected viruses from two different virus families and four different genuses, as presented on Table 3, to search for correlation between these classifications of viruses based on the centralities.

	CCT	CTT	TGC	TGG
CCT	-	2	-1	-2
CTT	2	-	-1	-2
TGC	-1	-1	-	1
TGG	-2	-2	1	-

Table 2. Similarity of example strings

Virus	Genus	Family
Avian retrovirus	Alpharetrovirus	Retrovirus
Feline leukemia	Gammaretrovirus	Retrovirus
Hepatitis C	Hepacivirus	Flaviviridae
Murine leukemia	Gammaretrovirus	Retrovirus
Yellow fever	Flavivirus	Flaviviridae
Zika	Flavivirus	Flaviviridae

Table 3. Examined viruses

As the result of our research, first we report the different centrality values of the viruses in our dataset, presenting only the node with the maximal centrality and its centrality value.

For the graph centrality calculations we used the Netrowkx python module [16].

To measure the centrality values of viruses, we defined the nodes as length of 10 subsequences and defined the minimum overlapping value as 7.

4.1. Retrovirus

Tables 4, 5 and 6 show the centrality value results and the subsequences of the nodes and edges with maximal centrality values of the retroviruses.

Later we will see that their maximal eigenvector, between, edge between and load centrality values are usually higher then what the viruses have in the family of flaviviridaes.

For example, in case of betweenness centralities this means that in these graphs more paths are passing through some nodes in average, then through others. As a consequence, in our case, the subsequences in these nodes are covered by more "overlap paths" than other subsequences in average. So these subsequences are more likely that could be used to determine similarity between whole genomes.

Centrality measure	Maximal value	Nodes, edges with maximal value
Degree centrality	0.004	ATGGCAGAAG
Eigenvector centrality	0.0776	TCATCCTCAG
Closeness centrality	0.093	AGGGAGGTTC
betweenness centrality	0.0269	AGGGAGGGGG
Edge betweenness centrality	0.0175	CAGTTGGCTA-TTGGCTACAG
Load centrality	0.0272	AGGGAGGGGG
Harmonic centrality	405.1009	AGGGAGGTTC

Table 4. Centrality values of Avian retrovirus

Centrality measure	Maximal value	Nodes, edges with maximal value
Degree centrality	0.0027	ACCGACCCCA
Eigenvector centrality	0.0785	GAAGAAAGAG
Closeness centrality	0.1238	CCTCTTGCTG
betweenness centrality	0.01	AGGAAAAACT
Edge betweenness centrality	0.005	AAAAACTCGA-AACTCGACCA
Load centrality	0.01	AGGAAAAACT
Harmonic centrality	1062.773	CCTCTTGCTG

Table 5. Centrality values of feline leukemia

Centrality measure	Maximal value	Nodes, edges with maximal value
Degree centrality	0.0036	CCTCCTCTGA, GGTGGAGAAG
Eigenvector centrality	0.0574	CACCAAGGCC
Closeness centrality	0.1196	CACCAAGGGC
betweenness centrality	0.0145	TGTCACCAAG
Edge betweenness centrality	0.0092	CCTGGCCACC-GGCCACCAAG
Load centrality	0.0146	TGTCACCAAG
Harmonic centrality	702.8147	CTGTGTTGTC, CTGTGTTGTG

Table 6. Centrality values of Murine leukemia

4.2. Flaviviridae

In tables 7, 8 and 9 we can see the centrality value results and the subsequences of the nodes and edges with maximal centrality values of the flaviviridaes.

As we discussed in the previous subsection, their maximal eigenvector, between, edge between and load centrality values are usually lower than the corresponding values in the family of retroviruses.

In contrary to the conclusion we made in section 4.1, in case of betweenness centralities for example, the subsequences in the nodes with the maximal centrality values could less likely be used to determine the similarity between whole genomes.

Centrality measure	Maximal value	Nodes, edges with maximal value
Degree centrality	0.0022	TCCTGGCGGG
Eigenvector centrality	0.0539	CTGCTCCTTC
Closeness centrality	0.1283	CCACTGGGGC
betweenness centrality	0.0089	CCACTGGCGG
Edge betweenness centrality	0.0044	CTCCTTCACT-CTTCACTACC
Load centrality	0.0089	CCACTGGCGG
Harmonic centrality	1301.339	GGGGGAGAAT

Table 7. Centrality values of Hepatitis C

Centrality measure	Maximal value	Nodes, edges with maximal value
Degree centrality	0.0021	TTTGGGAAAG, TGAGGAAAGT
Eigenvector centrality	0.0468	AATGACAACC
Closeness centrality	0.15	GGAAGAATGG
betweenness centrality	0.0067	GGGAAAGGAA
Edge betweenness centrality	0.0019	TGCCATGGGA-CATGGGAAAG
Load centrality	0.0067	GGGAAAGGAA
Harmonic centrality	1720.605	GGAAGAATGG

Table 8. Centrality values of yellow fever

Centrality measure	Maximal value	Nodes, edges with maximal value
Degree centrality	0.0027	AGAGAGGAGA
Eigenvector centrality	0.059	AGAATGGATG
Closeness centrality	0.1558	AGAGAGGATA
betweenness centrality	0.0059	AGAGAGGAAG
Edge betweenness centrality	0.0022	AGAGCATTCA-GCATTCACCA
Load centrality	0.006	AGAGAGGAAG
Harmonic centrality	1790.1804	AGAGAGGATA

Table 9. Centrality values of zika virus

5. Correlations

In this section we present the patterns and correlations we found in the calculated results. First the connection of centrality values between the viruses from different or same families and genuses, then the similarity of the subsequences from the nodes with maximal centrality values from different families.

5.1. Correlations in centrality values

On Figure 2 we can inspect the correlations of the maximal centrality values of the viral genomes. We can see that viruses from the same family has closer maximal centrality values at closeness centrality, betweenness centrality, edge betweenness centrality and load centrality. However the maximal values for eigenvector centrality is more mixed.

Furthermore we can see that maximal centrality values of viruses from different genuses in the same family also show larger difference from the maximal centrality values of viruses in the same genus in the same family in the case of the maximal values of closeness centrality, betweenness centrality, edge betweenness centrality and load centrality.

5.2. Correlations in similarity

Next we discuss the similarity of subsequences that are on the nodes with maximal centrality value. We measured the similarity of nodes with maximal centrality for two viruses from each of the two different families.

On Table 10 we can see that with degree centrality, viruses from the same family has more similar subsequences at the nodes with maximal degree centrality values than viruses from different families.

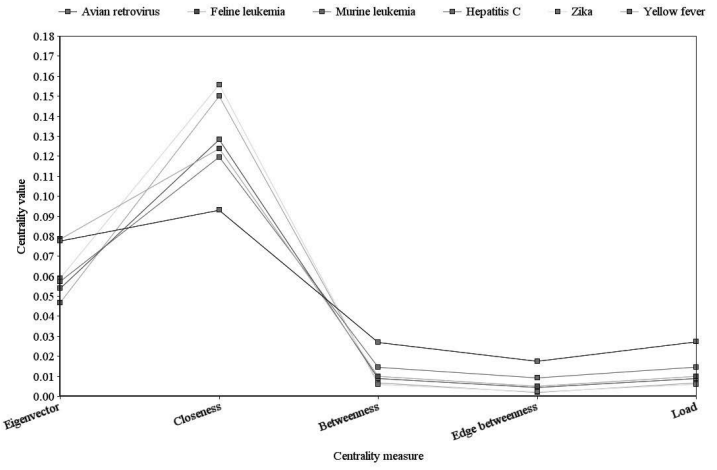

Figure 2. Correlation of the maximal centrality values

	feline	murine	yellow fever	zika
feline	-	-1	-4	-2
murine	-1	-	-4	-6
yellow fever	-4	-4	-	0
zika	-2	-6	0	-

Table 10. Similarity of nodes with maximal degree centrality

Table 11 shows similar correlations with harmonic centrality as in the case of degree centrality.

	feline	murine	yellow fever	zika
feline	-	-2	-6	-5
murine	-2	-	-7	-5
yellow fever	-6	-7	-	0
zika	-5	-5	0	-

Table 11. Similarity of nodes with maximal harmonic centrality

Looking at Table 12 we can make an interesting observation: murine leukemia got the same similarity values for all the other examined viruses. Considering that yellow fever and zika is still the most similar, we can only conclude that viruses from the same family have more or equally similar subsequences at the nodes with the maximum closeness centrality value.

	feline	murine	yellow fever	zika
feline	-	-3	-6	-5
murine	-3	-	-3	-3
yellow fever	-6	-3	-	0
zika	-5	-3	0	-

Table 12. Similarity of nodes with maximal closeness centrality

Discussing Figure 2 we could not conclude the existence of correlation of various eigenvector centrality values and families of viruses. Consequently, the similarity values presented in Table 13 also do not point towards correlation.

	feline	murine	yellow fever	zika
feline	-	-1	0	0
murine	-1	-	1	-4
yellow fever	0	1	-	-2
zika	0	-4	-2	-

Table 13. Similarity of nodes with maximal eigenvector centrality

6. Conclusion

We have analyzed the centrality values of overlap graph representations for virus genomes. Our research shows that viruses from different families and also from different genuses probably show difference in their maximal centrality values but viruses from the same families and also from the same genuses can show similar maximal centrality values.

We have also analyzed the similarity of the subsequences in the nodes with the maximal centrality values and we found that viruses from the same family probably show more similarity for these subsequences than viruses from different families using degree and harmonic centrality.

7. Future works

We are working on a website, where the centrality values of all nodes for all genome graphs will be published, with multiple parameter settings, so that researchers can access them and can search for patterns. The site will be available at `http://bioinformatics.elte.hu/`

Furthermore, it could be interesting to consider not only a single node with the maximal centrality value per genome, but consider multiple nodes with high centrality values for each genome, when analyzing the similarity of genomes.

We should also check our results against greater datasets and with automated classification.

References

[1] **Altschul, S.F.,** Generalized affine gap costs for protein sequence alignment, *Proteins: Structure, Function, and Bioinformatics*, 32.1, Wiley Online Library, 1998, 88–96.

[2] **Bockenhauer, H-J and D. Bongartz,** *Algorithmic Aspects of Bioinformatics*, Springer, 2007

[3] **Boldi, P. and S. Vigna,** Axioms for centrality, *Internet Mathematics*, 10.3-4, Taylor & Francis, 2014, 222–262.

[4] **Bonacich, Ph.,** Power and centrality: A family of measures, *American Journal of Sociology*, 92.5, University of Chicago Press, 1987, 1170–1182.

[5] **Brandes, U.,** A faster algorithm for betweenness centrality, *Journal of Mathematical Sociology*, 25.2, Taylor & Francis, 2001, 163–177.

[6] **Brandes, U.,** On variants of shortest-path betweenness centrality and their generic computation, *Social Networks*, 30.2, Elsevier, 2008, 136–145.

[7] **Butler, J., I. MacCallum, M. Kleber, I. A. Shlyakhter, M. K. Belmonte, E. S. Lander, C. Nusbaum and D. B. Jaffe,** ALLPATHS: de novo assembly of whole-genome shotgun microreads, *Genome Research*, 19.5, Cold Spring Harbor Lab, 2008, 810–820.

[8] **Carletti, V., P. Foggia, E. Garrison, L. Greco, P. Ritrovato and M. Vento,** Graph-Based Representations for Supporting Genome Data Analysis and Visualization: Opportunities and Challenges, *International Workshop on Graph-Based Representations in Pattern Recognition*, Springer, 2019, 237–246.

[9] **Chang, Y. J., C. C. Chen, C. L. Chen and J. M. Ho,** A de novo next generation genomic sequence assembler based on string graph and MapReduce cloud computing framework, *BMC Genomics*, 13.7, BioMed Central, 2012, S28.

[10] **Compeau, P. E., P. A. Pevzner and G. Tesler,** How to apply de Bruijn graphs to genome assembly, *Nature Biotechnologys*, 29.11, Nature Publishing Group, 2011, 987.

[11] **Ensembl Genomes**, `http://ensemblgenomes.org/`, Accessed 15 Nov 2019

[12] **Freeman, L. C.,** A set of measures of centrality based on betweenness, *Sociometry*, JSTOR, 1977, 35–41.

[13] **Freeman, L. C.,** Centrality in social networks conceptual clarification. *Social Networks*, 1.3, North-Holland, 1978, 215–239.

[14] **Garrison, E., J. Sirén, A. M. Novak, G. Hickey, J. M. Eizenga, E. T. Dawson, W. Jones, S. Garg, C. Markello, M. F. Lin, B. Paten and R. Durbin,** Variation graph toolkit improves read mapping by representing genetic variation in the reference, *Nature Biotechnology* Nature Publishing Group, 2018.

[15] **Goh, K. I., B. Kahng and D. Kim,** Universal behavior of load distribution in scale-free networks, *Physical Review Letters*, 87.27, APS, 2001, 278701

[16] **Hagberg, A., P. Swart, and S. D. Chult,** Exploring network structure, dynamics, and function using NetworkX, *Los Alamos National Lab.(LANL)* Los Alamos, NM (United States), 2008.

[17] **Haider, B., T. H. Ahn, B. Bushnell, J. Chai, A. Copeland and C. Pan,** Omega: an overlap-graph de novo assembler for metagenomics, *Bioinformatics*, 30.19, Oxford University Press, 2014, 2717–2722.

[18] **Hernandez, D., P. François, L. Farinelli, M. Østerås and J. Schrenzel,** De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer, *Genome Research*, 18.5, Cold Spring Harbor Lab, 2008, 802–809.

[19] **Katz, L.,** A new status index derived from sociometric analysis, *Psychometrika*, 18.1, Springer, 1953, 39–43.

[20] **Lee, Y., H. Li, J. Li, E. Rebman, I. Achour, K. E. Regan, E. R. Gamazon, J. L. Chen, X. H. Yang, N. J. Cox and Y. A. Lussier,** Network models of genome-wide association studies uncover the topological centrality of protein interactions in complex diseases, *Journal of the American Medical Informatics Association*, 20.4, BMJ Publishing Group, 2013, 619–629.

[21] **Melak, T. and S. Gakkhar,** Comparative genome and network centrality analysis to identify drug targets of mycobacterium tuberculosis h37rv, *BioMed Research International*, 2015, Hindawi, 2015.

[22] **Myers, E. W.,** The fragment assembly string graph, *Bioinformatics*, 21.3, Oxford University Press, 2005, 1179–1185.

[23] **NCBI National Center for Biotechnology Information**, `https://www.ncbi.nlm.nih.gov/`, Accessed 15 Nov 2019

[24] **Newman, M. E. J.,** Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality, *Physical Review E*, 64.1, APS, 2001, 016132.

[25] **Patitz, M. J.,** An introduction to tile-based self-assembly and a survey of recent results, *Natural Computing*, 13.2, Springer, 2014, 195–224.

[26] **Wang, J., G. Chen, M. Li and Y. Pan,** Integration of breast cancer gene signatures based on graph centrality, *BMC Sstems Biology*, 5.2, BioMed Central, 2011.

[27] **Zerbino, D. R. and E. Birney,** Velvet: algorithms for de novo short read assembly using de Bruijn graphs, *Genome Research*, 18.5, Cold Spring Harbor Lab, 2008, 821–829.

P. Lehotay-Kéry and A. Kiss
Department of Information Systems
Faculty of Informatics
Eötvös Loránd University
H-1117 Budapest, Pázmány Péter sétány 1/C
Hungary
lkp@caesar.elte.hu
kiss@inf.elte.hu