ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Social Network Analysis

Title: Music as a network

Ονοματεπώνυμο	Αριθμός Μητρώου	email
Θοδωρής Τσίγκανος	3160180	teotsi@gmail.com

Ακαδημαϊκό έτος 2019-2020

I. Introduction

Music has always picked the interest of man. An absurd amount of time and resources have been spent through the ages, in many artists' quest to compose the perfect musical piece. Any musical piece can be considered as a unique permutation of the elements in a set of notes (for the sake of simplicity beat patterns won't be examined in this report). For example, (B, B, B, C#, D, D, D, C#, C#, B) is the main riff of "Satisfaction", by The Rolling Stones. A dilemma that's always pestering every artist is how can one determine whether or not a certain permutation "makes sense"? Of course, over the years certain music patterns have become more prevalent. The purpose of this report is to explore a different composing method. Instead of following patterns and scales, can one study the work of an artist and find out how they compose music? Would the use of a certain note set be beneficial in regards to mainstream success? Music has always been deeply subjective, but is it possible to extract a pattern or set that "objectively" works?

II. Resources

The dataset used for the study was extracted from https://kithara.to/, a Greek website where users can submit musical compositions and the corresponding chords for each composition. It is mostly focused on guitar-based pieces but not exclusively. With almost 40,000 songs on the website's database, I decided to pick songs from a single artist. This was done for two main reasons: 1) Computational restrictions, 2) to minimize the range of genres covered by the dataset. That is what led me to pick Xatzifragketa as the artist. With over 100 unique songs submitted, a sufficient sample was available.

III. Tools used

The crawler/scraper used for the extraction of the chords was developed using Python 3.7. It is available in this GitHub repository, and should be included with the report as well. It makes use of the BeautifulSoup module that allows easy access to website elements (which was mandatory since kithara.to doesn't provide an API to access songs). A high level explanation of the code is the following: First, it retrieves Xatzifragketa's artist page, and then extracts the unique ID of each song available. There are quite a few duplicates submitted, but since user ratings are available, we keep the ID of the most popular version. Then, we go over the list of songs, retrieve its lyrics and chords from https://kithara.to/ssbd.php?id=<song_id>, extract the chords, translate them where it's needed (i.e. Eb is the same note as D#), normalize them, and add them as new nodes, while making sure to record the transitions between each chord (those are the edges of our graph). After that's done, all we have to do is create a "nodes.csv" file, containing each chord and its name, and an "edges.csv" file, that contains all the transitions that take place. Those files are imported in Gephi, which is a tool that allows us to visualize graphs, as well as draw conclusions by using integrated metrics and resources. As the previous statement implies, all graphs, pictures, and screenshots presented in this report were produced by Gephi.

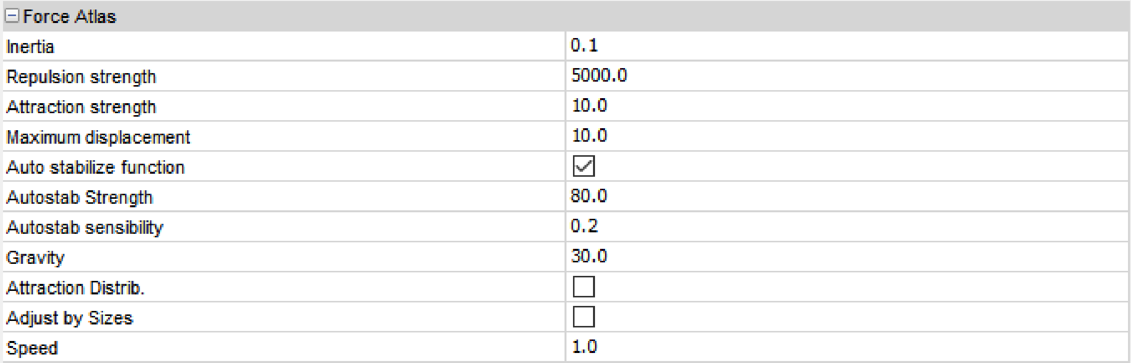
IV. Dataset

The dataset, as mentioned above, consists of two files, a "nodes.csv" file, and an "edges.csv" one. Both files should be bundled with this report, but they can also be generated using the scraper that's also included. All that is needed for the scraper to work properly, is to create a file called "cookie.txt", and add there the user session cookie from kithara.to. Otherwise, requests to the site will be blocked after a few songs.
Over a total of 103 distinct songs, 45 unique music chords were used. These are the nodes of the graph. Between these nodes, 3664 chord transitions were performed. Note that repeated patterns are not recorded on kithara.to (i.e. since each chorus, verse and bridge sequence is defined once), so the actual number of chords transitions is much higher.

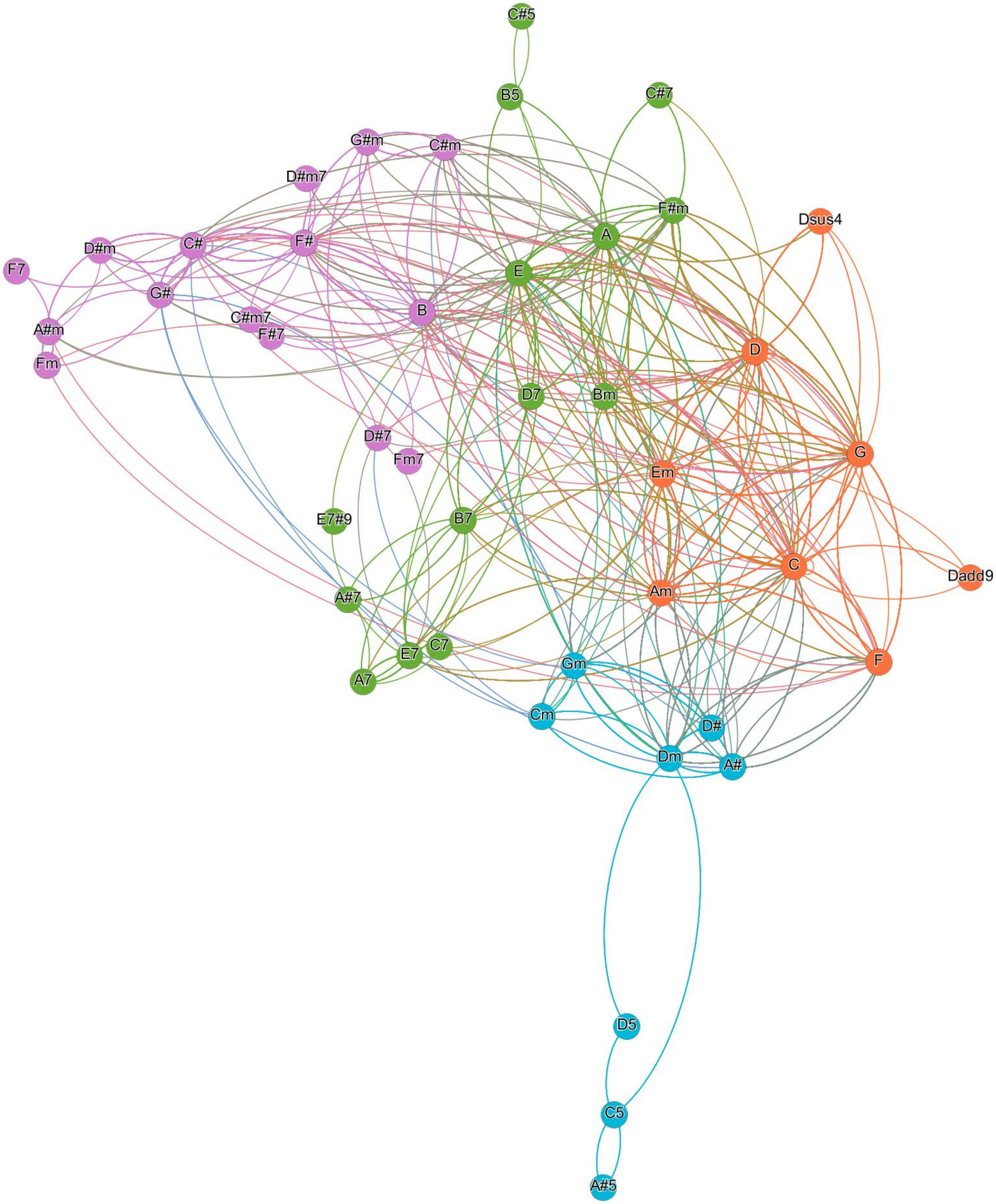
It is also crucial to keep in mind that the first chord of a song does not have a chord preceding it. That means that previously examined songs have no effect on newer ones.

V. Visualization

As mentioned above, Gephi handled all the visualization duties for this project. On the next page, a basic visualization (that is explained thoroughly in the report) is available. The layout (or rather the position of each node in the plane was generated using Gephi's Force Atlas layout, with a repulsion strength of 5000.



⊟ Force Atlas	
Inertia	0.1
Repulsion strength	5000.0
Attraction strength	10.0
Maximum displacement	10.0
Auto stabilize function	☑
Autostab Strength	80.0
Autostab sensibility	0.2
Gravity	30.0
Attraction Distrib.	□
Adjust by Sizes	□
Speed	1.0

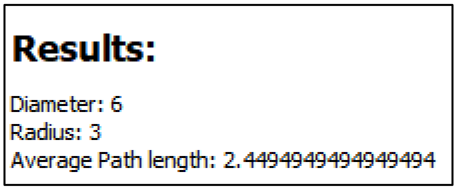

C#5
B5
C#7
G#m
C#m
D#m7
F#m
Dsus4
A
D#m
C#
F#
E
F7
G#
B
C#m7
A#m
F#7
Fm
D
D7
Bm
D#7
Fm7
G
Em
E7#9
B7
C
Dadd9
A#7
Am
C7
E7
Gm
F
A7
Cm
D#
Dm
A#
D5
C5
A#5

VI. Analyzing the Network

a) Topology

Formally, a graph G is defined by two sets, V and E. In this case, V contains chords and (variations of them) that are labeled following the English notation for notes and chords (opposed to Romance or Solfege, i.e. Do, Re, Mi…). Overall size of V is: |V|=45.
Set E contains the edges of the graph, which in this case are the chords transitions that take place during each song. Since the direction of each transition in a song is strictly defined by the artist, every edge on the graph is directed. Overall size of E is: |E|=3664.

Calculating the average path length on this type of graph may seem irrelevant or meaningless at first glance, but it actually allows us to understand how certain harmonic scales work when applied to an actual musical piece. There are always going to be limitations in accessing certain nodes, depending on the note on which the traversal begins. Furthermore, the diameter of the graph (aka the longest shortest path) sheds light into the necessary transitions to use two notes that normally are not combined. Do note that certain notes included in the graph (such as Dsus4 or Dsus2) have the same root note as D and Dm. That means that usually (but not always) they are used in alternating patterns with D and Dm, which minimizes their distance.



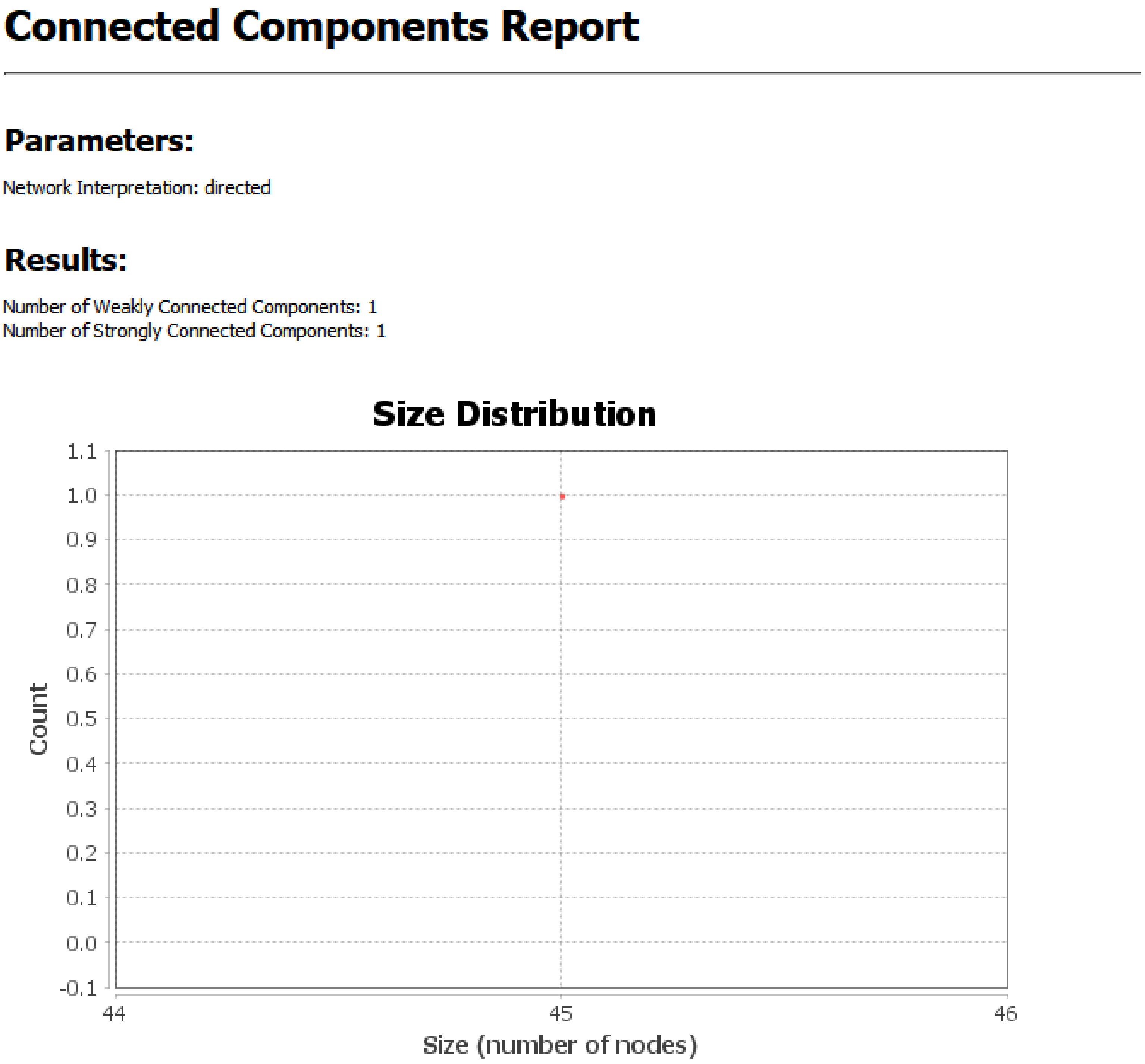
```
Results:

Diameter: 6
Radius: 3
Average Path length: 2.4494949494949494
```

As expected, the average path length is significantly smaller than the diameter of the graph, coming in at 2.45, while the diameter of the graph is 6.

b) Component measures

In this section we are going to explore the number of components in the graph. A component is essentially a subgraph whose vertices are connected solely to other vertices in the same subgraph. For example, if we consider the graph defined by "Satisfaction" once again, (B, B, B, C#, D, D, D, C#, C#, B) it is easy to observe that there are only transitions between chords B, C#, and D. In a hypothetical network where is coexists only with the song "Bulls on Parade" by Rage Against the Machine, which has a riff that consists solely of octaves of F, then there are two components, since there are no common vertices between the two subgraphs defined by each riff.
Of course, the graph that this report is focused on consists of a multitude of songs. Since nodes are not distinct for each song, it is highly likely that the graph consists of just one component, regardless of whether it's treated as a directional or undirected graph. After all, studying how chords can be combined is partially the purpose of this analysis, and

music theory has shown how with simple scales any two notes can coexist in a pattern. So, it is mostly a question of whether or not those patterns appear in Xatzifragketa songs.



Connected Components Report

Parameters:

Network Interpretation: directed

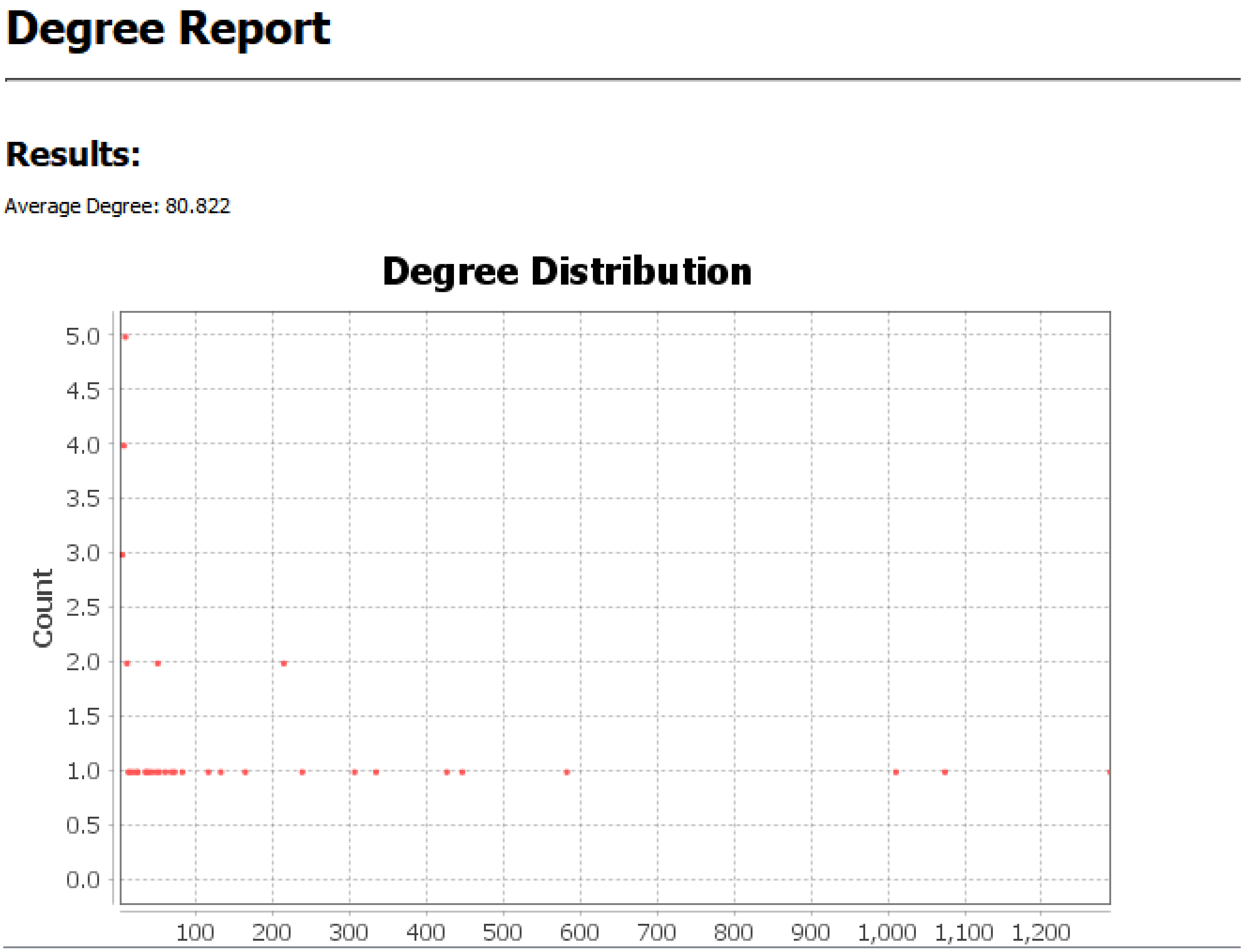
Results:

Number of Weakly Connected Components: 1
Number of Strongly Connected Components: 1

Sure enough, they do! As stated by the Gephi report, there's a single (Giant) component in the graph, consisting of 45 nodes. While trivial, this is evidence of the numerous applications of music theory in actual compositions. Any chord can coexist in a pattern with any other chord. Giant components can also be identified using Gephi's filters.

c) Degree measures

Next, various degree measures are presented. The degree of a node is defined as the number of edges that are adjacent to the node. In-Degree and Out-Degree define, as one can probably guess without much effort, the number of edges that have said node as target and source, respectively. This is undoubtedly an important measure in the graph that is being examined, since it allows to examine how certain chords may be considered for a certain position in a chord pattern.

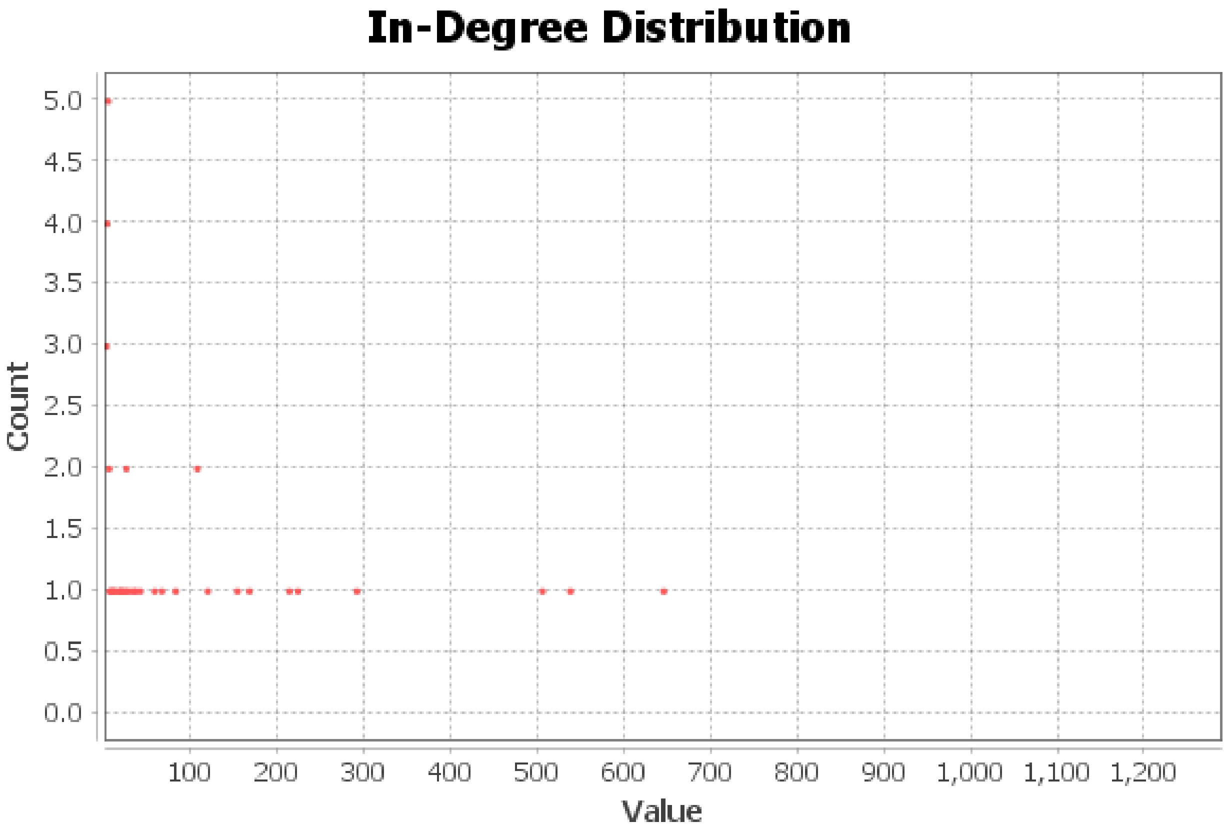


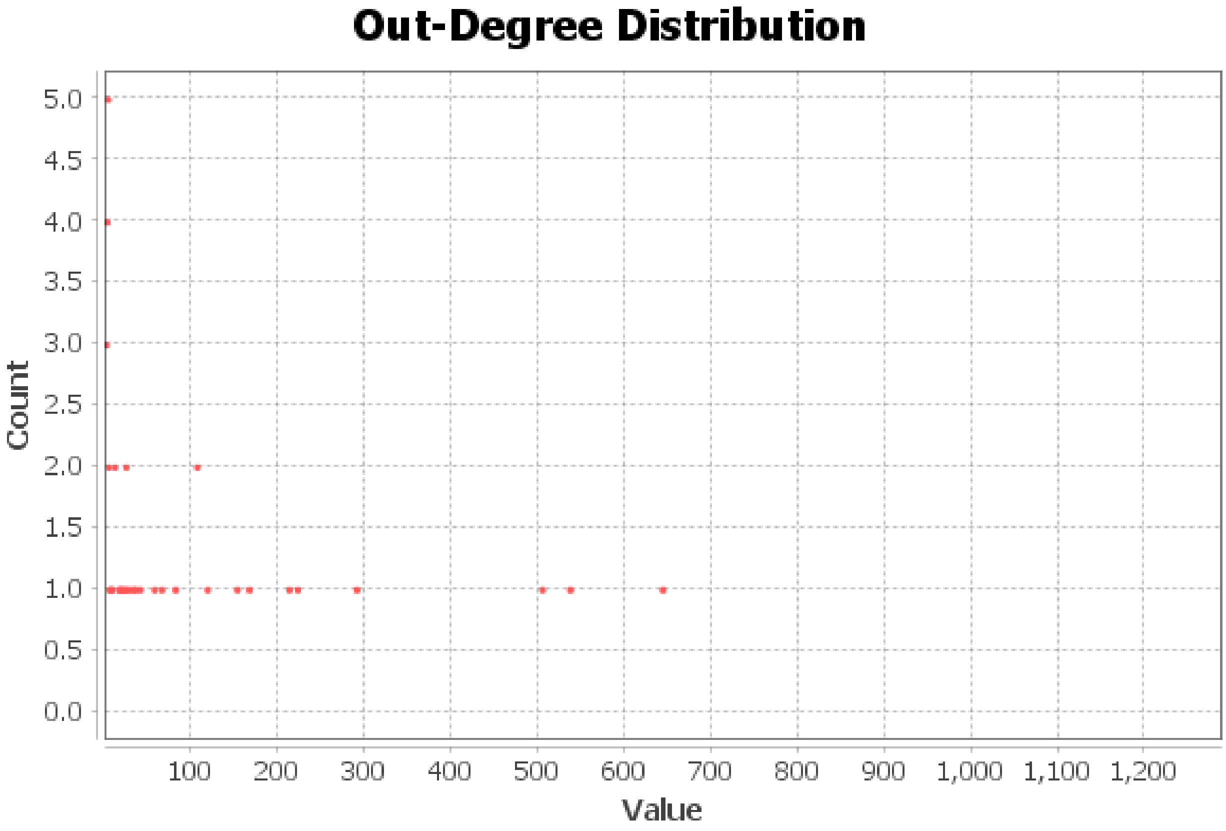
Degree Report

Results:

Average Degree: 80.822

According to Gephi, the average Degree of the nodes in the graph is 80.8.


In-Degree Distribution
Count
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
100
200
300
400
500
600
700
800
900
1,000
1,100
1,200
Value

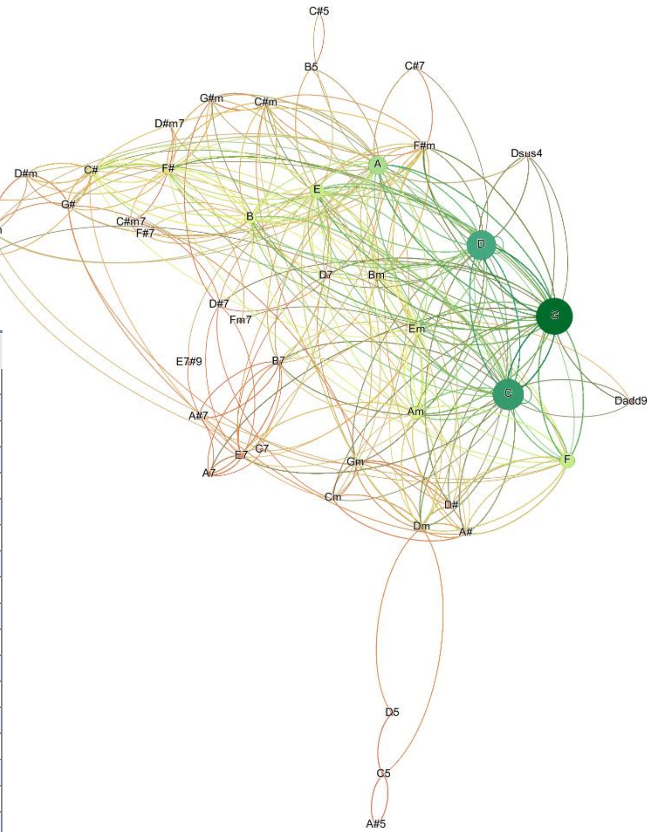

Out-Degree Distribution
Count
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
100
200
300
400
500
600
700
800
900
1,000
1,100
1,200
Value

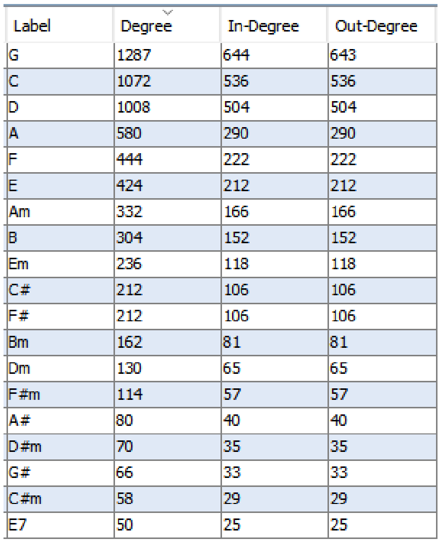
From the graphs above, certain conclusions can be drawn. First of all, the majority of nodes' degree is less than 100, with a few notable exceptions that have degrees over 1000. These are the chords G, C, and D. In fact, the distance between the degree of G and the degree of A, the chord with the fourth highest degree, is greater than the distance between A and the last place (Fm7, with a degree of 2). These results can be easily viewed in the Gephi "Data Laboratory" view.

The maximum degree in the graph is that of G, with an In-Degree of 644 and an Out-Degree of 643, for a total Degree of 1287. Interestingly, the vast majority of nodes have the same In-Degree and Out-Degree.

One could also conclude that G, C and D are the most popular chords in Xatzifragketa's songs, and they would not be necessarily mistaken. A significant amount of their songs is based on a chord progression of the three notes. However, using solely the degree to draw said conclusion is not enough.

The following visualization is using Gephi's size ranking to resize every node based on its Degree, while coloring the edges and vertices to signify transitions between high and low degree nodes. Deeper shades of green represent a high degree node, while the color transitions to a reddish shade when representing lower degree ones.



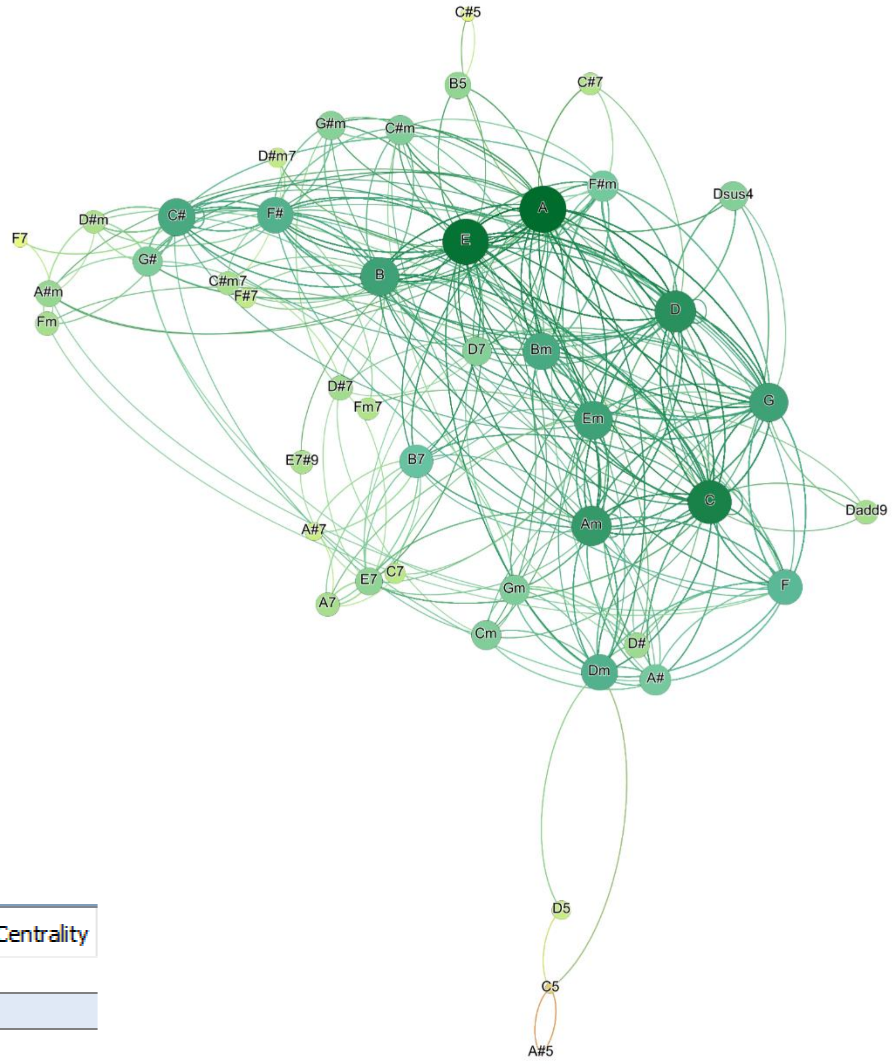


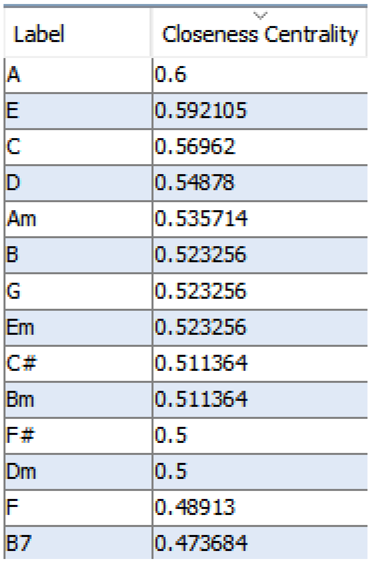
Label	Degree	In-Degree	Out-Degree
G	1287	644	643
C	1072	536	536
D	1008	504	504
A	580	290	290
F	444	222	222
E	424	212	212
Am	332	166	166
B	304	152	152
Em	236	118	118
C#	212	106	106
F#	212	106	106
Bm	162	81	81
Dm	130	65	65
F#m	114	57	57
A#	80	40	40
D#m	70	35	35
G#	66	33	33
C#m	58	29	29
E7	50	25	25

d) Centrality measures

Each node has been resized and colored relative to the measure that is being examined. Below each graph is a sample of the top ranked nodes, according to each measure.

1. Closeness Centrality

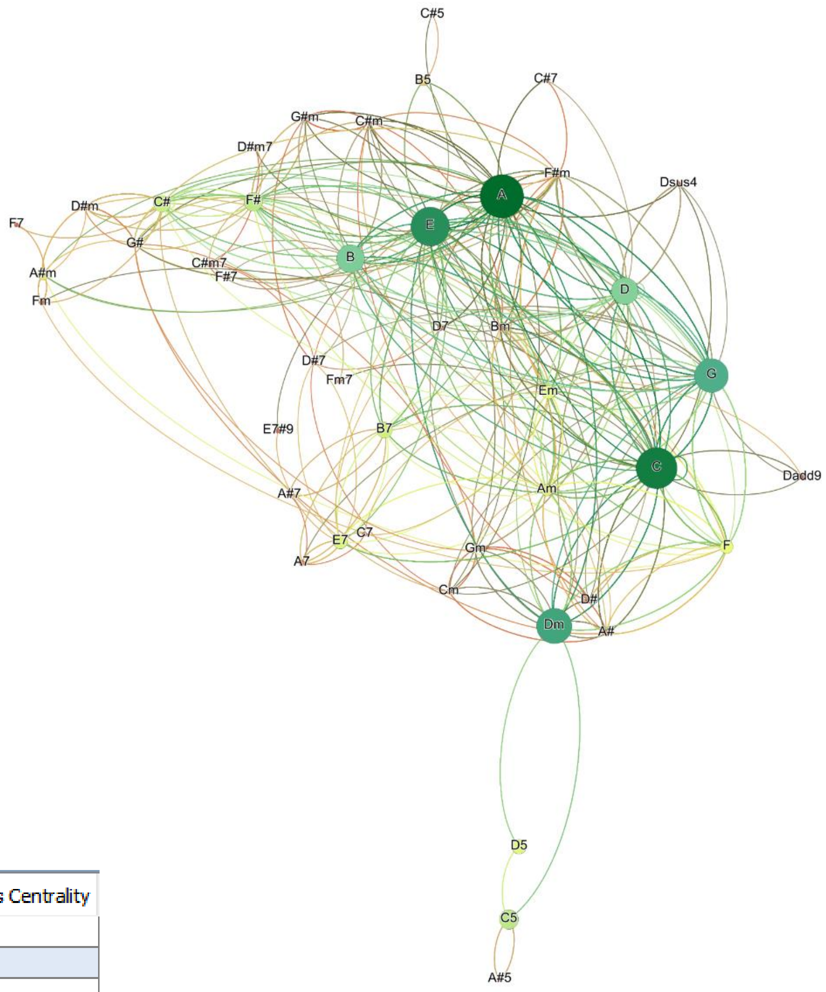


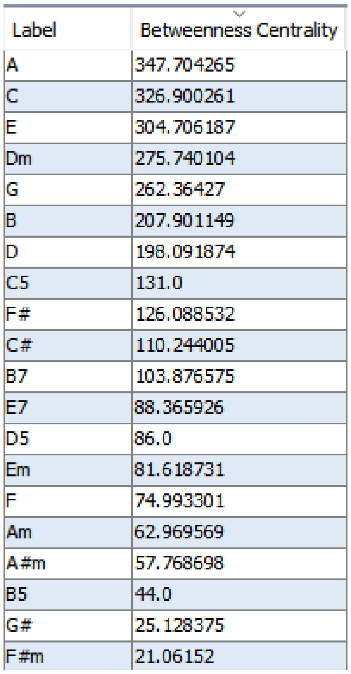


Label	Closeness Centrality
A	0.6
E	0.592105
C	0.56962
D	0.54878
Am	0.535714
B	0.523256
G	0.523256
Em	0.523256
C#	0.511364
Bm	0.511364
F#	0.5
Dm	0.5
F	0.48913
B7	0.473684

A comparison between the Degree Graph and the Closeness Centrality one reveals quite a bit about the way each measure perceives the dataset. Closeness Centrality scores appear to be on the upper percentile of the color gamut available, with only a few nodes painted orange. However, that should not come as a surprise. Closeness centrality attempts to locate nodes that spread information in the graph. It only makes sense that on a graph that consists of nodes that allow the artist to form paths that signify musical scales, that measure tends to be on the higher side.

2. Betweenness Centrality

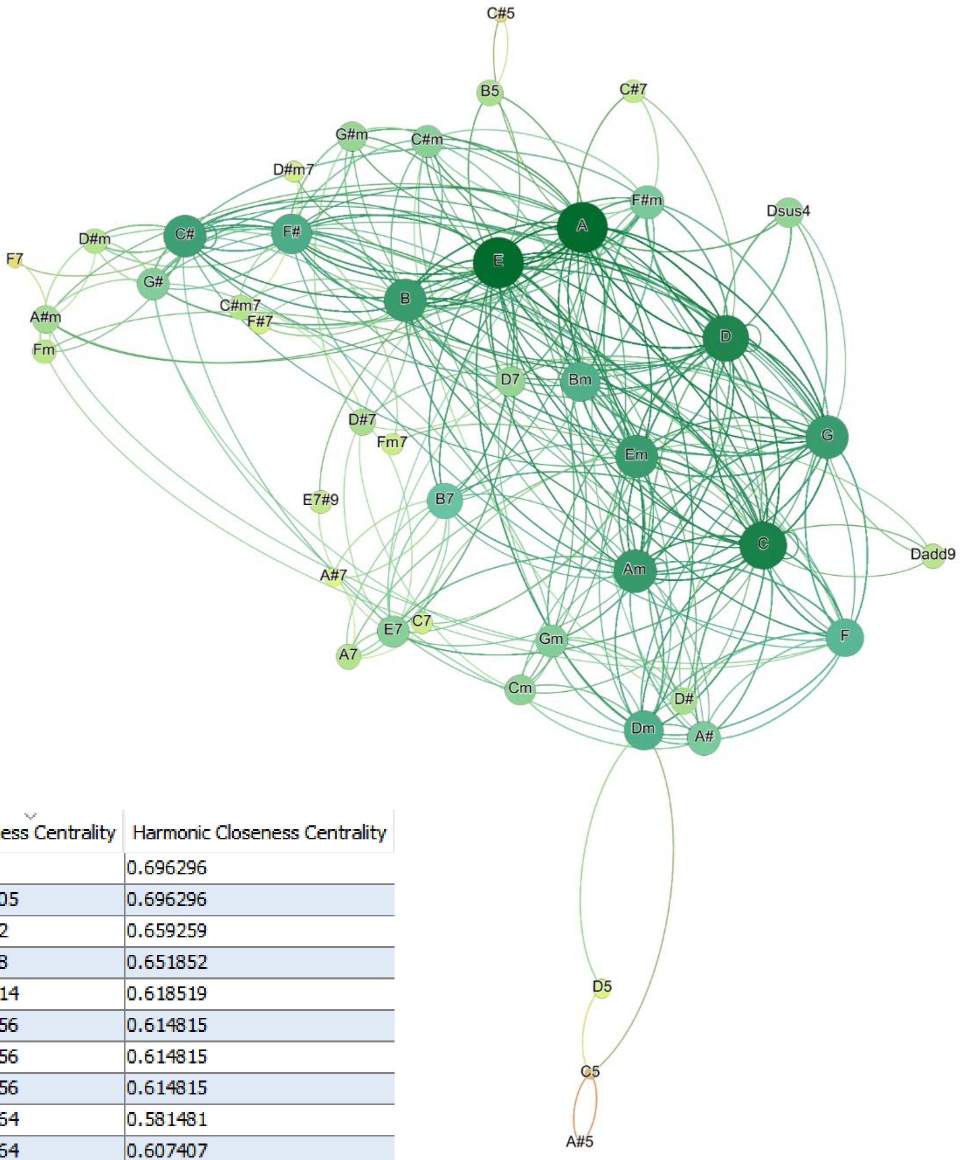


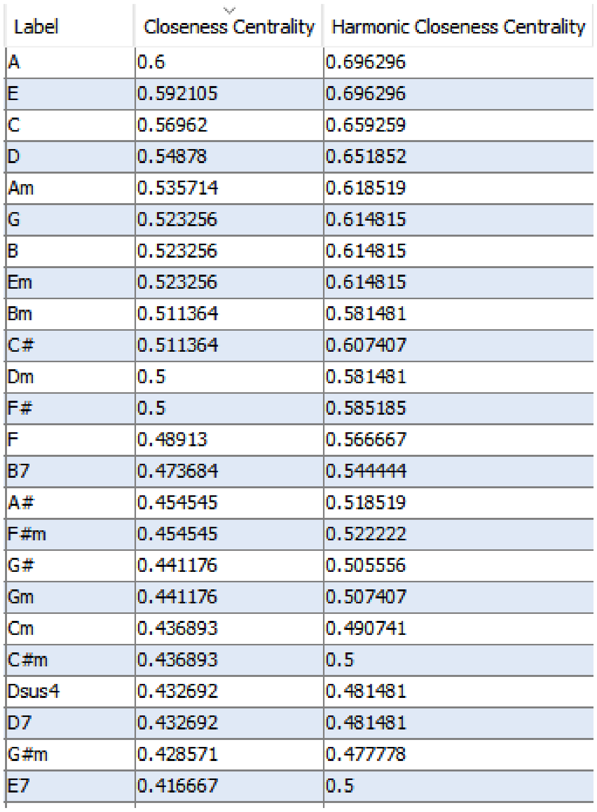


Label	Betweenness Centrality
A	347.704265
C	326.900261
E	304.706187
Dm	275.740104
G	262.36427
B	207.901149
D	198.091874
C5	131.0
F#	126.088532
C#	110.244005
B7	103.876575
E7	88.365926
D5	86.0
Em	81.618731
F	74.993301
Am	62.969569
A#m	57.768698
B5	44.0
G#	25.128375
F#m	21.06152

Betweenness centrality attempts to measure how the flow of information is affected by each node, keeping in mind that the shortest path is being preferred. This results in a graph (when node size and color is proportional to its betweenness centrality) that appears to be kind of a middle ground between the Degree graph and the Closeness Centrality graph. There are quite a few more weak nodes, but still there's a significant number of nodes that are close in size and shade of green. This goes to show how while Closeness Centrality does a good job of evaluating the actual capability of a node, a higher degree node is more likely to be involved in a shortest path. The top nodes/chords in this particular graph are A, E and C, both of which are in the Top 5 nodes when sorted by degree.

3. Harmonic Closeness Centrality

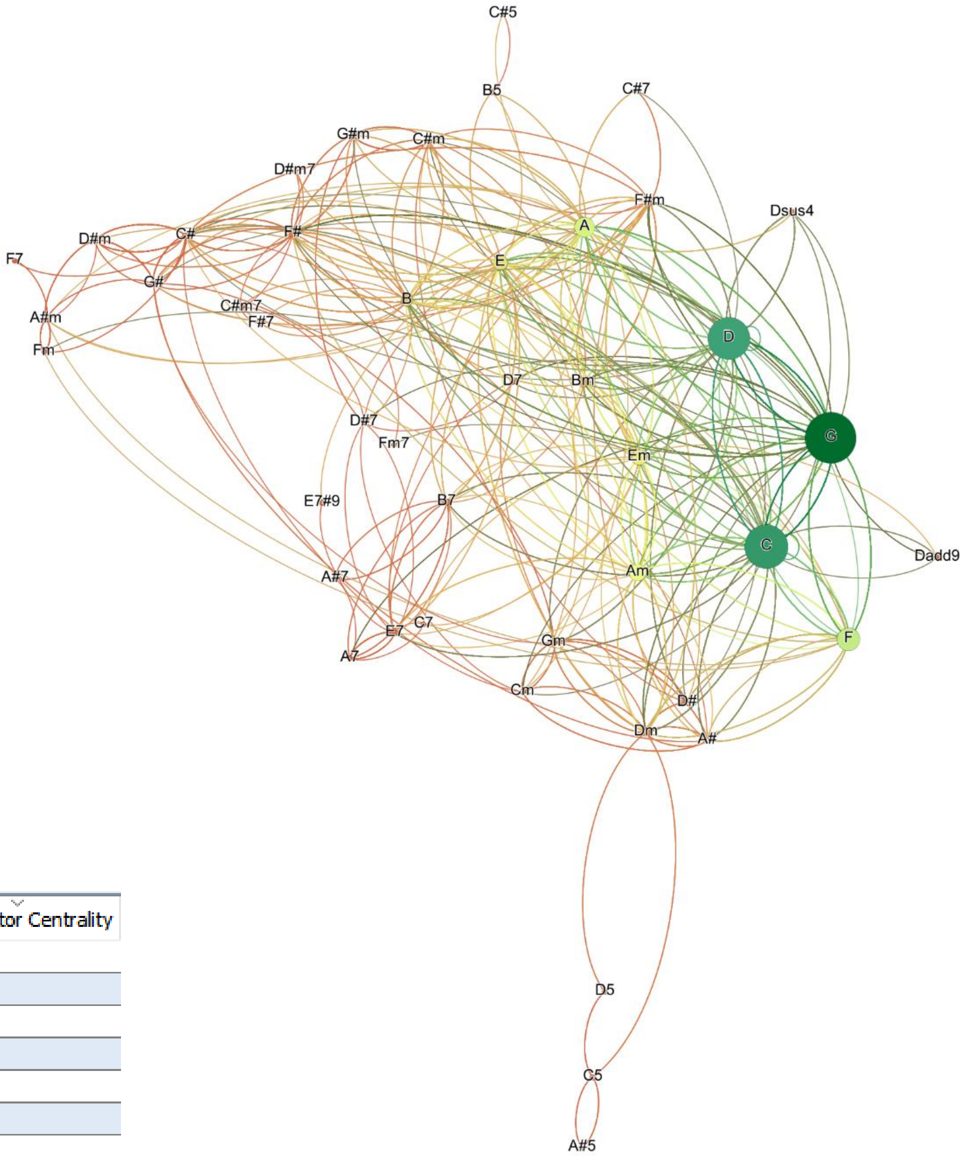


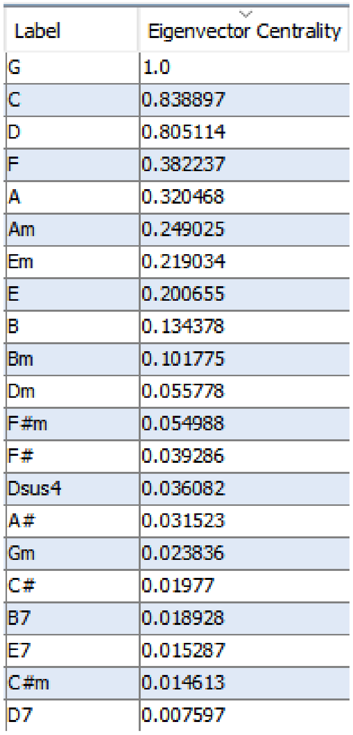


Label	Closeness Centrality	Harmonic Closeness Centrality
A	0.6	0.696296
E	0.592105	0.696296
C	0.56962	0.659259
D	0.54878	0.651852
Am	0.535714	0.618519
G	0.523256	0.614815
B	0.523256	0.614815
Em	0.523256	0.614815
Bm	0.511364	0.581481
C#	0.511364	0.607407
Dm	0.5	0.581481
F#	0.5	0.585185
F	0.48913	0.566667
B7	0.473684	0.544444
A#	0.454545	0.518519
F#m	0.454545	0.522222
G#	0.441176	0.505556
Gm	0.441176	0.507407
Cm	0.436893	0.490741
C#m	0.436893	0.5
Dsus4	0.432692	0.481481
D7	0.432692	0.481481
G#m	0.428571	0.477778
E7	0.416667	0.5

Harmonic Closeness Centrality (HCC) is a measure that is closely related to Closeness Centrality, and thus the results are expected to be pretty close, and so they are. A notable difference is that while both metrics normalize the score in [0,1], HCC has scored every node higher than Closeness Centrality, with a deviation of almost 20% in some cases. For example, in the table to the left (taken from Gephi’s Data Laboratory), node E7 is scored 0.5 on HCC, while Closeness Centrality rated it at just 0.416.

4. Eigenvector Centrality

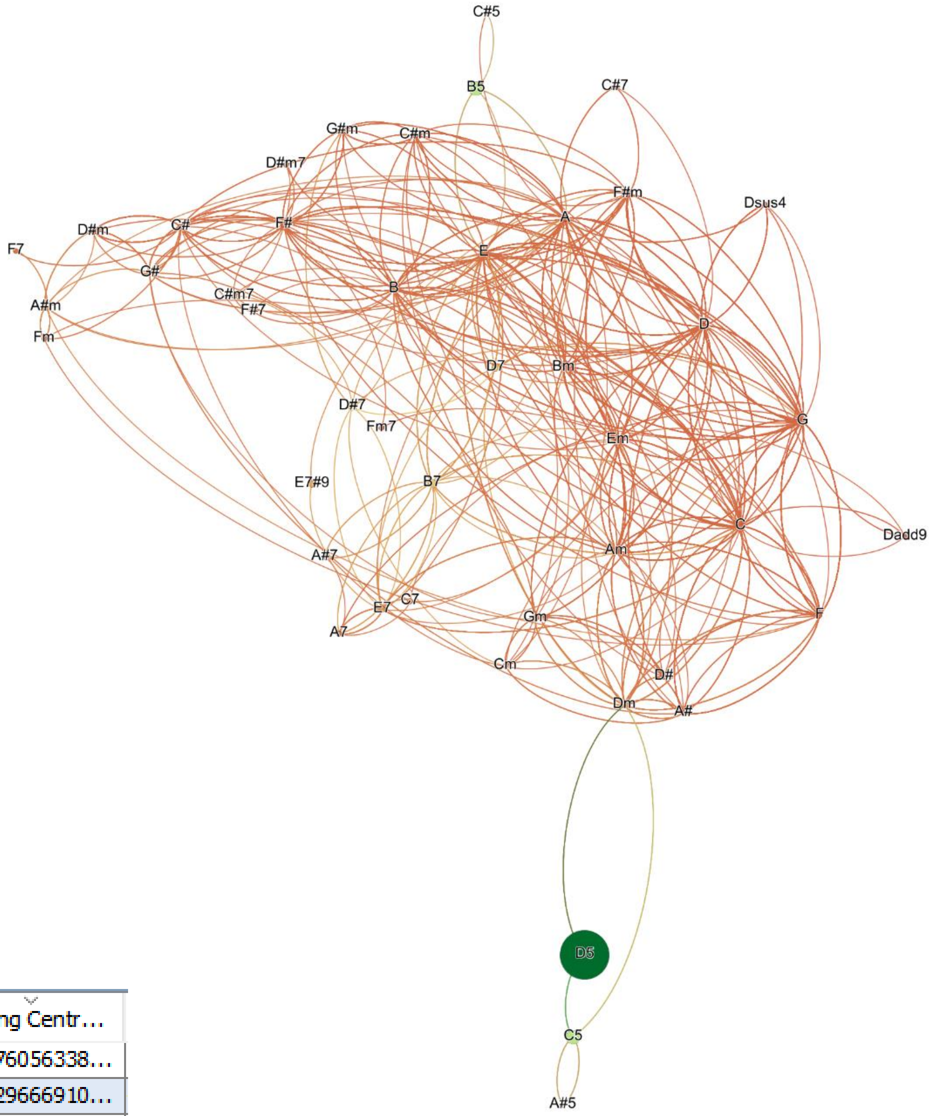


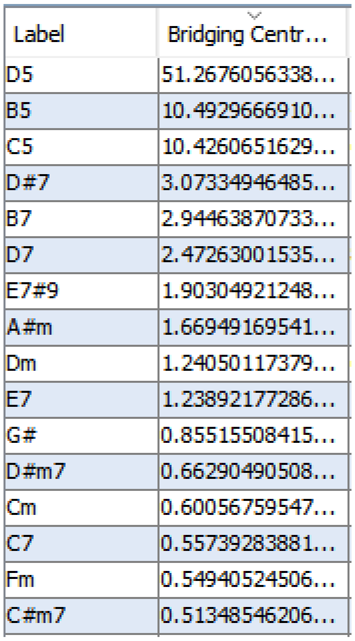


Label	Eigenvector Centrality
G	1.0
C	0.838897
D	0.805114
F	0.382237
A	0.320468
Am	0.249025
Em	0.219034
E	0.200655
B	0.134378
Bm	0.101775
Dm	0.055778
F#m	0.054988
F#	0.039286
Dsus4	0.036082
A#	0.031523
Gm	0.023836
C#	0.01977
B7	0.018928
E7	0.015287
C#m	0.014613
D7	0.007597

Eigenvector Centrality is yet another measure of the influence of each node on the graph. In this instance, the graph coloring and node sizes bear quite a resemblance to the Degree Graph, while the scores tend to be concentrated at the extremes of [0,1]. That is not a surprise when one considers the idea between Eigenvector Centrality. Nodes are scored based on whether or not they are connected to high score nodes. So, it only makes sense that high-degree nodes that rank highly in every metric so far to do just as well when evaluated with this measure.

5. Bridging Centrality





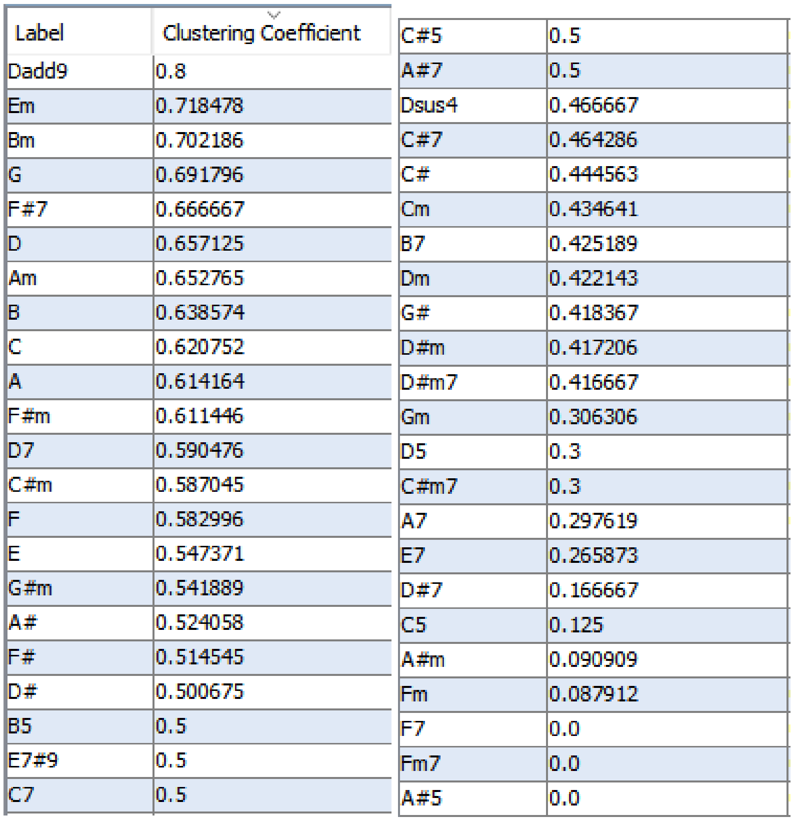
Label	Bridging Centr...
D5	51.2676056338...
B5	10.4929666910...
C5	10.4260651629...
D#7	3.07334946485...
B7	2.94463870733...
D7	2.47263001535...
E7#9	1.90304921248...
A#m	1.66949169541...
Dm	1.24050117379...
E7	1.23892177286...
G#	0.85515508415...
D#m7	0.66290490508...
Cm	0.60056759547...
C7	0.55739283881...
Fm	0.54940524506...
C#m7	0.51348546206...

The last centrality measure that we’re going to use is Bridging Centrality. This is not included in the stock version of Gephi; however, it is available by installing the homonymous plugin. This allows us to run a Bridging Centrality statistic, and color or resize the graph according to the results of said statistic. The reason Bridging Centrality is included in this report, is the fact that it is one of the few metrics that focuses both on information flow and potential local bridges. That produces a graph that is radically different from all other graphs produced so far. Nodes like D5, C5, B5 or D#7 are rated highly, because they are the sole bridge to other nodes in the group above, due to the fact that the graph is directed.

e) Clustering effects

Examining clustering in a graph of this nature is quite important. An attempt to color the nodes based on a metric, as well as rearranging the position of the nodes in the 2D plane may reveal tendencies (in regards to certain nodes being more likely to be connected with certain others) that cannot be observed otherwise.

The first clustering measure to examine will be Gephi's Average Clustering Coefficient, which in this case stands at 0.447. This would suggest that the nodes in the graph tend to avoid clustering ever so slightly.



Label	Clustering Coefficient
Dadd9	0.8
Em	0.718478
Bm	0.702186
G	0.691796
F#7	0.666667
D	0.657125
Am	0.652765
B	0.638574
C	0.620752
A	0.614164
F#m	0.611446
D7	0.590476
C#m	0.587045
F	0.582996
E	0.547371
G#m	0.541889
A#	0.524058
F#	0.514545
D#	0.500675
B5	0.5
E7#9	0.5
C7	0.5
C#5	0.5
A#7	0.5
Dsus4	0.466667
C#7	0.464286
C#	0.444563
Cm	0.434641
B7	0.425189
Dm	0.422143
G#	0.418367
D#m	0.417206
D#m7	0.416667
Gm	0.306306
D5	0.3
C#m7	0.3
A7	0.297619
E7	0.265873
D#7	0.166667
C5	0.125
A#m	0.090909
Fm	0.087912
F7	0.0
Fm7	0.0
A#5	0.0

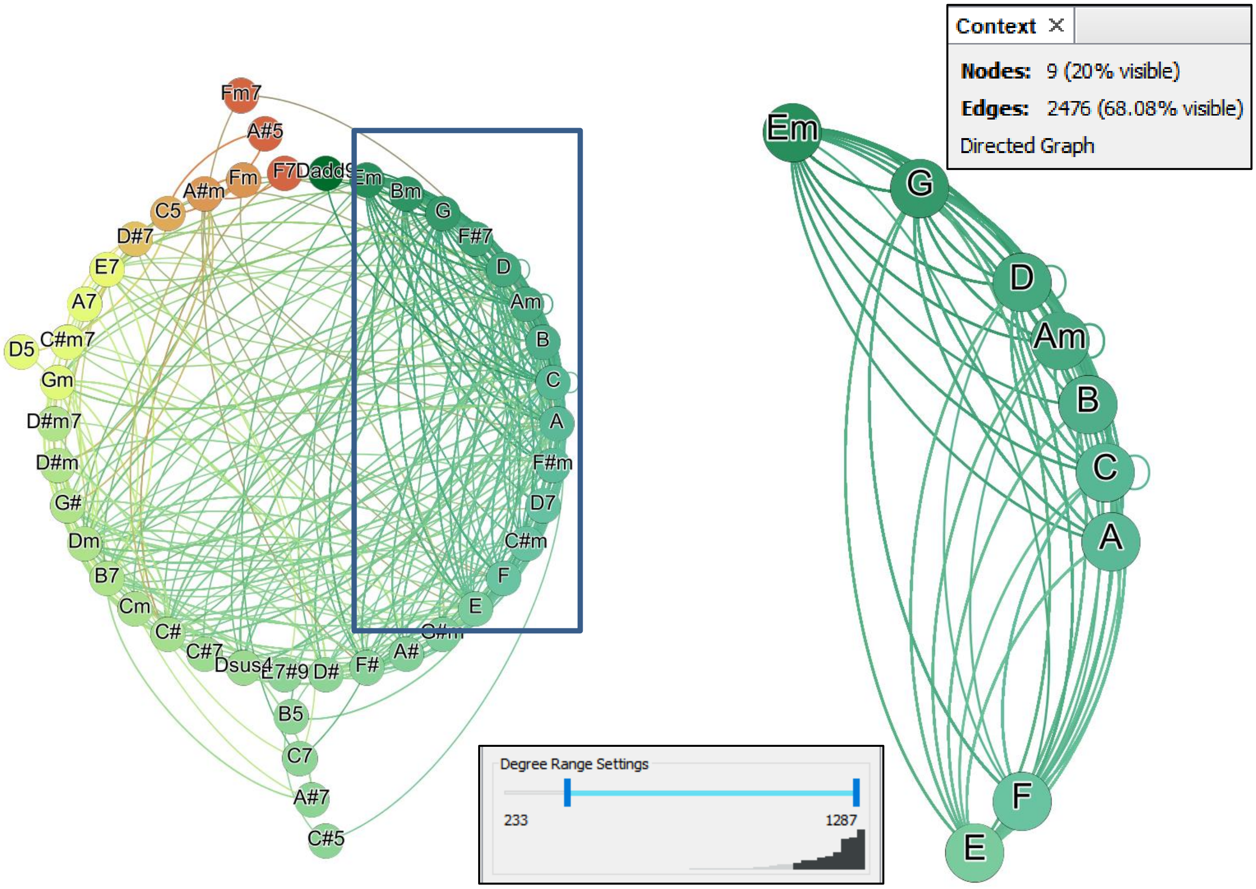
However, average results tend to be affected by the extreme values of a set. It is obvious that the bottom three nodes in Clustering Coefficient, F7, Fm7 and A#5 have a non-existent value. This is can be misleading. In fact, over half the nodes in the graph show a strong to mild tendency to form clusters.

At this point it is important to mention that yet another Gephi plugin was tested. It is called Clustering Coefficient, and claims to calculate (obviously) the Clustering Coefficient and also the number of triangles present in the graph, among other metrics. The Clustering Coefficient that was produced by said plugin was on almost every node quite a bit higher than the stock Gephi calculation. The average clustering coefficient, when calculated from the aforementioned column, was a significantly stronger 0.588.

Clustering is also affected by the number of triangles in the graph. Using the same plugin as previously, the total triangles in the graph were found to be 245891, along with an average Clustering Coefficient of 0.77. It must be noted that this was calculated using a triangle method, and thus yielded different results from the previous two implementations of Clustering Coefficient.

Studying clustering also entails determining whether or not triadic closure exists in this graph. Triadic closure is often dependent on metrics like the Clustering Coefficient. Assuming a high clustering coefficient, triadic closure does exist between three given nodes. By examining the graph, it becomes apparent that strong ties tend to appear in triangles that involve high-degree nodes like G, C and D. However, one could argue that while triadic closure does appear, due to the fact that these chords tend to appear in quite a few songs, and can be combined quite easily through some quite easy scales and patterns, it could be merely a side effect. On the other hand, there’s an argument to be made about what qualifies as friendship in this type of graph. Could co-existence in music theory scales be considered valid as a friendship factor? There seem to be quite a few indications that that is the case, especially if someone takes in consideration the Centrality measures analyzed previously.

The current state of the graph, in terms of the position of the nodes on the plane, was produced using Force-Atlas, suggested by the Gephi tutorials. Searching for a way to visualize the tendency (or lack thereof) to form clusters, the following graphs were produced using another Gephi plugin called Circular Layout. The name may be a bit misleading, since the plugin introduces a both a Circular Layout option in the Layout menu, as well as a Radial Axis Layout, that claimed to be able to visualize said tendencies. Radial Axis Layout offers a wide range of options, such as grouping nodes by a certain measure (Clustering Coefficient was used) as well as an option to select the measure with which the nodes will be ordered. The idea was to determine whether or not we could notice a clustering trend between certain nodes, by attempting to rearrange the nodes not in a way that is necessarily user-friendly, but easier to study from this specific standpoint.



Context ✕

Nodes: 9 (20% visible)

Edges: 2476 (68.08% visible)

Directed Graph

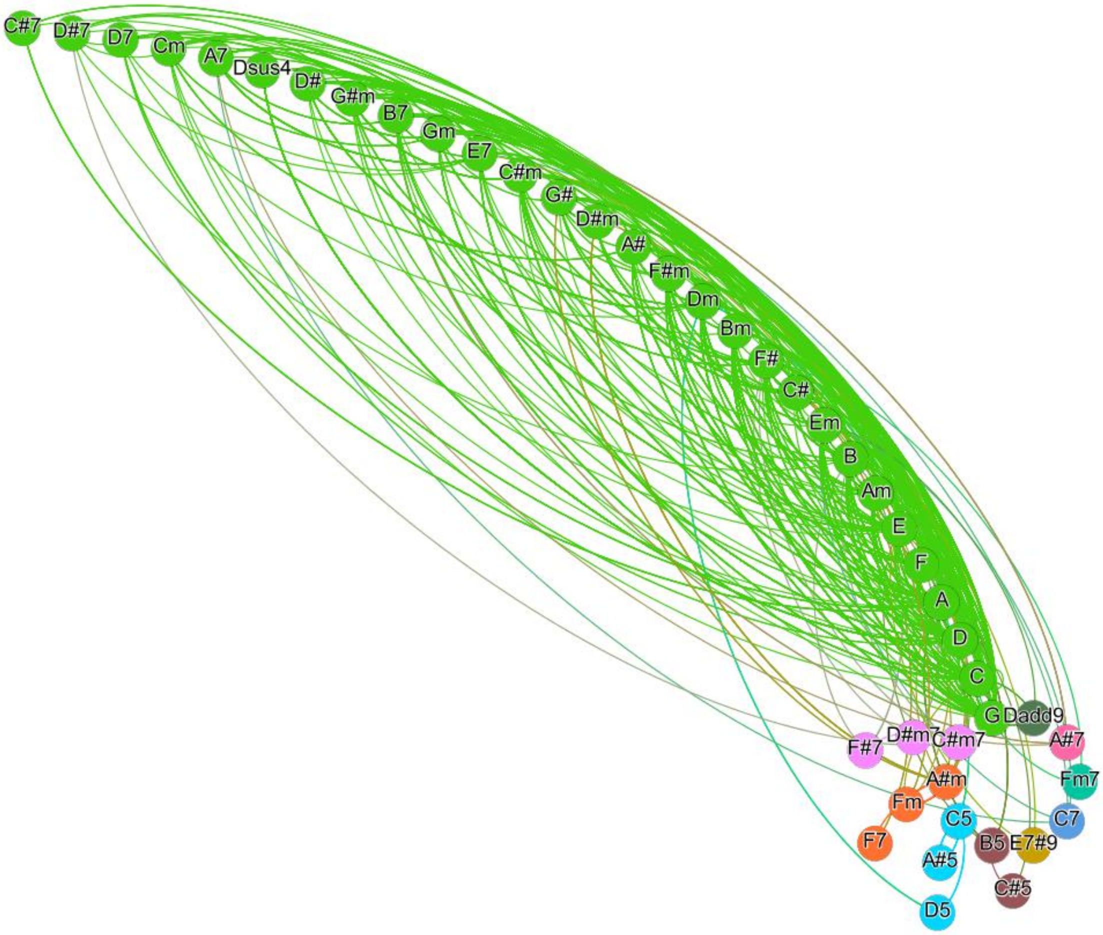
Degree Range Settings

233 1287

The graph to the left is the same graph as before, colored by Clustering Coefficient and arranged on a Radial Axis. At first glance, there's no clear conclusion, since even high value nodes have plenty of edges that are connected to supposedly weaker nodes.

In order to focus on the nodes most likely to produce a clustering symptom, a Degree Filter was used in Gephi. That created a subgraph that consisted of just 9 nodes, which even without using a metric show a very clustering tendency. To confirm this suspicion, the subgraph was evaluated once again using Gephi's Average Clustering Coefficient, which made an impressive jump from 0.447 to 0.70! This shows beyond doubt that clustering does exist in the graph, even in some smaller subsets. This particular one consists of just 20% of the nodes, but these nodes account for 68% of the graph's total edges. So it can actually be argued that this cluster plays a pivotal role when extracting information from the graph.

The last attempt to identify clusters was done using both Radial Axis and yet another Gephi plugin, called Leiden Algorithm. It's description claims that the Leiden algorithm is both efficient and good at recognizing well-connected communities. In contrast with the majority of Gephi measures used so far, that tend to allow little to no fine-tuning through parameters, Leiden allows the user to set the resolution, which controls the granularity of the graph, and also choose the quality function. In this case, the default "CPM" function was used, since the alternative (Modularity) is available through Gephi already, and is going to be examined later. It must be noted that when working with an algorithm that depends heavily into the parameters set by the user, the results have to be checked and the parameters tuned until a satisfying result is achieved.



Using a resolution of 0.2 and arranging the nodes using a Radial Axis produced 10 clusters, the nodes of each one was colored accordingly. Now obviously, the algorithm has identified clusters that contain just a single node. That is obviously a flawed conclusion to draw. However, it did manage to create a large cluster containing the nodes we examined earlier, and a couple of quite interesting smaller ones. Focusing on the F#7, D#7 and C#m7 cluster, it's interesting to notice that the algorithm managed to recognize that all these are 7th chords, meaning they consist of three individual notes (the low or root one being

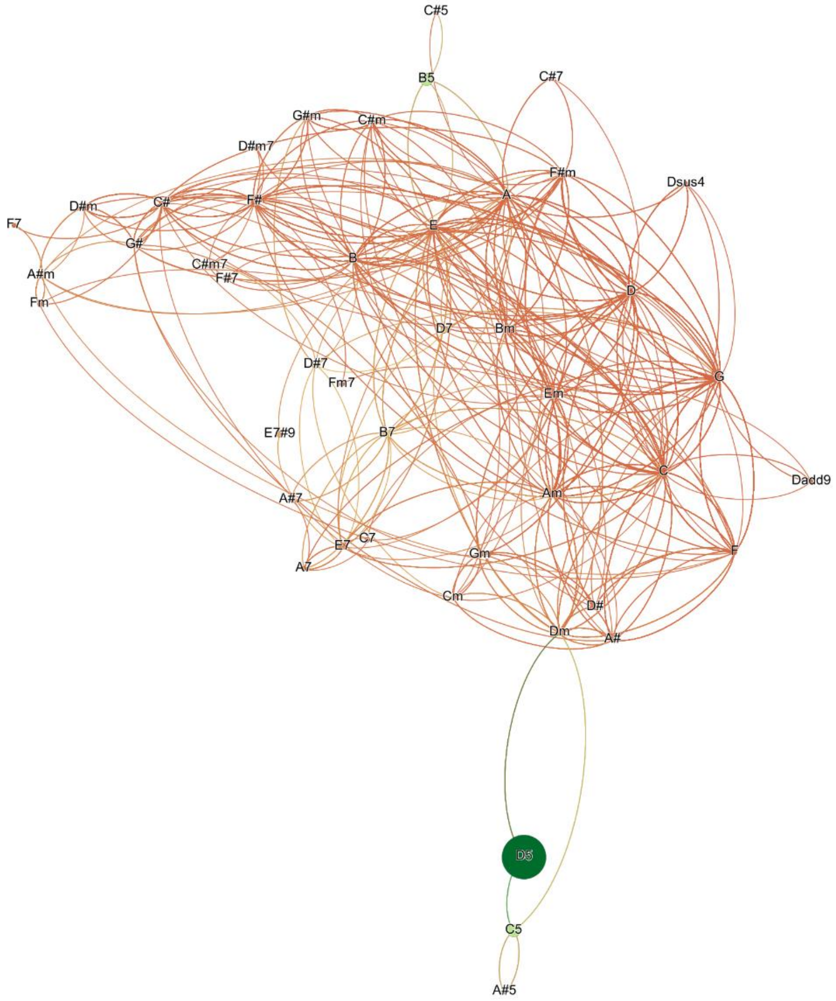
F, D, and C respectively) and a note that has an interval of a seventh above the chord's root. Effectively, this means it is easier to distinguish the direction or flow of the melody. The root notes of these chords are also part of the F# major scale (among others). Obviously, there's a high chance that these chords were used in a song and thus have strong ties between them, and the algorithm is merely pointing that out. However, if someone approached this graph being agnostic about the nature of the ties or transitions between each node, this particular clustering algorithm would point them towards exploring why this little cluster was created. There are also two more clusters of size 3, F7, Fm, A#m as well as D5, A#5, C5. The former contains essentially two chords with the same root note, and an A#. Interestingly, F# and A# belong in the F# Major Pentatonic scale, if we accept that as the root. As for the latter, it's easy to notice that it consists of fifth notes (also called power chords), that can be combined in another Major scale, the A Major.

It is clear that since the Leiden Algorithm shows a lot of flexibility, it can be misleading. The resolution parameter can be tuned to manipulate results to fit certain narratives, without guaranteeing that the results are actually valid. That is why it is important to use a wide range of metrics, especially on topics that tend to be vague, like Clustering.

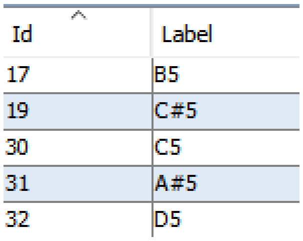
f) Bridges

In the Centrality Measures section, Bridging Centrality was examined. At the time this report was created (December 2019 to February 2020), Gephi does not provide an automated statistic or filter to recognize bridges, either global or local. Since we do have access to bridging centrality, we can still draw some conclusions. The graph on the next page is the Bridging Centrality one, where each node has been color-ranked and sized depending on its Bridging centrality.

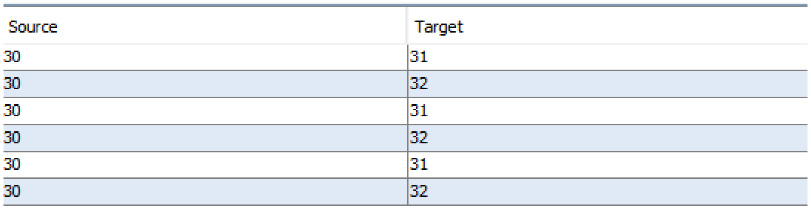
Something that is also quite important to keep in mind is the fact that every node has at least one edge incoming and one edge leaving (In-degree and Out-degree are at least 1). This means that if the graph is treated as undirected, there won't be any bridges unless the edges are merged. However, since this is a directed graph, we can still consider that the existence of an edge (and thus a transition from a note to another) is a one-way function. In that case, bridges could appear in a number of nodes. That may be a bit misleading, since in most cases two notes that co-exist in a scale can be used repeatedly, but it must be highlighted how certain transitions may make more sense than others, on a greater scope than the one defined by music theory. After all, any combination of notes in a scale is "correct" from a theory standpoint. The purpose of this report is to question whether or not there exists a rule to limit the number of said combinations, to increase the probability of the final product being aesthetically pleasing.



On this particular case, it is clear that node D5 forms a local bridge between the giant component and nodes C5, A#5. Deleting the edges that start from D5 and end at Dm, would mean that there is no way to draw a path from C5 or A#5 to Dm, or any other node of the graph. There might be edges that start from Dm and directly access C5, however as we previously mentioned, those are directed and there are no edges starting from C5 or A#5 towards the rest of the graph. Using Gephi's Data Laboratory confirms this claim.



Id	Label
17	B5
19	C#5
30	C5
31	A#5
32	D5

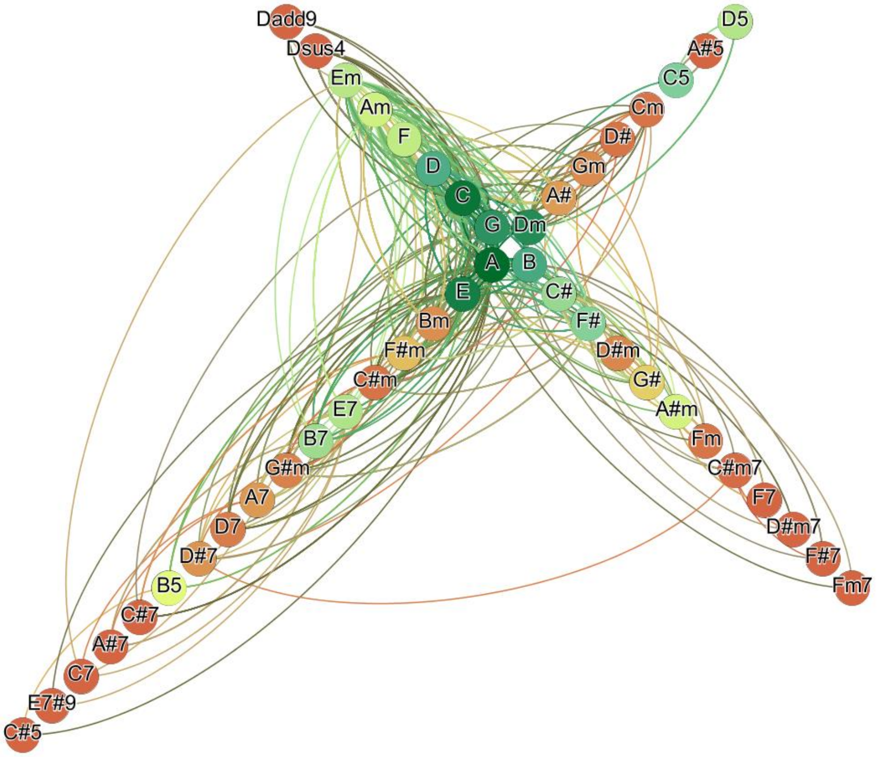


Source	Target
30	31
30	32
30	31
30	32
30	31
30	32

Another node that is rated quite highly by Bridging Centrality is B5. Examining the Data Laboratory reveals that B5 is the only node that has an edge coming in to it from C#5. What this means, is that this edge is common in every shortest path drawn where the target is C#5 (or, in case of a Steiner Tree, C#5 is one of the nodes we seek to cover).

g) Homophily

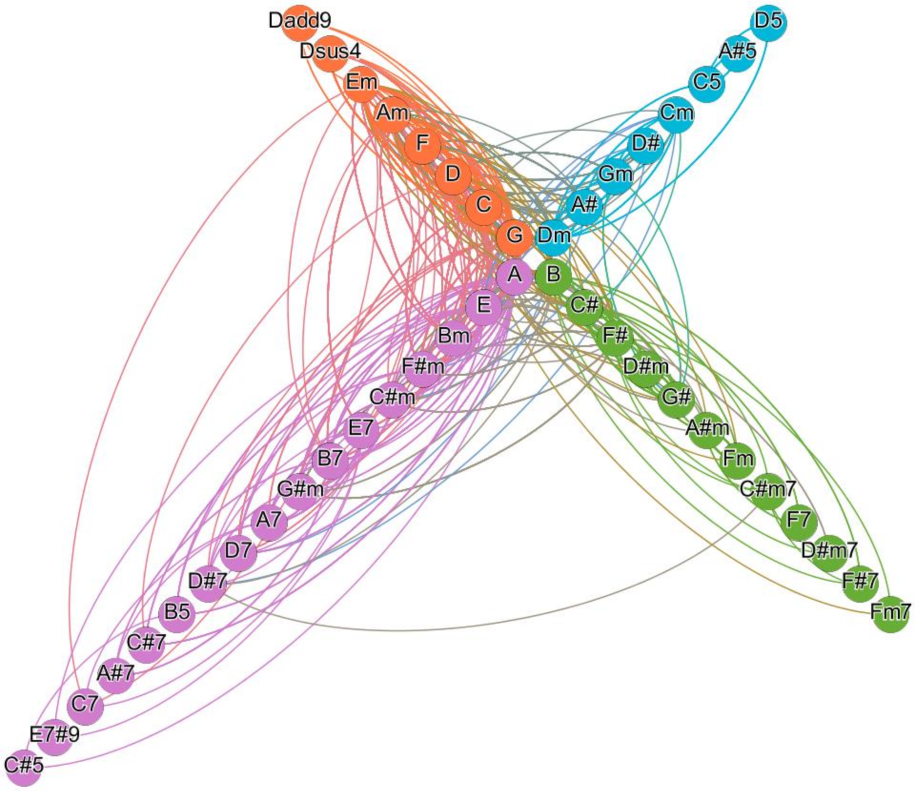
Homophily is a concept that is highly related to clustering. This means we will examine Radial Axis Graphs once again. The question is whether nodes that share a common characteristic tend to create friendships.



The graph above has been generated with the nodes grouped by their modularity class (which is another measure that is related to clustering, that will be examined later), and colored according to their Betweenness Centrality. Once again, the majority of the edges appears to be between nodes in the cluster that contains Em, D, C, and G amongst others. However, there doesn't appear to be much of tie between nodes that have been grouped together apart from that group. Most of the high-degree nodes belong in that specific cluster so it makes sense for them to gather so much attention from both their "friend" nodes, as well as nodes that belong in different communities.

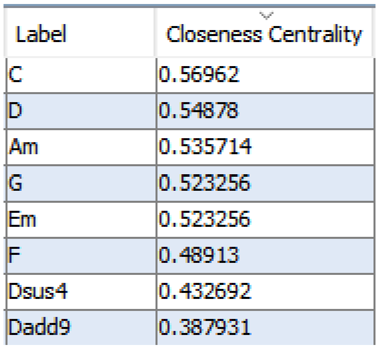
h) Modularity

Gephi has an integrated statistic called Modularity, which attempts to determine the how high the graph's tendency to create clusters (called communities) is. For this graph, with a resolution of 1.0 (same parameter used in the Leiden algorithm) 4 communities were discovered, with a modularity of 0.278, which suggest the graph doesn't really tend to create communities.



The graph above is once again created with a Radial Axis Layout and with the nodes grouped by their modularity class, with the centermost nodes being the ones with the highest degree in each community. They are also colored accordingly. It is fairly obvious that nodes tend to form ties with other nodes that belong in the same community, with the most popular ones being the high-degree nodes.

This seems trivial. However, taking a look at the Data Laboratory, reveals something quite interesting. Modularity Class 0 contains nodes G, C, Em, Am, D, along with chord F and two D chord variants (a suspended and an

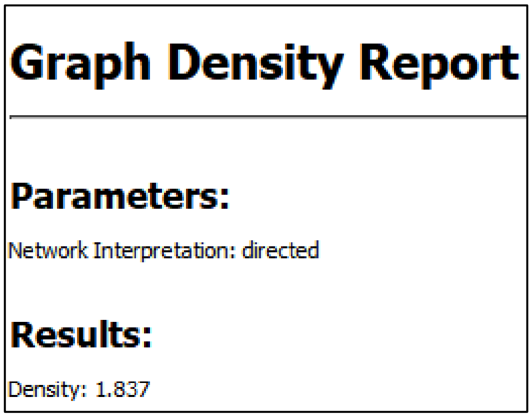


Label	Closeness Centrality
C	0.56962
D	0.54878
Am	0.535714
G	0.523256
Em	0.523256
F	0.48913
Dsus4	0.432692
Dadd9	0.387931

added chord). At first, this makes sense, since these are all quite high degree nodes and there's an increased chance that they share a big number of edges. That is not the case regarding Dsus4 and Dadd9, however they are often used in between D chord strumming. What's interesting is that C, D, Am, G, and Em are the exact chords used in the (by far) most popular Xatzifragketa song, "Taxiko". Effectively, the set of 45 chords that we began with could be shrunk into this community and still produce an iconic song. To be fair, there are tons of songs out there that make use of these exact chords, for example Metallica's "Unforgiven", that uses Am, C, D, Em, E, G, A and Asus2, and The Animals' "House of The Rising Sun", that uses Am, C, D, E. It is pretty interesting to note that this community managed to contain the exact chords needed, without one going missing. What makes this even more impressive is how nodes like A, E and B do not belong in this community, despite having a much higher score than Em in numerous measures, like Degree or Closeness centrality. Of course, chord F appears as well in the community, and is absent on the song, but it also belongs in the key of Am.

It needs to be stressed that the phenomenon observed above can be considered a fun coincidence. This community after all contains chords that are frequently combined, and can also be considered "beginner" chords. I still think it is noteworthy that the band's most popular song uses only chords that belong in this particular community.

i) Graph density



Graph Density Report

Parameters:

Network Interpretation: directed

Results:

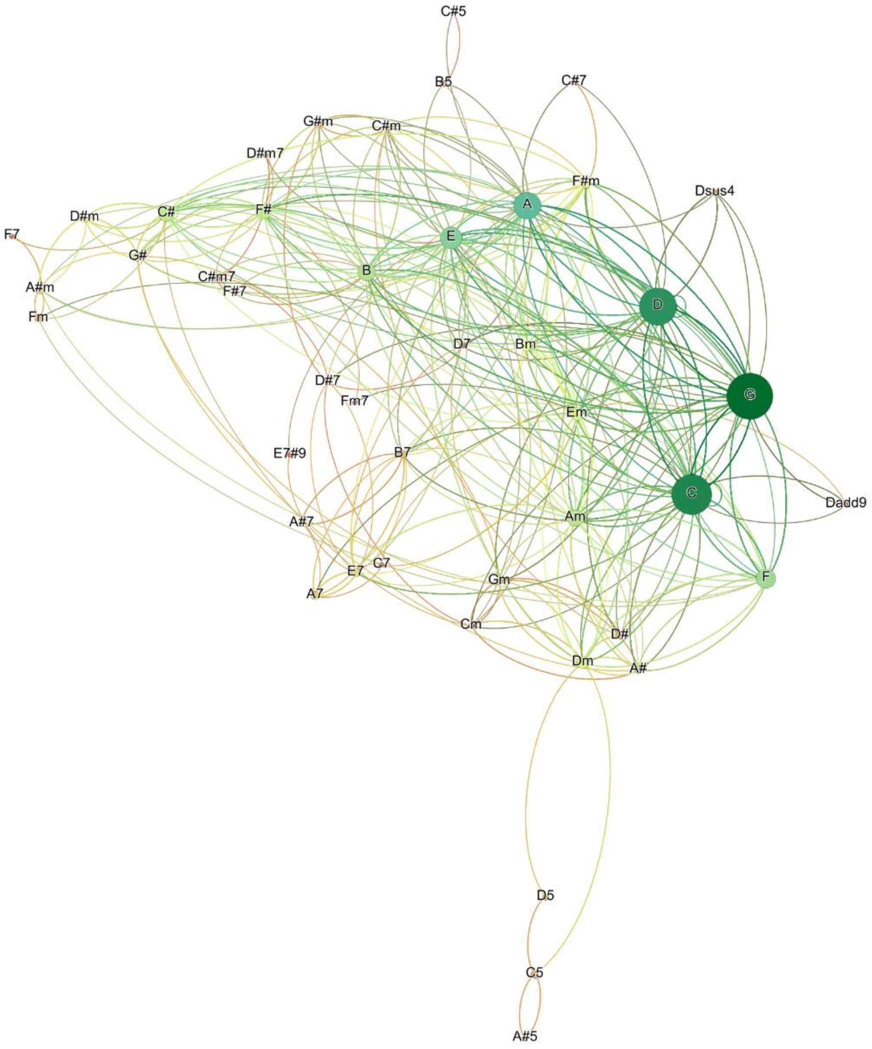
Density: 1.837

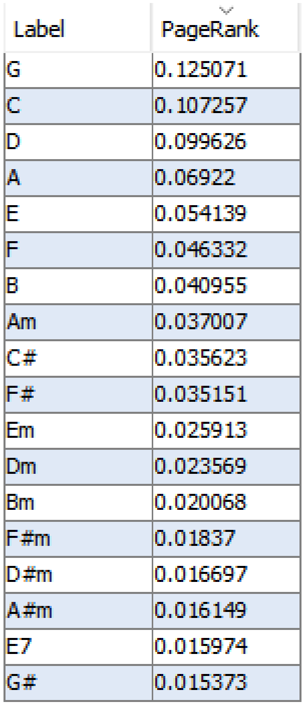
One of the last measures that will be used will be Graph density, which calculates how close the graph is to being a complete graph. In this case, where there are multiple nodes with loops and multiple edges between two nodes, this is not a highly important measure.

As expected, the graph is considered to be quite dense. Although it is not complete, the existence of a giant component with multiple edges between multiple nodes, points towards this result making sense. After all, while examining Clustering, it became obvious that 20% of the nodes are responsible for 68% of the edges in the graph.

j) PageRank

Last but not least, we will visualize the importance of each node based on its PageRank, an algorithm developed by Google's Larry Page. PageRank has been used widely through the years to measure the importance of Web Pages, so it will be interesting to see how it performs on a different type of graph.

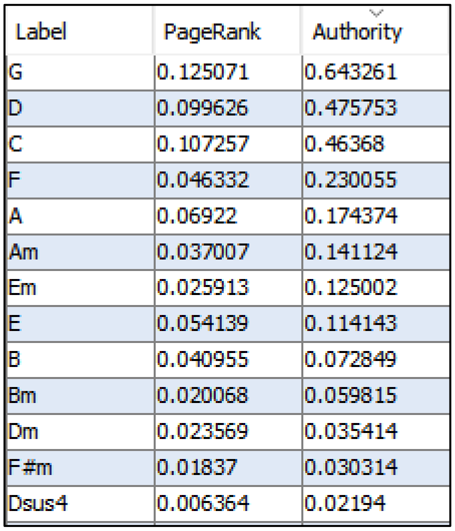




Label	PageRank
G	0.125071
C	0.107257
D	0.099626
A	0.06922
E	0.054139
F	0.046332
B	0.040955
Am	0.037007
C#	0.035623
F#	0.035151
Em	0.025913
Dm	0.023569
Bm	0.020068
F#m	0.01837
D#m	0.016697
A#m	0.016149
E7	0.015974
G#	0.015373

It is obvious that certain nodes are going to rank highly no matter the metric, and PageRank offers little to dispute this fact, other than minor adjustments. Consistency across measures is something everyone seeks when trying to extract information from a graph. PageRank is affected quite a bit by high degree values, and that is obvious in this list as well, though to be fair, chord F may have had a higher degree than E, but it trails the latter in the PageRank score table. What is interesting is that the PageRank scores really take a hit after the top 5 or 6 chords, suggesting a major decrease in importance.

It is also interesting to compare PageRank Results to the ones calculated using Gephi's HITS statistic, that is also developed primarily for webpages. Although they both agree that G is the most important node in the graph, chord E went from 5th place down to 8th, with its minor variation, Em, going from 11th to 7th.



Label	PageRank	Authority
G	0.125071	0.643261
D	0.099626	0.475753
C	0.107257	0.46368
F	0.046332	0.230055
A	0.06922	0.174374
Am	0.037007	0.141124
Em	0.025913	0.125002
E	0.054139	0.114143
B	0.040955	0.072849
Bm	0.020068	0.059815
Dm	0.023569	0.035414
F#m	0.01837	0.030314
Dsus4	0.006364	0.02194

VII. Conclusion

In order to summarize, any conclusions drawn out of this analysis must be interpreted as suggestive, not definitive. The nature of music is such that a graph of this kind offers an insight as to how certain music theory rules find their way into actual mainstream compositions. It also offers an insight on how each measure works, with some of them being affected by high degree nodes, while other measures attempted to dig deeper and search for other ways to identify important nodes.

That being said, there are multiple measures and graphs that point towards reinforcing the idea that theoretical concepts in modern music are still highly prevalent (i.e. the Leiden algorithm communities). At the same time, one can also observe that in most of the rankings, major chords are more important than minor chords, despite these two groups having strong ties with one another. Sometimes though, the devil's in the details. Minor chords may sound sad, but they always have a place in a song. Like Bob Ross once said, "Gotta have a little sadness once in a while so you know when the good times come."

VIII. Sources

a. Social Network Analysis lectures and presentations, DMST AUEB

b. Gephi Layouts Tutorial, https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

c. Gephi Wiki, https://github.com/gephi/gephi/wiki

d. “From Louvain to Leiden: guaranteeing well-connected communities”, https://www.nature.com/articles/s41598-019-41695-z

e. Dataset scraped from https://kithara.to/

f. Centrality measures, https://en.wikipedia.org/wiki/Centrality