Γράφος (G)

S – υποσύνολο του G, set από κόμβους

Τ- υποσύνολο του G, set από κόμβους που δεν ανήκουν στο S

Στη περίπτωση μας το S είναι οι κόμβοι που ανήκουν σε κάποιο cluster από αυτά που έχει ένα annotation αρχείο και Τ είναι οι υπόλοιποι κόμβοι.

Ο κώδικας σε Python του Conductance είναι εδώ: <https://networkx.org/documentation/stable/_modules/networkx/algorithms/cuts.html#conductance>

Η συνάρτηση conductance κάνει χρήση των εξής συναρτήσεων (ο κώδικας τους είναι επίσης στο πάνω link):

**Cut\_size** (αριθμός) – το άθροισμα των βαρών των ακμών μεταξύ των δύο set S & T. Εάν ο γράφος είναι χωρίς βάρη, το βάρος κάθε ακμής είναι 1.

**Volume** (αριθμός) - το άθροισμα των degrees των nodes από S & T. Αν ο γράφος είναι directed, αντί για degrees 🡪 out-degrees

**Edge\_Boundary** (source code: [https://networkx.org/documentation//networkx-1.10/\_modules/networkx/algorithms/boundary.html#edge\_boundary](https://networkx.org/documentation/networkx-1.10/_modules/networkx/algorithms/boundary.html#edge_boundary))

*Το χρησιμοποιεί το cut\_size*

1. Για κάθε node (έστω n1) που ανήκει στο S

2.Για κάθε node γενικά του γράφου G (έστω n2) που συνδέεται με το n1

3.Αν το n2 υπάρχει στο Τ 🡪 Επιστρέφει το ζεύγος (n1,n2) και εμείς παίρνουμε το άθροισμα των βαρών των ζευγών αυτών.

**Conductance = cut\_size / min(volume(S),volume(T))**

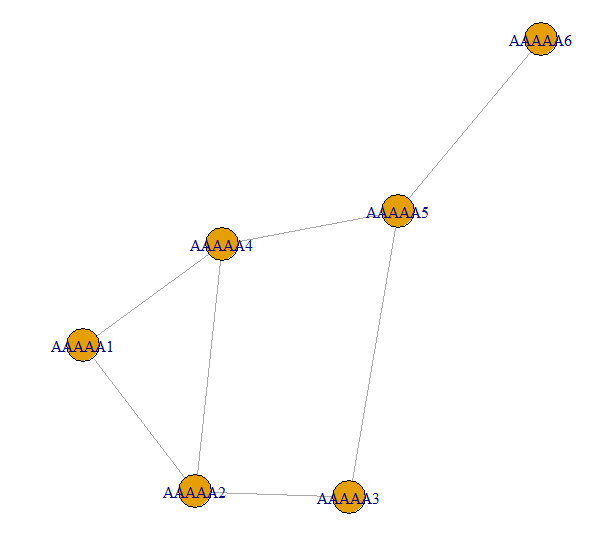
Το conductance ορίζεται για ένα set κόμβων και όχι για πολλαπλά sets. Το κάθε set περιέχει τους κόμβους του κάθε cluster από το annotation file.

To conductance μετράει πόσο καλό είναι ένα cluster και όχι πόσο καλά ένας ολόκληρος γράφος μπορεί να χωριστεί.

Ιδανικά, καλό είναι να έχουμε χαμηλό conductance (κοντά στο 0) και αυτό επιτυγχάνεται εφόσον έχουμε μικρό αριθμό ακμών που συνδέουν τα S και Τ και ταυτοχρόνως υψηλό αριθμό στα Volume των S και Τ.

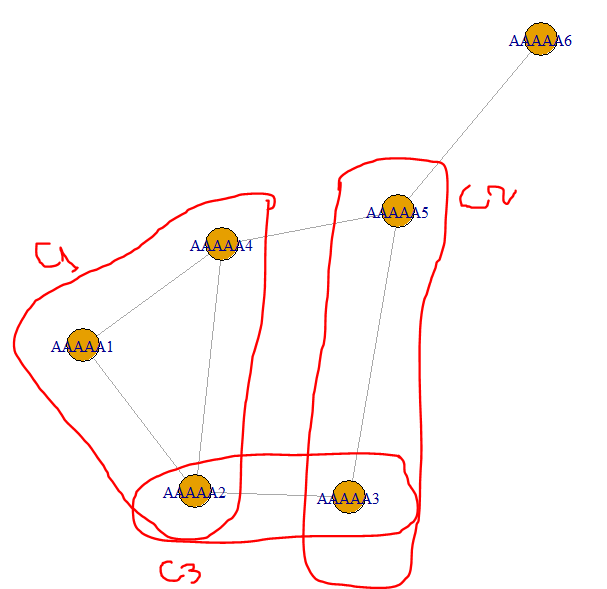
Παράδειγμα 1

Έστω πως το G είναι αυτό



Από το annotation έχουμε τα εξής cluster





Εδώ τα set διαμορφώνονται ως εξής:

**C1**

S1 = {"AAAAA1","AAAAA2","AAAAA4"}

T1= {"AAAAA6","AAAAA3","AAAAA5"}

Cut\_size1=2

Volume(S1)=8

Volume(T1)= 6

**Cond1=0.333**

**C2**

S2= {"AAAAA3","AAAAA5"}

T2= {"AAAAA6","AAAAA1","AAAAA2","AAAAA4"}

Cut\_size2=3 (από «ΑΑΑΑΑ4»-«ΑΑΑΑΑ5», «ΑΑΑΑΑ6»-«ΑΑΑΑΑ5» & «ΑΑΑΑΑ3»-«ΑΑΑΑΑ2»)

Volume(S2)=5

Volume(T2)= 2+3+3+1=9

**Cond2=3/5=0.6**

**C3**

S3= {"AAAAA3","AAAAA2"}

T3 = {"AAAAA6"}

Cut\_size3=3 (από «ΑΑΑΑΑ2»-«ΑΑΑΑΑ1», «ΑΑΑΑΑ2»-«ΑΑΑΑΑ4» & «ΑΑΑΑΑ3»-«ΑΑΑΑΑ5»)

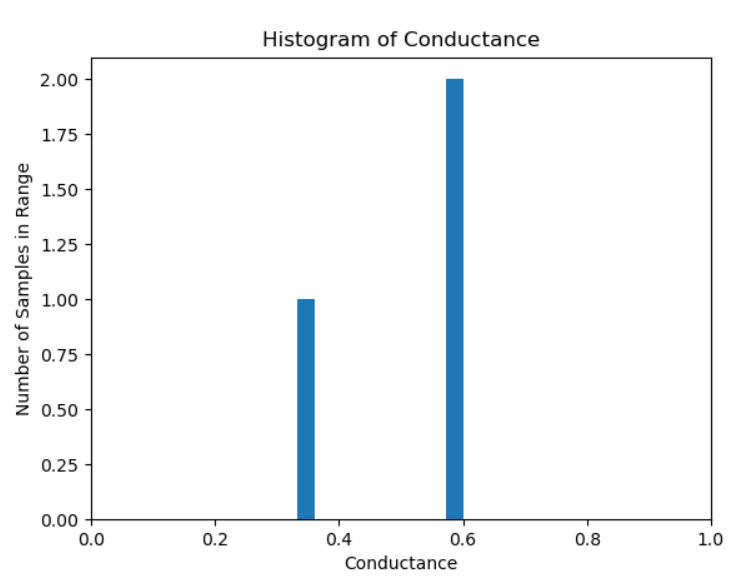
Volume(S3)=5

Volume(T2)= 9

**Cond2=3/5=0.6**

**Code results**

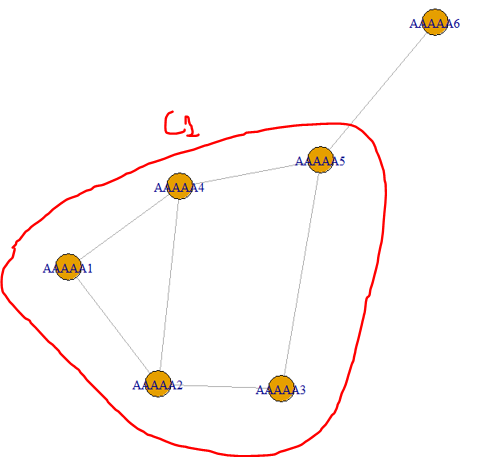




*Σημείωση: σε περίπτωση που έχουμε στο annotation file κόμβους που δεν υπάρχουν στο δίκτυο μας, τότε το volume τους δεν μπορεί να υπολογιστεί γιατί δεν ξέρουμε που βρίσκονται μέσα στο δίκτυο.*

Παράδειγμα 2

Έστω πως για τον ίδιο γράφο G έχουμε το παρακάτω clustering:



Cluster1 AAAAA1,AAAAA2,AAAAA4,AAAAA5,AAAAA3

**C1**

S1 = {"AAAAA1","AAAAA2","AAAAA4","AAAAA3","AAAAA5"}

T1= {"AAAAA6"}

Cut\_size=1

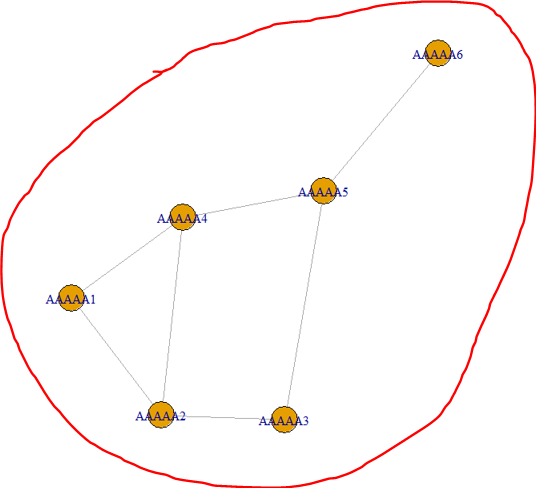
Volume(S1)=13

Volume(T1)= 1

**Cond=1/1 = 1**

Παράδειγμα 3

Για τον ίδιο γράφο G, έστω πως όλοι οι κόμβοι είναι σε ένα Cluster:



Cluster AAAAA1,AAAAA2,AAAAA4,AAAAA5,AAAAA3,AAAAA6

S = {"AAAAA1","AAAAA2","AAAAA4","AAAAA3","AAAAA5","AAAAA6"}}

T= {}

Cut\_size=0

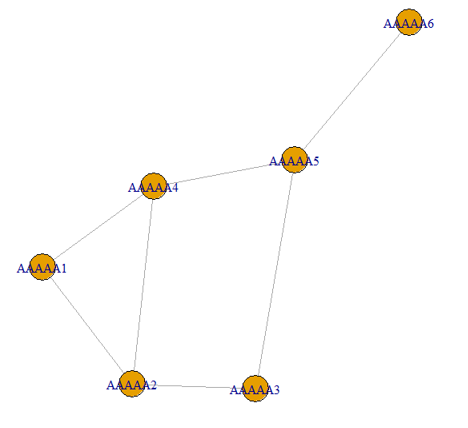
Volume(S1)=14

Volume(T1)= 0

**Cond=0/0 🡪 0**

Παράδειγμα 4

Weighted Graph – G



1

4

32

4

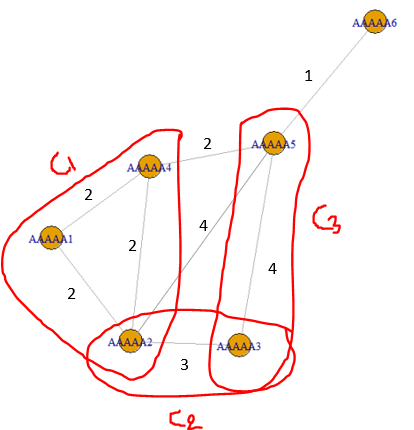
2

2

2

2

Από το Annotation File έχουμε:



**C1**

S1 = {"AAAAA1","AAAAA2","AAAAA4"}

T1= {"AAAAA6","AAAAA3","AAAAA5"}

Cut\_size1=2+3+4=9

Volume(S1)=2\*2+4+4+3+6=21

Volume(T1)= 7+11+1=19

**Cond1=9/19=0,4736**

**C2**

S2={"AAAAA3","AAAAA2"}

T2= {"AAAAA6","AAAAA1","AAAAA5","AAAAA4"}

Cut\_size2=2+2+4+4=12

Volume(S2)=4+4+3+3+4=18

Volume(T2)= 4+6+11+1=22

**Cond2=12/18=0,666**

**C3**

S3= {"AAAAA3","AAAAA5"}

T3 = {"AAAAA6","AAAAA1","AAAAA2","AAAAA4"}

Cut\_size3=3+4+2+1=10

Volume(S3)=3+4+1+8+2=18

Volume(T2)= 22

**Cond2=10/18=0,5555**

