Methods

Centralities

We consider a directed, simple network $G(V, E)$ with a set of N nodes V and an ordered set of edges E. A node $v \epsilon V$ denotes a protein and an edge $e(v, u) \in E$ denotes a directed interaction from protein v to protein u. Each edge has been assigned to a signed weight $w_{v,u} \in [-1, 1]$.

The essentiality consensus of a protein in the protein - protein interaction network is most commonly predicted by centrality measures (Jalili et al. 2016). In this work we used the degree, betweenness, weighted betweenness, closeness and information degree centrality. The historically first centrality used for the prediction of the essential proteins is the degree centrality in the influential paper (Jeong et al. 2001) which introduced the **centrality - lethality rule**. The degree centrality (DC) of a node v is defined as :

$$DC(v) = deg(v), \tag{1}$$

where deg(v) is the number of neighbors of node v.

Degree centrality predicts that hubs are more likely to be essential than non - hubs. This is a simplified view because there are essential proteins that are not hubs.

Because the network is signed we can further distinguish the degree to positive and negative degree. Positive degree (PD):

$$PD(v) = deg^{+}(v), \tag{2}$$

where $deg^{+}(v)$ is number of nodes the have positive interactions with node v.

Also because the network is weighted we define Positive Weighted Degree Centrality (PWDC) as:

$$PWDC(v) = \sum_{u}^{N} (w_{v,u}^{+} + w_{u,v}^{+}), \tag{3}$$

where $w^{+}_{v,u}$ are the positive weights from node v to its u neighbors and $w^{+}_{u,v}$ is the reverse.

Another classification of proteins in respect to network topology is to examine whether they are *bottlenecks*. Bottlenecks are the nodes that are located between highly connected clusters and their importance is measured through betweenness centrality (BC) (Freeman 1979; Joy et al. 2005; Yu et al. 2007). Betweenness centrality (BC) of a node v is defined as:

$$BC(v) = \sum_{s \neq t \neq v \in V} \frac{g_{st}(v)}{g_{st}}, \tag{4}$$

where g_{st} is the number of all geodesic directed paths between all pairs of nodes, except pairs with v, and $g_{st}(v)$ is the number of geodesics that pass through node v.

Weighted betweeness centrality (WBC) is defined as :

$$WBC(v) = \sum_{s \neq t \neq v \in V} \frac{g^{w}_{st}(v)}{g^{w}_{st}}, \tag{5}$$

where the geodesic distance is $g^{w}_{st} = min(\sum w_{st})$, that is the minimum distance between nodes s and t is the path with the minimum sum of weights. In this implementation of betweenness, edge weights must be non negative numbers and higher values of weights have negative impact on path distance. So we took the absolute values of edge E weights of G. Note that this is a crude method of handling weights that in our case isn't biologically appropriate but nevertheless we have included it in the analysis for comparison reasons.

Another centrality index we used is closeness centrality (CC) which is defined as :

$$CC(v) = \sum_{v \neq t \in V} \frac{1}{g_{v,t}}, \tag{6}$$

And finally we computed the information centrality (IC) defined as :

$$IC(v) = informationcentrality, \tag{7}$$

The computations of the centralities were performed in R using the igraph package (Csardi and Nepusz 2006) except from the information centrality which was calculated manually.

Decision trees

Decision trees are supervised machine learning tools used to build classification models (Kotsiantis 2013; Quinlan 1986; Kabacoff 2011). We implemented decision trees on the centrality measures mentioned before to test if the integration of centralities provides better results than single centrality indices for the prediction of essential proteins. We used three algorithms, the algorithm in the rpart package (Therneau, Atkinson, and Ripley 2017), the C4.5 algorithm from the J48 function in RWeka package (Hornik, Buchta, and Zeileis 2009) and the latest algorithm C5.0 from the C5.0 package (Kuhn et al. 2015). After the tree creation each protein was assigned probabilities of essentiality from the 3 different algorithms.

Method comparison

In order to evaluate the performance of each method for predicting essentiality we used 3 methods, the precision - recall, the ROC curve and the Jackknife curve (Holman et al. 2009; Manning, Prabhakar, and Schutze 2008). All these methods use the statistical terms :

- True positives (TP) : essential proteins correctly predicted as essential
- False positives (FP) : nonessential proteins falsely predicted as essential
- True negatives (TN) : nonessential proteins correctly predicted as nonessential
- False negatives (FN) : essential proteins falsely predicted as nonessential

These terms form the confusion matrix of a binary classifier which in our case is essentiality consensus and are used to calculate the following fractions :

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$False\ Positive\ Rate = \frac{FP}{TP + FN} \tag{10}$$

Precision (equation 8) is the ratio of the number of correct predictions to the total number

of predictions. On the other hand recall (equation 9) is the ratio of the number of correct predictions to the total number of possible correct predictions. Using these measures we can plot the Precision - Recall curve through an iterative process. In the first iteration k top ranked proteins (in terms of a variable, i.e degree) are retrieved and the precision and recall are measured. In the next iteration $k+1$ proteins are retrieved, if the protein is nonessential then recall remains the same but precision decreases. If the protein is essential then both recall and precision increase.

False positive rate (equation 10) is the ratio of the wrong predictions to the total number of possible correct predictions. This measure and the recall measure, also called true positive rate, are plotted to create the receiver operating characteristic curve (ROC curve). The ROC curve of a random predictor is the $y = x$ line, any predictor above this line is considered better. The area under ROC curve is called AUC. The ROC curve is plotted with similar way as the precision - recall curve. Both methods were computed using the ROCR package (Sing et al. 2005).

The Jackknife curve was first presented in (Holman et al. 2009) and is a simple alternative method to evaluate predicting tools for binary classifiers. In our case it expresses the relationship between the number of essential proteins in respect to the number of top ranked proteins retrieved based on a variable. This curve is created by incrementally increasing the number of retrieved proteins and the theoretical 100% succesful model is plotted in the $y = x$ line.

Perron - Frobenius decomposition

The topology structure of a network is possible to reflect its function. The work of Frobenius and Perron on matrices can provide some useful insights when implemented on graphs. The following definitions and theorems are well documented with proofs and further details in the books of (Varga 2000) and (Gantmacher 1987). The Perron - Frobenius graph decomposition can illustrate the flow of information in a directed network. If a network is strongly connected as defined in 1 then the information can reach all nodes from all nodes. This means that there is not distinction between nodes or clusters, in terms of information distribution, in the network. In addition, we can explore further the inner structure of a

strongly connected component using the paragraph 5 of theorem 2[1]. After the calculations of the eigenvalues we can evaluate if there are more than one eigenvalues that equal to spectral radius of the component. If this is true, then there exist a cycle in the component and its permuted adjacency matrix takes the form of matrix 12.

If the network is weakly connected, or equally its adjacency matrix is reducible as stated in theorem 1, then we have to find its strongly connected components. The most efficient algorithm to perform this task was developed by (Tarjan 1971) and is included in the *Graph BOOST Library* (Siek, Lee, and Lumsdaine 2001) which has an interface in R (Carey, Long, and Gentleman 2016). By implementing Tarjan's algorithm we identify the network's strongly connected components and single nodes that aren't participating in any strongly connected component. After we can partition the network into tree components:

1. Input: Nodes that have only out edges
2. Processing: Nodes that have incoming and outgoing edges
3. Output: Nodes that have only incoming edges

This structure indicates that information flow is directed in the network.

Definition 1 (Strongly connected) *A directed graph with n nodes is strongly connected if, for any ordered pair* (P_i, P_j) *of nodes, with* $1 \leq i, j \leq n$*, there exist a direct path connecting* P_i *to* P_j*.*

Theorem 1 *An* $n \times n$ *complex matrix A is irreducible if and only if its directed graph* $G(A)$ *is strongly connected.*

Definition 2 (Reducibility) *A* $n \times n$ *complex matrix A, is reducible if there exists a* $n \times n$ *permutation* [2] *matrix such that A takes an upper triangular form:*

$$PAP^T = \begin{pmatrix} B & C \\ 0 & D \end{pmatrix}, \tag{11}$$

where B and D are square matrices. If there isn't such permutation then A is irreducible. In case A is reducible and B or D are also reducible then they are further permutated to components. This process is repeated for as many times needed for all the upper triangular

[1] Theorem 2 is the famous theorem that was proved independently from Perron in 1907 for positive matrices and from Frobenious in 1912 for non-negative matrices.

[2] Permutation matrix is a square matrix that has one entry unity in each row and column and zeros elsewhere.

components of A to be irreducible.

Theorem 2 (Frobenius, 1912) *When A is a square and nonnegative matrix then :*

1. *A has a positive real eigenvalue equal to its spectral radius, r.*
2. *To r there corresponds an eigenvector $x > 0$.*
3. *r increases when any entry of A increases*
4. *r is a simple eigenvalue of A*
5. *if A has has h eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_h$ equal to its spectral radius r ($|\lambda_1| = |\lambda_2| = \cdots = |\lambda_h| = r$) and $h > 0$, then A can be permuted to the following "cyclic" form:*

$$PAP^T = \begin{pmatrix} O & A_{1,2} & O & \cdots & O \\ O & O & A_{2,3} & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \cdots & A_{h-1,h} \\ A_{h,1} & O & O & \cdots & O \end{pmatrix}, \tag{12}$$

where there are square blocks along the main diagonal.

Definition 3 (Primitive matrix) *If a irreducible matrix $A \geq 0$ has h eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_h$ equal to its spectral radius r ($|\lambda_1| = |\lambda_2| = \cdots = |\lambda_h| = r$), then A is called **primitive** if $h = 1$ and **imprimitive** if $h > 1$. In the latter case h is called index of impimitivity of A.*

Enrichment analyses

We performed gene ontology singular enrichment analysis in order to decipher the biological processes that are over-represented in our protein set (Ashburner et al. 2000; Rhee et al. 2008). We used R bioconductor packages AnnotationDbi (Pagès et al. 2017) and org.Dm.eg.db (Carlson 2016b) for *D.melanogaster's* protein ID conversion, GO.db (Carlson 2016a) for protein ID mapping on gene ontology terms and topGO (Alexa and Rahnenfuhrer 2016) to facilitate Fisher's exact test for over-representation of biological process terms. Fisher's exact test uses a background distribution of GO terms and occurences that is compared with a specific test. In our case, we used all protein IDs of the signed network of

D.melanogaster (Vinayagam et al. 2014) as a background to test a subset of this network with essential interacting proteins (Figure ??). From the statistical test we obtained the biological process terms associated with a p-value, a bonferroni correction and FDR. We choose to use the simple p-value at a=0.5 significance level.

Singular enrichment analysis results in a long format table with one column representing the statistically significant GO terms and another column with the protein IDs. This can be considered as a bipartite network with the 2 sets of nodes being GO terms and the protein IDs belonging to them. By projecting the bipartite network to the one-mode network of GO terms we investigate the functional relationships between GO terms. This analysis is called functional enrichment analysis.

Modular essentiality

Each protein complex has many proteins and each protein can participate in many complexes. How are the essential proteins distributed amongst complexes? In order to answer this question we have to do a statistical test with the hypothesis claiming that the distribution of essential proteins in complexes is random. The null distribution was created using the bootstrap procedure. We performed sampling with replacement to the essentiality consensus of the proteins of complexes for 1000 rounds using the sample() function of base R. That way complexes had always the same size. After we calculated the essentiality fraction (EC) of a complex c_{i} which is defined as:

$$EC(c_i) = \frac{number\ of\ essential\ proteins\ in\ c_i}{total\ proteins\ of\ c_i} \in [0, 1] \tag{13}$$

$EC(c_i)$ was calculated for the original data and for each one of the 1000 permutations. Then we sorted the complexes in 5 equally sized bins according to their essentiality fraction. Afterwards, for each bin of the original data and the 1000 permutations we counted the included complexes. So for each bin we had a null distribution for hypothesis testing and p-value calculation. Next we calculated the mean number of complexes in each bin of the permutations in order to compare the expected with the observed number of complexes. The comparison was made with the log ratio:

$$Log - ratio(bin(EC)) = log_2(\frac{number\ of\ complexes \in bin(EC)}{mean\ estimated\ number\ of\ complexes \in bin(EC)}) \quad (14)$$

Tools

All the calculations and analyses were done in R (R Core Team 2016) using the R Studio (RStudio Team 2016) interface. Data handling and manipulation were perfomed with the packages dplyr (Wickham and Francois 2016), tidyr (Wickham 2017) and readr (Wickham, Hester, and Francois 2017). Data visualization was done with the packages ggplot2 (Wickham 2009) and ggraph (Pedersen 2017) and graphic design of Figures **??** and **??** was done with AUTODESK® GRAPHIC application. In addition, all scripts were written in rmarkdown (Allaire et al. 2017) with text alongside the code so all results are easily reproducible (Peng 2011; Piccolo and Frampton 2016). The machine used is a late 2013 model Macbook Pro with 13" retina screen, 2.4GHr Intel Core i5 processor, 8GB RAM memory and macOS Sierra operating system. The thesis was conducted in R Studio using rmarkdon and LaTeX.

Alexa, Adrian, and Jorg Rahnenfuhrer. 2016. *topGO: Enrichment Analysis for Gene Ontology*.

Allaire, J J, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, Aron Atkins, Rob Hyndman, and Ruben Arslan. 2017. *rmarkdown: Dynamic Documents for R*. https://cran.r-project.org/package=rmarkdown.

Ashburner, Michael, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, et al. 2000. "Gene Ontology: tool for the unification of biology." *Nat. Genet.* 25 (1): 25–29. doi:10.1038/75556.

Carey, Vince, Li Long, and R Gentleman. 2016. *RBGL: An interface to the BOOST graph library*. http://www.bioconductor.org.

Carlson, Marc. 2016a. *GO.db: A set of annotation maps describing the entire Gene*

Ontology.

———. 2016b. *org.Dm.eg.db Genome wide annotation for Fly*.

Csardi, Gabor, and Tamas Nepusz. 2006. "The igraph software package for complex network research." *InterJournal* Complex Sy: 1695. http://igraph.org.

Freeman, Linton C. 1979. "Centrality in social networks conceptual clarification." *Soc. Networks* 1 (3): 215–39. doi:10.1016/0378-8733(78)90021-7.

Gantmacher, F.R. 1987. *The Theory of Matrices vol 2*. 2nd ed. AMS Chelsea Publishing. doi:10.1007/978-3-642-99234-6.

Holman, Alexander G, Paul J Davis, Jeremy M Foster, Clotilde KS Carlow, and Sanjay Kumar. 2009. "Computational prediction of essential genes in an unculturable endosymbiotic bacterium, Wolbachia of Brugia malayi." *BMC Microbiol.* 9 (1): 243. doi:10.1186/1471-2180-9-243.

Hornik, Kurt, Christian Buchta, and Achim Zeileis. 2009. "Open-Source Machine Learning: {R} Meets {Weka}." *Comput. Stat.* 24 (2): 225–32. doi:10.1007/s00180-008-0119-7.

Jalili, Mahdi, Ali Salehzadeh-Yazdi, Shailendra Gupta, Olaf Wolkenhauer, Marjan Yaghmaie, Osbaldo Resendis-Antonio, and Kamran Alimoghaddam. 2016. "Evolution of Centrality Measurements for the Detection of Essential Proteins in Biological Networks." *Front. Physiol.* 7 (August): 375. doi:10.3389/fphys.2016.00375.

Jeong, H, S P Mason, a L Barabási, and Z N Oltvai. 2001. "Lethality and centrality in protein networks." *Nature* 411 (6833): 41–42. doi:10.1038/35075138.

Joy, Maliackal Poulo, Amy Brock, Donald E. Ingber, and Sui Huang. 2005. "High-betweenness proteins in the yeast protein interaction network." *J. Biomed. Biotechnol.* 2005 (2): 96–103. doi:10.1155/JBB.2005.96.

Kabacoff, Robert I. 2011. *R in Action : Data analysis and graphics with R*.

Kotsiantis, S. B. 2013. "Decision trees: A recent overview." *Artif. Intell. Rev.* 39 (4): 261–83. doi:10.1007/s10462-011-9272-4.

Kuhn, Max, Steve Weston, Nathan Coulter, and Mark Culp. C code for C5.0 by R. Quinlan. 2015. *C50: C5.0 Decision Trees and Rule-Based Models*. https://cran.r-project.org/

package=C50.

Manning, Christopher D., Raghavan Prabhakar, and Hinrich Schutze. 2008. *Introduction to Information Retrieval*. 1st ed. Vol. 1. Cambridge: Cambridge University Press 2008. doi:10.1017/CBO9781107415324.004.

Pagès, Hervé, Marc Carlson, Seth Falcon, and Nianhua Li. 2017. *AnnotationDbi: Annotation Database Interface*.

Pedersen, Thomas Lin. 2017. *ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*. https://cran.r-project.org/package=ggraph.

Peng, Roger D. 2011. "Reproducible Research in Computational Science." *Science (80-.).* 334: 1226–7. doi:10.1126/science.1213847.

Piccolo, Stephen R., and Michael B. Frampton. 2016. "Tools and techniques for computational reproducibility." *Gigascience* 5 (1). GigaScience: 30. doi:10.1186/s13742-016-0135-4.

Quinlan, J. R. 1986. "Induction of Decision Trees." *Mach. Learn.* 1 (1): 81–106. doi:10.1023/A:1022643204877.

R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.r-project.org/.

Rhee, Seung Yon, Valerie Wood, Kara Dolinski, and Sorin Draghici. 2008. "Use and misuse of the gene ontology annotations." *Nat. Rev. Genet.* 9 (7): 509–15. doi:10.1038/nrg2363.

RStudio Team. 2016. *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc. http://www.rstudio.com/.

Siek, JG, LQ Lee, and Andrew Lumsdaine. 2001. *The Boost Graph Library: User Guide and Reference Manual*. Boston, MA: Pearson Education.

Sing, T, O Sander, N Beerenwinkel, and T Lengauer. 2005. "ROCR: visualizing classifier performance in R." *Bioinformatics* 21 (20): 7881. http://rocr.bioinf.mpi-sb.mpg.de.

Tarjan, Robert. 1971. "Depth-first search and linear graph algorithms." *12th Annu. Symp. Switch. Autom. Theory (Swat 1971)* 1 (2): 146–60. doi:10.1109/SWAT.1971.10.

Therneau, Terry, Beth Atkinson, and Brian Ripley. 2017. *rpart: Recursive Partitioning and Regression Trees*. https://cran.r-project.org/package=rpart.

Varga, Richard S. 2000. *Matrix Iterative Analysis*. Edited by H. Yserentant, R. Bank, R.L.

Graham, J. Stoer, and R. Varga. 2nd editio. Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-05156-2.

Vinayagam, Arunachalam, Jonathan Zirin, Charles Roesel, Yanhui Hu, Bahar Yilmazel, Anastasia A. Samsonova, Ralph A. Neumüller, Stephanie E. Mohr, and Norbert Perrimon. 2014. "Integrating protein-protein interaction networks with phenotypes reveals signs of interactions." *Nat Methods* 11 (1): 94–99. doi:doi:10.1038/nmeth.2733.

Wickham, Hadley. 2009. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://ggplot2.org.

———. 2017. *tidyr Easily Tidy Data with 'spread()' and 'gather()' Functions*. https://cran.r-project.org/package=tidyr.

Wickham, Hadley, and Romain Francois. 2016. *dplyr: A Grammar of Data Manipulation*. https://cran.r-project.org/package=dplyr.

Wickham, Hadley, Jim Hester, and Romain Francois. 2017. *readr: Read Rectangular Text Data*. https://cran.r-project.org/package=readr.

Yu, Haiyuan, Philip M. Kim, Emmett Sprecher, Valery Trifonov, and Mark Gerstein. 2007. "The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics." *PLoS Comput. Biol.* 3 (4): 713–20. doi:10.1371/journal.pcbi.0030059.