Supplementary materials

Giulia Menichetti[1], Ginestra Bianconi[2], Gastone Castellani[1], Enrico Giampieri[1] and Daniel Remondini[1]

[1]Department of Physics and Astronomy and INFN, Bologna University, Viale B. Pichat 6/2 40127 Bologna, Italy
[2]School of Mathematical Sciences, Queen Mary University of London, London E1 4NS, United Kingdom

1 Comparison of entropy measures: example toy model

We recall here the entropy-like measure $\{S_F^i\}$ proposed in [1] and we compare it to our network entropy measure S. The results presented in [1] are substantially different from our achievements and the reason lies in the conceptual foundation of the two measures. We clarify their relation thank to a simple toy model.
The "entropy of information flux distribution" $\{S_F^i\}$ is a single-node entropy measure, i.e. $\{S_F^i\}$ is a collection of entropy values, one for each node of the designed network ($i = 1, ...N$ where N is the total number of nodes). This measure behaves similarly to the local disparity-heterogeneity index (or participation ratio PR) of weighted networks, and is defined as follows:

1. compute the correlation matrix $\{c_{ij}\}$ (R Pearson correlation coefficient)
2. Rescale the correlation values c_{ij} between 0 and 1, i.e.
$$d_{ij} = (1 + c_{ij})/2 \tag{1}$$
3. Compute the Hadamard product between the designed adjacency matrix $\{a_{ij}\}$ and the distance matrix $\{d_{ij}\}$
4. For any given node i with neighbors j we can then assign a "probability distribution" as follows
$$p_{ij} = \frac{d_{ij}}{\sum_j d_{ij}} \tag{2}$$
5. Entropy follows as
$$S_F^i = -\frac{1}{\log k_i} \sum_{j \in \mathcal{N}(i)} p_{ij} \log p_{ij} \tag{3}$$
6. At the end we obtain, for each pool of samples, a vector of entropy values $\{S_F^i\}$.

The proposed entropy S_F quantifies the amount of randomness/disorder of the local flux distribution surrounding any given node i. When the flux is constrained along one direction (i.e. only one component of p_{ij} is nonzero for a give node) $S_F^i = 0$, while when the flux is equally distributed among all neighbors ($p_{ij} = 1/k_i$) $S_F^i = 1$.

Our network entropy S differs for many reasons:

1. We associate a unique scalar entropy value S for each sample, based on the whole adjacency matrix (PPI) and on single-sample gene expression profile. We then can perform statistical analysis based on the entropy values of a set of samples (e.g. metastatic samples). The S_F measure instead generates an entropy value for each node (gene), that is based on all the samples taken together (in order to calculate correlation matrix). In this approach is is thus not possible to separate informations regarding single samples.

2. Our measure S is properly network-like, i.e. it refers to the system as a whole, not divisible in its single components (nodes/genes).

3. S can be considered as the logarithm of the number of "typical" networks compatible with some features, as stated by the principle of Statistical Mechanics of ensembles, based on the assumption that each real network (describing a single sample) is just an instance of all the possible networks (an *ensemble*) satisfying the same characteristics, namely the constraints on degree sequence and on weight distribution. The S_F measure represents instead a "local disorder" , or dishomogeneity index of the neighbours of each single node (in a Shannon entropy sense).

4. The related single node entropy $\{S_i\}$ is a vector characterizing again a single sample, and is not based on the whole sample dataset. This entropy measure is formally similar to $\{S_F^i\}$, i.e. it is Shannon-like and it is related to a probability distribution for a given node i. Anyway our probability distribution covers all the possible nodes $(N-1)$ because is related to the link probability distribution $\{p_{ij}\}$. The link probability distribution is obtained thanks to the maximization of S with some constraints. The link weights used to compute S_i tell us how likely is to observe a given link between node i and node j once we draw a network from the ensemble.

5. Having a vector $\{S_i\}$ for each sample allows a statistical analysis of different conditions at single-gene and single-sample level. Moreover, our analysis can be extended to multiple levels, i.e. performing an enrichment study for the KEGG pathways onto which the genes are annotated.

1.0.1 Toy model

We show now a simple example remarking the main differences between the two entropy measures. We consider a network of 400 nodes composed by 4 main clusters (see Fig. 1), each one a Poisson random network with average degree 10 and number of nodes $N_c = 100$.
Each node can be associated with different features, i.e. a temporal signal or its gene expression value through different samples. Once defined the profiles for all the nodes we can easily compute the R Pearson correlation coefficient for

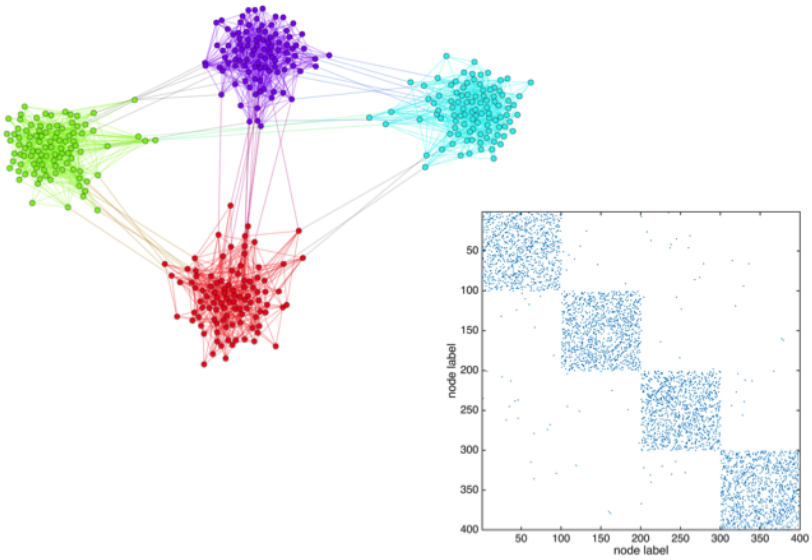

Figure 1: Toy model: network visualization with 4 main clusters (on the left) and its related adjacency matrix (on the right).

each couple. In this toy model each node i is related to a temporal signal. We consider two scenarios:

Case 1 Nodes belonging to the same cluster have similar temporal behaviours, i.e. they are synchronized.

Case 2 Each node has its own specific temporal signal, not related to its neighborhood

For case 1 we randomly choose one of the clusters, and we associate a series $x_i(t)$ of N_T time points with each of its nodes, i.e.

$$x_i(t) = A_i \sin(t) + \varepsilon \qquad t = 1, ..., N_T \tag{4}$$

where A_i is the amplitude of the signal, a random integer between 1 and 5, and ε is a noise variable, drawn from a uniform continuos distribution between 0 and 1. For the other three clusters, we obtain three time series from the original series by permuting the time ordering of the points, so to have correlation only inside each cluster and not between them. For case 2, we generated a unique $400 \times N_T$ matrix following Eq. 4 for each row, and we randomly permuted the values each row in order to have the same values but completely uncorrelated series between all nodes.

For case 1 and case 2 we computed the Pearson correlation matrix, and we rescaled it accordingly to Eq. 1. The rescaled correlation matrices are shown in Fig. 2.

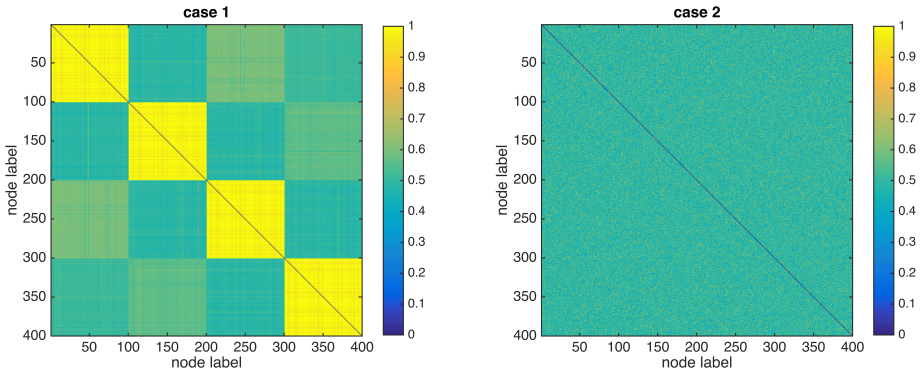

Figure 2: Rescaled correlation matrices for case 1 and 2. In case 1 the syncronization of nodes belonging to the same cluster is highlighted by the yellow color related to high values of correlation.

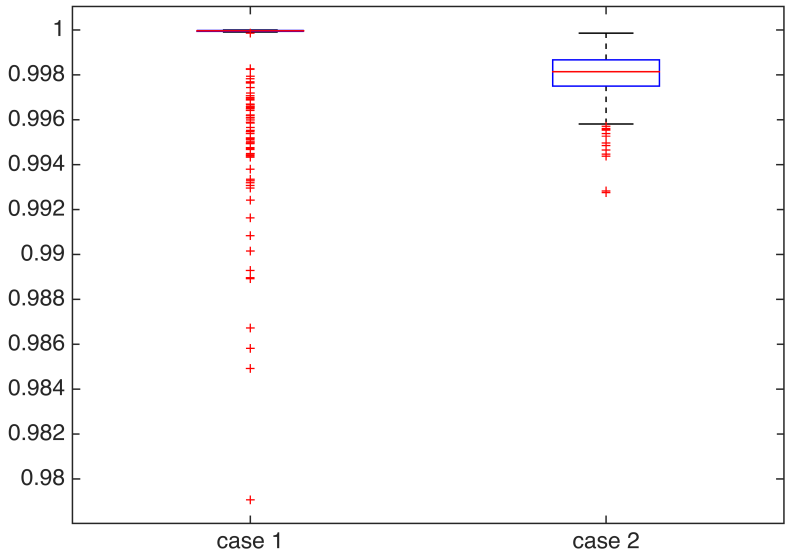

Figure 3: Box plot comparing the entropy of information flux distribution $\{S_F^i\}$ for case 1 and case 2. The median value is respectively 1 for case 1 and 0.9982 for case 2, and they differ significantly ($P < 10^{-20}$).

Once defined the two matrices we can easily measure $\{S_F^i\}$ in both situations, using Eq. 3. The two entropy distributions are displayed in Fig 3. The median value is respectively 1 for case 1 and 0.9982 for case 2, and they differ significantly ($P < 10^{-20}$). The entropy distribution $\{S_F^i\}$ seems to quantify in case 1 a higher uniformity of correlation values in the neighborhood of nodes, independently on the effective value of such correlation.
Even if the background is different from our application of S entropy in the

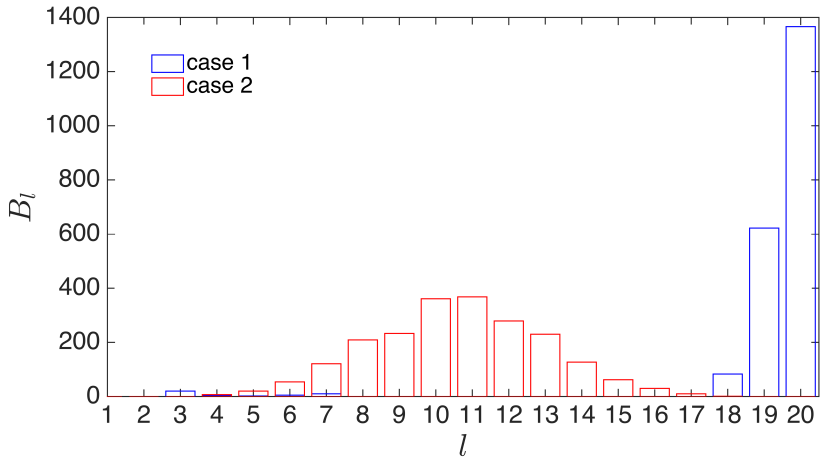

Figure 4: Histogram of number of links B_l per bin l in case 1 and 2, used to calculate S values for the "spatial network ensemble". In both situations we use a linear binning with 20 bins. Since for the calculation of the entropy constraints only the shape of the bin distribution is relevant, and not the value of the bin centroids, we show the bin plots with arbitrary scale on the x axis (i.e. bins numbered from 1 to 20).

paper, we can test how the entropy of network ensembles behaves with these two datasets, by using the definition of "spatial ensemble" as described in the main text. The constraints for the toy model are then the degree sequence of the network (the same in both case 1 and case 2) and the binned distribution of the number of links belonging to each correlation bin (specific for each case). We defined 20 bins in both case 1 and case 2 (see Fig. 4) in order to have exactly the same number of constraints. These constraints are fixed on average over the ensemble of networks, selected by the maximization of S with the given constraints. The value of S defines the logarithm of the number of "typical" networks in the ensemble, i.e. it "counts" the number of networks compatible with the observed degree sequence and correlation distribution of the links. Using the previous set of signals for case 1 and case 2, the network entropy values are respectively $6.7718 \cdot 10^3$ and $9.5644 \cdot 10^3$, meaning that the number of networks compatible with the constraints of case 1 is smaller than case 2.
The results of the two methods thus have an opposite trend, with a different interpretation as stated above. Further on, we validated this scenario computing 50 different realizations of the signal set, both for case 1 and case 2. The results for the entropy of flux distribution are shown in Fig. 5 where we plotted the distribution of the medians. The results for the network entropy measure are displayed in Fig. 6.

2 Algorithms

In this section we examine in depth the algorithms that we used and that we attach as supplementary files: We provide a Matlab implementation for

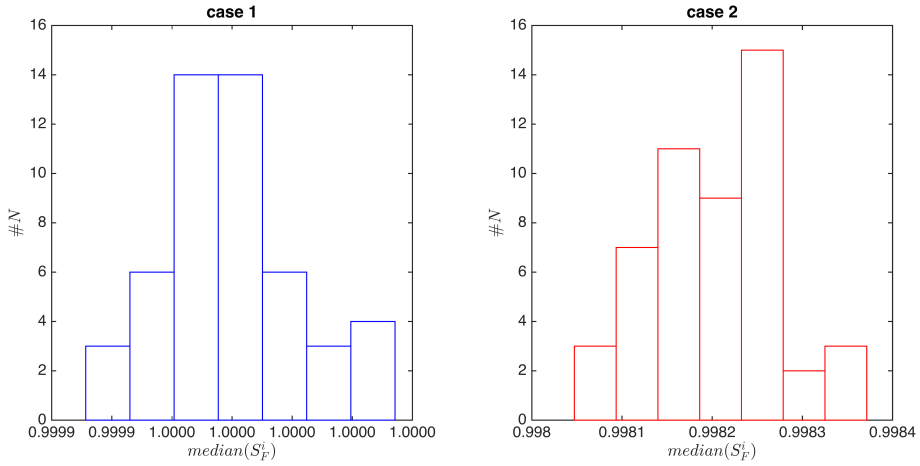

Figure 5: Distribution of the median values for $\{S_F^i\}$ related to 50 different realizations of the signal set for case 1 and case 2.

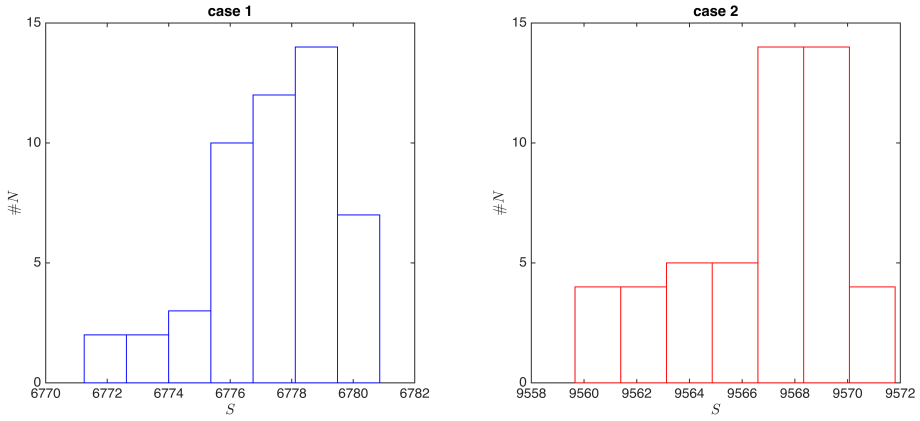

Figure 6: Distribution of network entropy values $\{S\}$ related to 50 different realizations of the signal set for case 1 and case 2.

the calculation of network entropy for the spatial ensemble (*spatial.m*) and the configuration ensemble (*configuration.m*), also implemented in a parallel computing version requiring the Matlab Parallel Toolbox (*spatialp.m* and *configurationp.m* respectively). The python code (*networkentropy.py*) is a small library that gathers together the different ensemblecalculations as functions, only in a parallelized version.
In the different scripts the algorithm is implemented with the same iterative approach, i.e. starting from some chosen conditions for the Lagrangian multipliers the algorithm loops until it reaches convergence (the convergence is reached whenever the maximum relative error affecting the components of the Lagrangian multipliers is $< 10^{-5}$, a tunable parameter inside the scripts, that for our applications was set to this value since it sufficed to find significant result). Different random starting guesses do not affect substantially the final value of

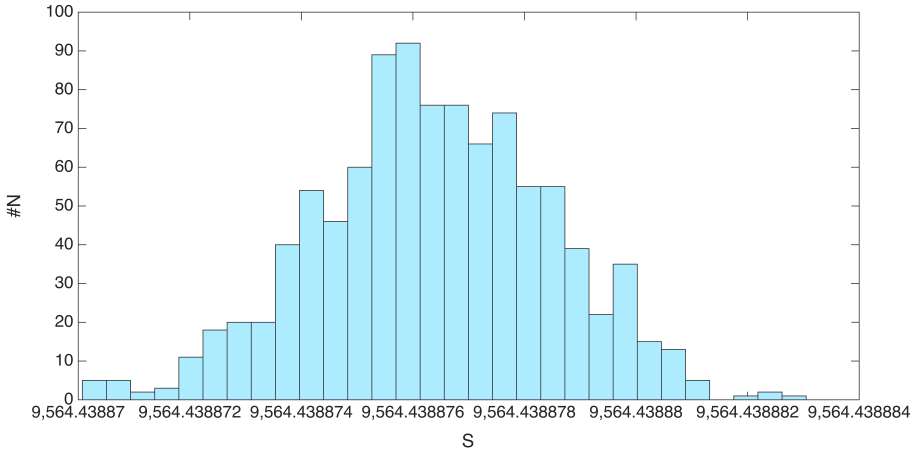

Figure 7: Test for the uniqueness of the network entropy solution in case 2. We considered 1000 different starting Lagrangian multipliers, drawn from an uniform random distribution between 0 and 1.

network entropy. As shown in Fig. 7, we tested 1000 different vectors of starting conditions for the Lagrangian multipliers related to case 2 in the previous toy model. In particular, we drew the values from an uniform random distribution between 0 and 1, but this is not relevant for the final value of network entropy. We remark that the uniqueness of the final solution (i.e. the p_{ij} values and the resulting S) is guaranteed from problems of constrained entropy maximization whenever the constraints are linear functions of the variables to be maximized, as it is in our case (see [2] for details). Moreover, we studied the time complexity of our algorithms. In Fig. 8 we compared a single run of each different implementation (configuration ensemble) over a set of Poisson random graphs with fixed average degree $\langle k \rangle = 10$ and different number of nodes. Time increases as a function of network size as a power law $t = a \cdot N^b$, with the fitted exponents b for the power laws, for scalar Matlab and parallelized Python scripts respectively, equal to 2.62 ± 0.02 (95% C.I. and $R^2 = 0.9997$) and 2.43 ± 0.04 (95% C.I. and $R^2 = 0.9989$). The parallel version of the Matlab script asymptotically shows a similar behavior. The three versions of the same algorithm seem to have a common power law relation with exponent ≈ 2.5 between time and network size, but different multiplicative constants a (affected by parallelization).

3 Sample Discrimination by Network Entropy

We have evaluated the discrimination power of the network entropy for each dataset. The classification has been performed with a Random forest with 100 randomized binary trees [3], with a leave-one-out cross-validation procedure. The classification performances are consistent if we consider a number of trees greater then 10.

Starting from the groups shown in Tab. 1, the resulting classification performances are displayed in Tab. 2. The specificity of the method (its ability to correctly classify a negative case) is consistently high (all values are ≥ 0.8),

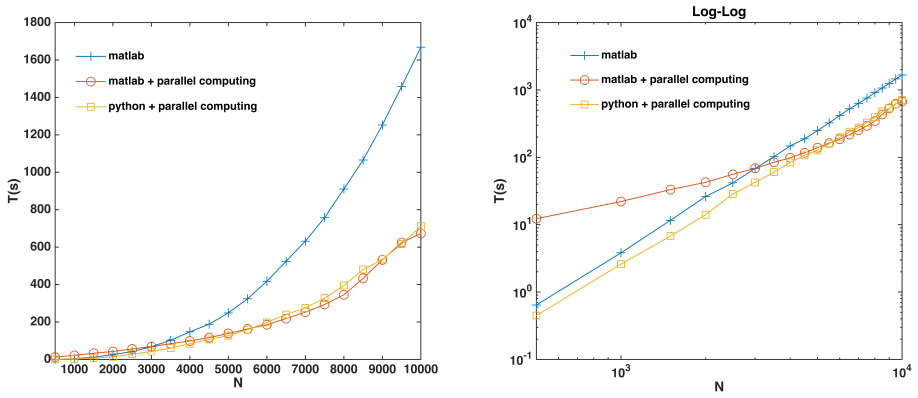

Figure 8: Plot and log-log plot of the time performances for a single run of each different implementation (configuration ensemble) over a set of Poisson random graphs with fixed average degree $\langle k \rangle = 10$ and different number of nodes.

	Ageing	Colon	Ewing	Met	Rel
negative	5	8	30	97	179
positive	5	15	7	28	107

Table 1: We consider as "negative" classes (for Ageing, Colon, Ewing, Met and Rel databases respectively) the following: "A", "normal","non-met", "non-met" and "non-rel", while "positive" classes are "E", "crc","met", "met" and "rel" (in the same order).

while the sensitivity (the ability to correctly classify a positive case) shows larger variations (from 0.87 to 0.21).

	Ageing	Colon	Ewing	Met	Rel
true negative	4	7	24	80	122
false positive	1	1	6	17	57
true positive	4	13	2	6	47
false negative	1	2	5	22	60
sensitivity	0.80	0.87	0.29	0.21	0.44
specificity	0.80	0.88	0.80	0.82	0.68

Table 2: Classification performances

The same classification has been tested with different classification algorithms (data not shown): Naive Bayes, Logistic Regression and Linear Support Vector Machine. The results were consistent with this analysis both for the specificity and the sensitivity values.

We guess that, with enough data available, Network Entropy could prove to be an interesting measure even for diagnostic purposes.

4 Pathway relations

For each dataset (Colon, Ewing, Met, Rel, Ageing), given the single-node entropy values $\{S_i\}$ for each sample, we checked by a nonparametric Wilcoxon rank sum test for significant differences between the groups ($P < 0.05$). Moreover, considering the a priori biological knowledge available from the KEGG database, we performed a functional enrichment analysis of KEGG pathways based on the hypergeometric distribution (i.e. counting the number of genes with significant differences in single-node entropy for a particular pathway, given the total number of significant variations in the whole network). A $P < 0.05$ value was considered for significant enrichment.

The list of all 191 KEGG pathways, divided into 42 metapathways and 6 functional groups, are shown in Supplementary Table 1 (xlsx file) The list of all P values for all the used datasets, for single nodes and for KEGG pathways, are shown in Supplementary Table 2 (xlsx file).

In the following, for each data set we show the network of significant KEGG pathways (selected by the previous mentioned hypergeometric test) that are linked if they share at least one significant gene (in terms of Wilcoxon test). The significant paths related to "Human Diseases" are not shown. Moreover, we chose to display only those pathways whose size (i.e. number of genes) is ≥ 5. Connected components are marked with a different color, only for display purposes.

5 Supplementary Table 3

In Supplementary Table 3 (xlsx file) we compare the list of KEGG pathways enriched by genes which significantly change their single-node entropy with the analogous list given by significant variations in gene expression profiles. As stated in the main text, for each data-set the pathways selected by the two methods are different, remarking the novel information encoded in our approach. The lists display only those pathways whose size is ≥ 5.

References

[1] Andrew Teschendorff and Simone Severini. Increased entropy of signal transduction in the cancer metastasis phenotype. *BMC Systems Biology*, 4(1):104, 2010.

[2] Steve Pressé, Kingshuk Ghosh, Julian Lee, and Ken A. Dill. Principles of maximum entropy and maximum caliber in statistical physics. *Rev. Mod. Phys.*, 85:1115–1141, Jul 2013.

[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

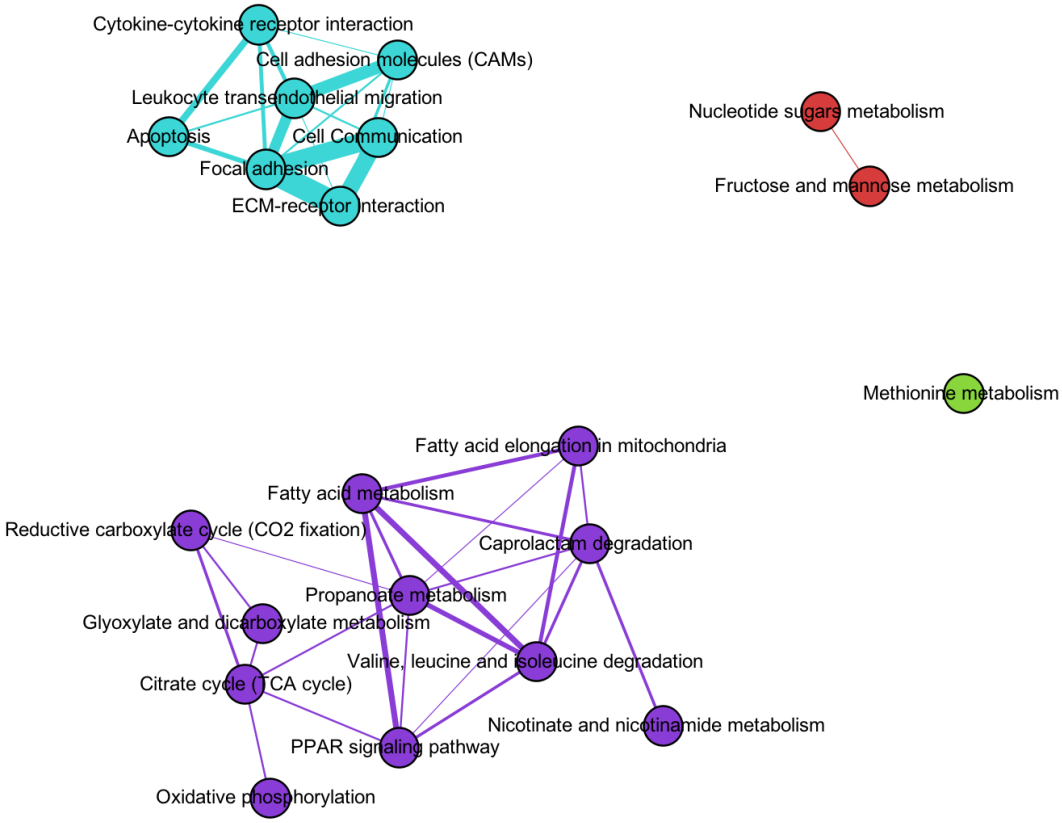

Figure 9: **Colon data set** (minimum weight 1, maximum weight 24, 4 connected components)

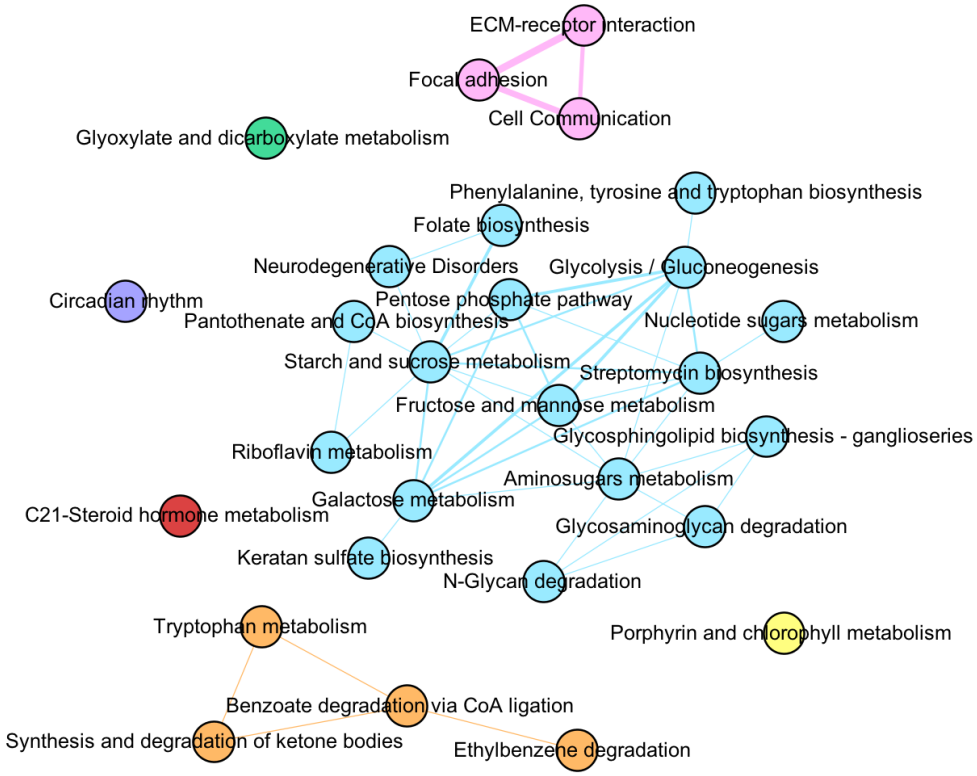

Figure 10: **Ewing data set** (minimum weight 1, maximum weight 7, 7 connected components)

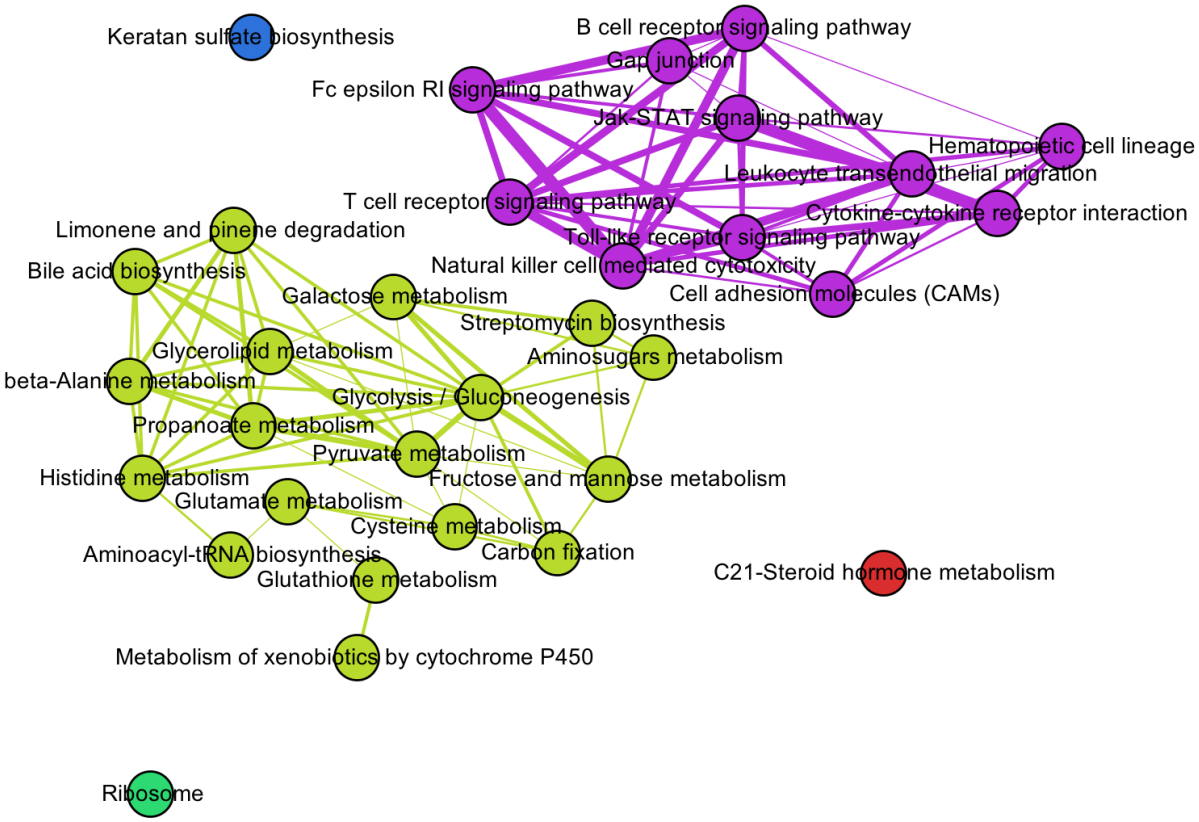

Figure 11: **Met data set** (minimum weight 1, maximum weight 13, 5 connected components)

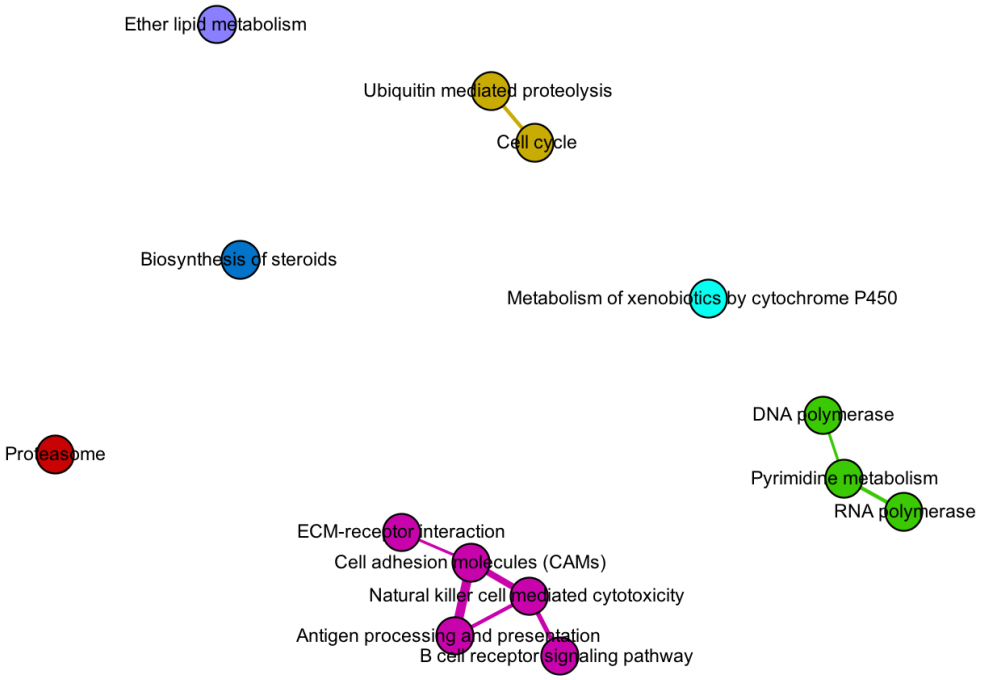

Figure 12: **Rel data set** (minimum weight 1, maximum weight 14, 7 connected components)

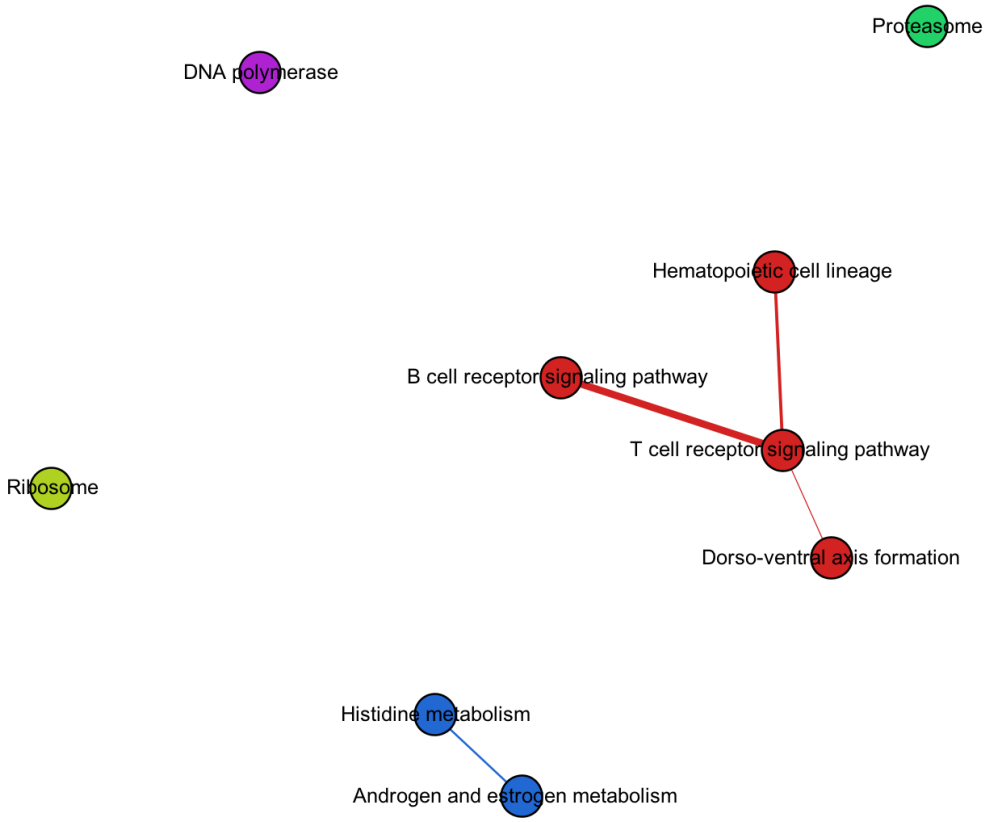

Figure 13: **Ageing data set** (minimum weight 1, maximum weight 11, 5 connected components)