# Tissue-specific evolution of protein coding genes in human and mouse.

Nadezda Kryuchkova[1,2], Marc Robinson-Rechavi[1,2,*]

[1]Department of Ecology and Evolution, University of Lausanne, Lausanne,1015, Switzerland
[2]Swiss Institute of Bioinformatics, Lausanne, 1015, Switzerland

* To whom correspondence should be addressed. Tel: +41 21 692 4220; Fax: +41 21 692 4165; Email: marc.robinson-rechavi@unil.ch

## ABSTRACT

Protein-coding genes evolve at different rates, and the influence of different parameters, from gene size to expression level, has been extensively studied. While in yeast gene expression level is the major causal factor of gene evolutionary rate, the situation is more complex in animals. Here we investigate these relations further, especially taking in account gene expression in different organs as well as indirect correlations between parameters. We used RNA-seq data from two large datasets, covering 22 mouse tissues and 27 human tissues. Over all tissues, evolutionary rate only correlates weakly with levels and breadth of expression. The strongest explanatory factors of strong purifying selection are GC content, expression in many developmental stages, and expression in brain tissues. While the main component of evolutionary rate is purifying selection, we also find tissue-specific patterns for sites under neutral evolution and for positive selection. We observe fast evolution of genes expressed in testis, but also in other tissues, notably liver, which are explained by weak purifying selection rather than by positive selection.

## INTRODUCTION

Understanding the causes of variation in protein sequence evolutionary rates is one of the major aims of molecular evolution, and has even been called a "quest for the universals of protein evolution" (1). Studies in a variety of organisms have reported that protein evolutionary rates correlated with many parameters, structural and functional (2, 3). Most notably, expression level has been shown to be the best predictor of evolutionary rate in yeasts and bacteria: highly expressed proteins are generally more conserved (4–6). In animals and plants, our understanding has been complicated by the fact that genes can have different expression levels depending on tissue or life history stage, and by correlations with multiple other factors such as recombination rate, gene length or compactness, and gene duplications (7–11). In mammals, expression breadth has been suggested to be more important than expression level (12, 13). It has also been suggested that selection against protein misfolding is sufficient to explain covariation of gene expression and evolutionary rate across taxa, including mouse and human (14). This notably explains the slower evolution of brain-expressed genes; the relation with the influence of breadth of expression is unclear. Moreover, it was shown that conserved sites and optimal codons are significantly correlated in many organisms, including mouse and human (14).

These correlations of evolutionary rate to many other parameters, which are themselves correlated (e.g., gene length and GC content), poses problems to determining causality. To disentangle which factors could be determining evolutionary rates a solution is to use partial correlation, taking into account the relationship between gene structure and other parameters when considering the correlations with evolutionary rate (7, 15).

Here we aim to disentangle aspects of protein evolutionary rate and its causal factors in human and mouse. We use partial correlations, taking into account not only different structural parameters, but also different aspects of gene expression (level, tissue specificity), using expression in more than 20 tissues. We also used three measures of protein evolutionary rate estimated from the branch-site model (16): strength of negative selection (value of dN/dS on sites under negative selection); proportion of neutrally evolving sites; and evidence for positive selection (ΔlnL between models with and without positive selection; see Material and Methods). This allows us to distinguish fast evolution due to weak purifying selection from that due to positive selection.

## MATERIAL AND METHODS

We used RNA-seq data for mouse from the ENCODE project (17) and for human from Fagerberg et al. (18). For mouse, the raw reads in FASTQ format obtained from the ENCODE FTP server (ftp://hgdownload.cse.ucsc.edu/goldenPath/mm9/encodeDCC/wgEncodeCshlLongRnaSeq/) were processed with TopHat and Cufflinks (19), using the gene models from Ensembl version 69 (20). For human, processed data from Fagerberg et al. (18) were retrieved from the ArrayExpress database (E-MTAB-1733) (21). 22 tissues for mouse and 27 tissues for human were analyzed, of which 16 are homologous between the two species. Processed RNA-seq expression was further treated as follows (R script in Supplementary Material): multiplied by $10^6$ (to avoid values under 1, which are negative after log transformation); $\log_2$ transformation; and quantile normalization, replacing zero values by $\log_2(1.0001)$.

We used either global parameters of expression: median expression, maximal expression, and specificity among all tissues; or expression in each tissue separately. Expression specificity τ was calculated as follows (22), where *x* is the vector of expression levels over all *n* tissues for a gene:

$$\tau = \frac{\sum_{i=1}^{n}(1-\hat{x_i})}{n-1};\ \hat{x_i} = \frac{x_i}{\max\limits_{1\le i\le n}(x_i)}$$

Values of expression specificity close to zero indicate that a gene is broadly expressed, and close to one that it is specific of one tissue.

Additional analysis was performed on microarray expression data from 22 human and 22 mouse tissues selected from the Bgee database (23), as well as 8 human and 6 mouse tissues from Brawand et al. RNA-seq (24). The corresponding results are presented in Supplementary Materials.

As measures of protein-coding evolutionary rate, we used the estimates from the branch-site model (16) as precomputed in Selectome (25): purifying selection estimated by the dN/dS ratio $\omega_0$ on the proportion of codons under purifying selection (noted "Omega" in the figures), evidence for positive selection estimated by the log-likelihood ratio ΔlnL of $H_1$ to $H_0$ (models with or without positive selection), and the proportion of neutrally evolving sites $p_1$. The evolutionary rate parameters were estimated from the Euteleostomi gene trees on the Murinae branch for mouse and the Homininae branch for human. We also present in Supplementary Materials another estimation of evolutionary rate, using the exon based MI score (26, 27).

For all parameters the longest coding transcript was chosen as a representative of the gene, as the evolutionary rate data were available only for this transcript. Analysis was also redone for mouse using the most expressed transcript; results are presented in Supplementary Materials.

Intron number, intron length, CDS length (coding sequence length) and GC content were taken from Ensembl 69 (20). Essentiality data were manually mapped and curated (Walid Gharib, personal communication) for human from the OMIM database (28) and for mouse from the MGI database (29).

Data of expression at different developmental stages were obtained from Bgee (23). The parameter stage number indicates the number of stages in which the gene was reported as expressed. Mouse development was divided in 10 stages: 1. Zygote 2. Cleavage 3. Blastula 4. Gastrula 5. Neurula 6. Organogenesis 7. Fetus 8. Infant 9. Adolescent 10. Adult; and human in 8 stages: 1. Cleavage 2. Blastula 3. Organogenesis 4. Fetus 5. Infant 6. Adolescent 7. Early adult 8. Late Adult.

Phyletic age was downloaded from the OGEE database (30), as ordinal data. Phyletic age stages used are: 1. Mammalia 2. Chordata 3. Metazoa 4. Fungi/Metazoa 5. Eukaryota 6. Cellular organism.

Correlation between the different parameters was performed in two ways: simple pairwise Spearman correlation and partial Spearman correlation (results for Pearson correlation are also presented in Supplementary Materials). For partial correlation each pair of parameters were compared taking into account all other parameters: first a linear model according to all other parameters for each of the two analyzed parameters was calculated; then the Spearman correlation was calculated on the corresponding residuals. All R code is available as Supplementary Materials.

Partial correlation was used to determine the correlation between two parameters excluding dependencies from other parameters. The principle of the partial correlation can be shown on a toy

example (Supplementary Table S1). As example data for human height and leg length were simulated, so that either a) the length of both legs is calculated depending on height, or b) the length of the left leg is calculated from height, and the length of the right leg is calculated from the length of left leg. With simple correlation the two cases cannot be distinguished, as all three parameters correlate strongly with each other. With partial correlation we can distinguish the two cases: in case a) left leg and right leg length don't correlate with each other if we exclude influence of the height, but in case b) we see a strong correlation between them, as expected, while right leg length no longer correlates with height.

Expression, intron length, intron number, CDS length, $\tau$, $\omega_0$, paralog number were $\log_2$ transformed before calculations. $p_1$ and $\Delta$lnL were normalized by taking the fourth root (31, 32). For parameters containing zeros a small value was added before log transformation, chosen as the minimal non zero value of the parameter (except for RNA-seq, see detailed treatment above). Altogether 9553 protein-coding genes for human and 9485 protein-coding genes for mouse were analyzed.

## RESULTS

We detail here the results of Spearman partial correlation analyses (Table 1); standard Spearman and Pearson as well as partial Pearson correlations are provided in Supplementary Materials. Spearman correlation was preferred as most of the data analyzed are not normally distributed (Supplementary Figure S1), even after transformation, and to avoid a large influence of outliers. It should be noted that parameters that are expected to have strong direct relations remain strongly correlated in the Spearman partial correlation. For example the correlation between coding sequence (CDS) length and intron number, in mouse, is $\rho$=0.683 for partial vs. $\rho$=0.760 for simple correlation, showing that longer genes have more introns. Similarly, partial correlations still show that higher expressed genes are broadly expressed, and that specific genes have lower expression in general. Thus little relevant information is lost, while spurious correlations can be hopefully avoided.

### Evolutionary rate: global influences on selection

Evolutionary rate is represented by three parameters in this study, taken from the branch-site model (see Methods): $\omega_0$ = dN/dS, measures the intensity of purifying selection on the subset of sites determined to be under purifying selection; $p_1$ is the proportion of neutrally evolving sites; and $\Delta$lnL measures the strength of evidence for positive selection.

In both mouse and human, none of the aspects of gene expression yields a strong partial correlation to any feature of evolutionary rate (Table 1; Figure 1). There is a weak correlation of $\omega_0$ to expression specificity $\tau$ in both human and mouse ($\rho$ = 0.085 and 0.067 respectively), confirming that more broadly expressed genes evolve under stronger purifying selection. Purifying selection $\omega_0$ is also negatively correlated to maximum expression, although this is weaker in human, indicating that genes with high expression in at least one tissue have a tendency to evolve under strong purifying selection. More surprisingly, purifying selection $\omega_0$ is positively correlated to median expression. Note that these are partial correlations; without correcting for other parameters, as expected, $\omega_0$ correlates negatively with median expression, i.e., highly expressed genes are under strong purifying selection. It appears

that this negative correlation is driven by the effect of breadth of expression and of maximum expression, with the residual effect actually in the opposite direction.

Evolutionary features of the genes, paralog number and phyletic age, have a stronger partial correlation with $\omega_0$ than expression: older genes, and genes with more paralogs, evolve under stronger purifying selection; again, this is after removing the effect of high levels of expression, as well as the correlation between gene age and number of paralogs. In human, GC content also appears to have a strong influence on $\omega_0$, but much less so in mouse.

It remains that none of these parameters can explain much of the differences in purifying selection. The total variance of $\omega_0$ that they explain (using partial Pearson correlation, as Spearman ρ does not relate directly to variance) is 10.2% for human and 13.8% for mouse, thus leaving more than 85% of the variance unexplained.

The strongest correlation with $\omega_0$ is for $p_1$, the proportion of sides evolving neutrally (Figure 1). This partial correlation is $\rho = 0.748$ in mouse and $\rho = 0.598$ in human; genes under strong purifying selection have a smaller proportion of sides evolving neutrally. This is not due to the way that these parameters are estimated in the branch-site test, since $\omega_0$ is computed on a distinct set of codons from $p_1$. This proportion $p_1$ of neutrally evolving sites is otherwise mostly correlated with evolutionary features (phyletic age, paralog number) in human, and with structural features (intron length, GC content) in mouse, but correlations are weak (all $\rho < 0.09$).

Evidence for positive selection correlates negatively with median expression in both human and mouse (Figure 1), i.e. highly expressed genes are under weaker positive selection ($\rho = -0.105$ and $-0.187$ respectively). It should be noted that this correlation concerns relatively weak evidence for selection, since only 4 human and 23 mouse genes in the dataset have significant support for branch-site positive selection (using the false discovery rate of 10% cut-off of Selectome, see Methods).

**Tissue-specific analysis**

When the correlation between expression level, selective pressure, and other parameters, is analyzed for each tissue separately, there are large differences, notably in the correlation between expression and purifying selection (Figure 2).

In both human and mouse, the strongest correlation with purifying selection $\omega_0$ is for level of expression in the brain, as expected from previous studies with less tissues (12, 14, 33, 34). After correcting for all other effects, the residual correlation is rather weak (ρ between -0.065 and -0.107 depending on species and brain part), but always in the direction of stronger purifying selection on genes with higher brain expression. In human, there are also significant partial correlations for esophagus, prostate, adrenal, colon, and endometrium (Figure 2B). In mouse, there are correlations for all sampled tissues except liver, placenta and testis (Figure 2A); in human the homologous tissues to these three also have among the lowest partial correlations. Interestingly, the only positive partial correlation with $\omega_0$ is for human testis expression, i.e. higher expression in testis correlates with weaker purifying selection.

The strongest correlations with the proportion of neutrally evolving sites are also for brain tissues, in human and in mouse. Again the correlation is negative, indicating less neutral evolution (i.e., more

selection) for more highly expressed genes. There are almost no other tissues with significant partial correlation of expression and $p_1$, although for mouse large intestine the correlation is significantly positive.

Concerning evidence of positive selection, on the other hand, there are significant negative partial correlations for all tissues, meaning that for each tissue genes with higher expression have less evidence of positive selection. Brain tissues again have some of the strongest correlations, although they stand out less than for $\omega_0$ or $p_1$. In both mouse and human the correlation is weakest for testis expression, and also quite weak for placenta.

All these correlations include for each tissue both house-keeping and tissue-specific genes; the former might confuse tissue-specific patterns. Thus we repeated the analysis restricted to tissue-specific genes, defined as $\tau > 0.2$ (Supplementary Figure S2). The global picture is similar, with notably significant negative partial correlations to $\omega_0$ only for expression in human brain and mouse cerebellum, significant negative correlation to $p_1$ only for mouse brain parts, and conversely significant positive correlations to expression in human and mouse testis.

**Gene age and duplication**

As expected, older genes have more paralogs (positive correlation in Figure 1) (35). Tissue specificity has a rather strong positive partial correlation with paralog number, and a significant weak negative correlation with phyletic age was detected; both correlations are stronger in human than in mouse. That means that, correcting notably for the correlation between gene age and paralog number, new genes and genes with more paralogs tend to have more specific expression. While in simple correlation, phyletic age and expression level (median or maximum) have a strong positive correlation (older genes have higher expression), this effect is almost completely lost in the partial correlation, and so is probably spurious.

The phyletic age of the genes correlates negatively with purifying selection but almost no correlation can be seen to neutral evolution or positive selection. This is consistent with previous observations that older genes evolve under stronger purifying selection ((36, 37) but see (38)).

Paralog number correlates negatively with purifying selection in both organisms (-0.064 for mouse and -0.136 for human). This indicates a stronger effect of the biased preservation of duplicates under stronger purifying selection (39–41), than of the effect of faster evolution of duplicated genes (42).

**Gene structure**

Genes with higher GC content have higher expression level, as shown previously (43), although the effect is not very strong in partial correlation. Previous findings that highly expressed genes are shorter were only partly confirmed: there is a strong negative partial correlation between CDS length and maximal expression, but the partial correlation between median expression and CDS length is weakly positive. Curiously, the partial correlation with intron number is opposite, indicating that genes with high maximum expression tend to have more introns than expected given their CDS length.

### Differences between human and mouse, and between datasets

In general correlations in human are slightly weaker than in mouse, but very consistent (Supplementary Figure S3). The strongest difference is between the correlations of GC content and stage number; and of GC content and maximal expression.

There are also noticeable differences between mouse and human in the partial correlations among $\omega_0$, $p_1$ and ΔlnL (evidence for positive selection). In human ΔlnL correlates negatively with $p_1$ and positively with $\omega_0$, indicating that genes with high proportion of neutrally evolving sites and weak purifying selection show little evidence for positive selection. In mouse the correlations are not significant, and in the opposite directions, but the correlation between $\omega_0$ and $p_1$ is much stronger. GC content and paralog number also have stronger correlations to purifying selection in human than in mouse.

We repeated our analyses with large microarray experiments (see Methods, and Supplementary Materials), to control for putative biases in RNA-seq data. There are a few differences, although they do not change our biological conclusions. First, with microarrays tissue-specificity τ appears overall lower, and the correlations between expression parameters (τ, maximal expression, median expression) are stronger. This might be due to the better detection of lowly expressed genes by RNA-seq than by microarray, whereas there seems to be less difference for highly expressed genes (44). Conversely, correlations of expression parameters with all other parameters are much stronger for RNA-seq. The correlation between $\omega_0$ and expression in each tissue separately is stronger with microarrays then with RNA-seq, and significant for all tissues, but the same tissues have the strongest (resp. weakest) correlation between $\omega_0$ and expression with both techniques. Inversely, the evidence for positive selection has almost no significant correlations with expression in single tissues with microarray data.

We also reproduced our analysis using the precomputed "MI" score for most conserved exon (26, 27) instead of the branch-site model $\omega_0$, and all results are similar despite the differences in multiple sequence alignment and in evolutionary model (Supplementary Figure S4): e.g., phyletic age is the strongest correlation to MI and median expression has a weak positive partial correlation.

Finally, we repeated our analysis with the RNA-seq data for human and mouse 6 tissues from Brawand et al. (24); results are extremely similar to those with the large RNA-seq experiments used in our main results (Supplementary Material), with less detail of tissues, and less resolution for τ, due to the smaller sampling.

Overall, our results appear quite robust across species and experimental techniques.

## DISCUSSION

### Technical limitations and generality of observations

We use partial correlations to hopefully detect causal relations. Of note, the lack of partial correlation between two parameters does not mean that they are not correlated in practice, but that the correlation is not causally informative, or insufficiently to be detected.

Our analysis was performed on approximately half of the known protein coding genes (9509 for human and 9471 for mouse), for which evolutionary rate could be computed reliably. While this may introduce some bias, it does not appear to have a large influence, since correlations other than to evolutionary rate are very similar on the other half of the coding genes (Supplementary Material).

**Global study of evolutionary rate**

Our aim is to understand the causes of variation in evolutionary rates among protein-coding genes in mammals. In yeast or bacteria, the major explanatory feature is the relation between the level of gene expression and purifying selection (1–3). In mammals, firstly levels of expression are more complex to define, due to multicellularity and tissue-specificity, and secondly several other features have been reported to correlate as much or more with evolutionary rate, in studies which did not necessarily incorporate all alternative explanations.

In this study, we have focused on the dN/dS ratio, or ω, and distinguished further the three forces which affect this ratio, under the classical assumption that dS is overwhelmingly neutral (although see (45–47)). The intensity of purifying selection is clearly the main component of the overall ω: on average more than 85% of codons are in the purifying selection class of the evolutionary model used. Analyzing separately neutral evolution and purifying and positive selection, we find that (i) these three forces do not affect protein coding genes independently, and (ii) they have different relations to gene expression and to other features. Notably, genes which are under stronger purifying selection have less codons predicted under neutral evolution. Importantly, we computed evolutionary rates on filtered alignments (25), which probably eliminates mostly neutrally evolving sites, thus underestimating $p_1$. Still, it appears that to the best of our knowledge these two forces act in the same direction. The relation is less clear concerning the evidence for positive selection, with opposite correlations in human and mouse. But we are limited by the weak evidence for positive selection on the branches tested, at the human-chimpanzee and mouse-rat divergences. Overall, these relations between forces acting on ω deserve further investigation with more elaborate evolutionary models (e.g. (48, 49)). Despite the limitations of the estimation of positive selection, this is the component of evolutionary rate which has the strongest partial correlation with the level of gene expression, both with the median expression over all tissues, and with expression in brain tissues. This implies that when expression patterns constrain the protein sequence, they also strongly limit adaptation (strong purifying selection and very low positive selection).

So what explains evolutionary rate? The strongest partial correlation of $\omega_0$ is with phyletic age: older genes evolve under stronger purifying selection. While the use of partial correlation allows us to correct for some obvious biases in detecting distant orthologs, such as gene length, we cannot exclude that results be partially caused by the easier detection of orthologs in distant species for proteins with more conserved sequences (37, 38, 50). I.e., genes with weak purifying selection may be reported as younger than they are, because the orthologs were not detected by sequence similarity. We obtain similar results with an exon-based index of sequence conservation, MI (Supplementary Figure S4). Whatever the contributions of methodological bias and biological effect, this correlation is

not very informative about causality, since stronger selection will not be caused by the age of the gene.

The next strongest partial correlation with $\omega_0$ is the GC content of the gene. In mammals, the variation in GC content of genes seems mostly due to GC-biased gene recombination (51), and this in turn has been show to impact estimation of dN/dS (52). But while GC-biased gene recombination is expected to lead to high GC and an overestimation of ω, we find a negative correlation between $\omega_0$ and GC content, consistent with previous observations in Primates (53). Of note, estimating the actual biased recombination rate rather than GC content is limited by the rapid turn-over of recombination hotspots (54), and recombination rate appears to have only a very weak effect on dN/dS in Primates once GC content is taken into account (53). The previously reported relation between dN/dS and intron length seems to be mostly an indirect effect of the strong correlation between GC content and intron length (51, 55).

The significant, although weaker, partial correlation of $\omega_0$ to paralog number is consistent with previous observations that genes under stronger purifying selection are more kept in duplicate (9, 39–41).

The level of gene expression has been reported repeatedly to be the main explanatory variable for dN/dS (4–6, 56, 57), notably in *S. cerevisiae*. Our first observation is that no aspect of expression in human and mouse adult tissues is as strong an explanatory factor for any component of evolutionary rate as what was reported in yeast. Our second observation is that three aspects of expression influence evolutionary rate most strongly: breadth of expression τ; number of developmental stages (Figure 1; Table 1); and expression in brain tissues (Figure 2). The third, surprising, observation is that median expression is positively correlated with $\omega_0$: taking into account other parameters, genes which have higher expression on average are under weaker purifying selection; whereas the correlation with maximal expression is negative, as expected. Thus in mammals the negative correlation between median expression and evolutionary rate appears to be an indirect effect of stronger selection on broadly expressed genes and on genes with high maximal expression in at least one tissue (this is also true if we take the mean instead of median expression, see Supplementary Materials).

We confirm previously reported observations that expression breadth is more important then expression level itself in mammals (13). dN was previously found to be threefold lower in ubiquitous than in tissue-specific genes, while dS did not vary with expression specificity (12). Other studies indicate that genes expressed in few tissues evolve faster then genes expressed in a wide range of tissues (13, 57, 58), or that tissue-specific genes have more evidence for positive selection (59). In mouse, but not human, τ is weakly negatively correlated to evidence for positive selection: broadly expressed genes seem to be more affected by positive selection, *contra* Haygood et al (59). We also notice that tissue-specificity and maximal expression are correlated, i.e. more tissue-specific genes have higher maximum expression in one tissue. Thus these two forces appear to act on different genes: some genes are under strong purifying selection because they are broadly expressed, suggesting an important role of pleiotropy, while other genes are under strong purifying selection because they are highly expressed in few tissues, suggesting an important role of the tissue-specific optimization of protein sequences. Of note, analyzing separately only brain expression relative to

maximal and breadth of expression in other tissues gave similar results, thus brain expression alone is not driving these patterns (not shown).

Other studies have reported that expression level and tissue specificity are less important than gene compactness and essentiality in mammals (10). Liao et al. (10) reported that compact genes evolve faster, but this correlation is very weak in our study. We could not either confirm that highly expressed genes are shorter (11, 43, 60). We have used the longest transcript for each protein coding gene, as evolutionary parameters (ΔlnL, $\omega_0$, $p_1$) were calculated for the transcript. But this might not be the transcript most expressed and used in all tissues (61). We repeated calculations with the most expressed transcript (Supplementary Material), but results were unchanged; we show these results only in supplementary materials, as the estimation of transcript-level expression does not yet appear to be very reliable (62, 63). Finally, we tried to investigate the impact of essentiality, but we found no significant effect (Supplementary Figure S5); we note that we have very low power to test this effect, especially in human.

The largest partial correlations that we observed for components of evolutionary rate are between brain expression level and evidence for positive selection, at -0.203 to -0.188 in mouse (for different brain parts), and -0.168 in human (whole brain). For purifying selection we find weaker but significant partial correlations with brain expression and with the number of stages, between 0.065 and 0.119. And brain tissues also have the strongest partial correlation over expression in tissues for neutral evolution (Figure 2). It has been previously reported that brain expression is a major component of evolutionary rate in mammals and other animals (12, 14, 33, 34), and here we confirm the dominance of this component, even taking other effects into account. Importantly we show that this affects all forces acting on protein evolutionary rate: purifying selection, neutral evolution, and positive selection. Thus the median expression of genes over more than 20 tissues is a poor explanation of protein evolutionary rate, relative to brain expression.

**Tissue specific patterns**

There are striking differences between tissues in the extent of the correlations with structural and evolutionary parameters. As already mentioned, brain tissues present the strongest partial correlations with evolutionary rate; results are consistent when only tissue-specific genes are used. We observe this for the three evolutionary forces estimated. In most comparisons, the correlation is stronger for brain expression than for any global measure of expression. This is consistent with the translational robustness hypothesis, which proposes that highly expressed genes are under stronger pressure to avoid misfolding caused by translational errors, thus these genes are more conserved in evolution (4), and that neural tissues are the most sensitive to protein misfolding (14). This slow evolution of genes expressed in neural tissues has been repeatedly reported (12, 34, 64), especially for the brain (13); it has also been related to higher complexity of biochemical networks in the brain than in other tissues (34).

Fast evolution of genes expressed in testis is also well documented (24, 33, 64), and could be due to lower purifying selection, an excess of young genes and leaky expression, or to positive selection due to sexual conflict. We observe neither a stronger correlation between expression in sexual tissues

and evidence for positive selection, nor a stronger correlation between expression in sexual tissues and the proportion of sites evolving neutrally. What we do observe is that the weakest partial correlation between expression in a tissue and purifying selection is for testis, and that it is also quite weak for placenta, with even a surprising positive correlation between $\omega_0$ and expression in human testis, which remains when only tissue-specific genes are used. This is consistent with the "leaky expression" explanation: being expressed in the testis does not appear to be an indicator of function carried by the protein sequence. Interestingly, expression in testis is negatively correlated with the number of paralogs, significantly so in mouse: genes which are more expressed in testis have less paralogs, after correcting for other effects.

While the strong correlation of $\omega_0$ with expression in the brain, and the weak correlation with expression in testis are expected, we also observe less expected patterns. Most notably, liver expression has the next weakest correlation with $\omega_0$ after testis (and placenta in mouse). Although it was reported before that liver expressed genes are evolving faster (12, 33), it was reported with much fewer tissues, and not highlighted. Liver expression is also positively correlated with the proportion of neutral sites, unlike brain or testis expression, although this is not significant. Interestingly, liver has the strongest correlation of expression with phyletic age, implying that despite low purifying selection, old genes are more expressed in liver. In any case, this outlier position of liver has important practical implications, since liver is often used as a "typical" tissue in studies of gene expression for molecular evolution (e.g., (65–67)).

## CONCLUSION

The main result of our study is that average adult gene expression is quite lowly informative about protein evolutionary rate, while purifying selection on genes highly expressed in the brain and breadth of expression are our best bets for a causal factor explaining evolutionary rates. A practical consequence is that great care should be taken before using expression from other tissues, including widely used ones such as liver, as proxies for the functional importance of mammalian genes.

Finally, all calculations were performed with expression in adult tissues. It is possible that expression in embryonic development be more important for evolutionary constraints in mammals, and this should be explored further.

## ACKNOWLEDGEMENT

We thank Julien Roux for helpful comments on the manuscript. The computations were performed at the Vital-IT Center (http://www.vital-it.ch) for high-performance computing of the SIB Swiss Institute of Bioinformatics.

## FUNDING

This work was supported by the Swiss National Science Foundation [grants number 31003A 133011/1 and 31003A_153341/1] and Etat de Vaud.

## REFERENCES

1. Rocha,E.P.C. (2006) The quest for the universals of protein evolution. *Trends Genet.*, **22**, 412–6.

2. Pál,C., Papp,B. and Lercher,M.J. (2006) An integrated view of protein evolution. *Nat. Rev. Genet.*, **7**, 337–48.

3. Rocha,E.P.C. and Danchin,A. (2004) An analysis of determinants of amino acids substitution rates in bacterial proteins. *Mol. Biol. Evol.*, **21**, 108–16.

4. Drummond,D.A., Bloom,J.D., Adami,C., Wilke,C.O. and Arnold,F.H. (2005) Why highly expressed proteins evolve slowly. *Proc. Natl. Acad. Sci. U. S. A.*, **102**, 14338–43.

5. Pál,C., Papp,B. and Hurst,L.D. (2001) Highly Expressed Genes in Yeast Evolve Slowly. *Genetics*, **158**, 927–931.

6. Wall,D.P., Hirsh,A.E., Fraser,H.B., Kumm,J., Giaever,G., Eisen,M.B. and Feldman,M.W. (2005) Functional genomic analysis of the rates of protein evolution. *Proc. Natl. Acad. Sci. U. S. A.*, **102**, 5483–8.

7. Larracuente,A.M., Sackton,T.B., Greenberg,A.J., Wong,A., Singh,N.D., Sturgill,D., Zhang,Y., Oliver,B. and Clark,A.G. (2008) Evolution of protein-coding genes in Drosophila. *Trends Genet.*, **24**, 114–23.

8. Makino,T., Hokamp,K. and McLysaght,A. (2009) The complex relationship of gene duplication and essentiality. *Trends Genet.*, **25**, 147–52.

9. Yang,L. and Gaut,B.S. (2011) Factors that contribute to variation in evolutionary rate among Arabidopsis genes. *Mol. Biol. Evol.*, **28**, 2359–69.

10. Liao,B.-Y., Scott,N.M. and Zhang,J. (2006) Impacts of gene essentiality, expression pattern, and gene compactness on the evolutionary rate of mammalian proteins. *Mol. Biol. Evol.*, **23**, 2072–80.

11. Li,S.-W., Feng,L. and Niu,D.-K. (2007) Selection for the miniaturization of highly expressed genes. *Biochem. Biophys. Res. Commun.*, **360**, 586–92.

12. Duret,L. and Mouchiroud,D. (2000) Determinants of substitution rates in mammalian genes: expression pattern affects selection intensity but not mutation rate. *Mol. Biol. Evol.*, **17**, 68–74.

13. Park,S.G. and Choi,S.S. (2010) Expression breadth and expression abundance behave differently in correlations with evolutionary rates. *BMC Evol. Biol.*, **10**, 241.

14. Drummond,D.A. and Wilke,C.O. (2008) Mistranslation-induced protein misfolding as a dominant constraint on coding-sequence evolution. *Cell*, **134**, 341–52.

15. Warnefors,M. and Kaessmann,H. (2013) Evolution of the Correlation Between Expression Divergence and Protein Divergence in Mammals. *Genome Biol. Evol.*, **5**, 1324–1335.

16. Zhang,J., Nielsen,R. and Yang,Z. (2005) Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Mol. Biol. Evol.*, **22**, 2472–9.

17. The ENCODE Project Consortium (2011) A user's guide to the encyclopedia of DNA elements (ENCODE). *PLoS Biol.*, **9**, e1001046.

18. Fagerberg,L., Hallstrom,B.M., Oksvold,P., Kampf,C., Djureinovic,D., Odeberg,J., Habuka,M., Tahmasebpoor,S., Danielsson,A., Edlund,K., *et al.* (2013) Analysis of the human tissue-specific

expression by genome-wide integration of transcriptomics and antibody-based proteomics. *Mol. Cell. Proteomics*, 10.1074/mcp.M113.035600.

19. Trapnell,C., Roberts,A., Goff,L., Pertea,G., Kim,D., Kelley,D.R., Pimentel,H., Salzberg,S.L., Rinn,J.L. and Pachter,L. (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat. Protoc.*, **7**, 562–78.

20. Flicek,P., Ahmed,I., Amode,M.R., Barrell,D., Beal,K., Brent,S., Carvalho-Silva,D., Clapham,P., Coates,G., Fairley,S., *et al.* (2013) Ensembl 2013. *Nucleic Acids Res.*, **41**, D48–55.

21. Rustici,G., Kolesnikov,N., Brandizi,M., Burdett,T., Dylag,M., Emam,I., Farne,A., Hastings,E., Ison,J., Keays,M., *et al.* (2013) ArrayExpress update--trends in database growth and links to data analysis tools. *Nucleic Acids Res.*, **41**, D987–90.

22. Yanai,I., Benjamin,H., Shmoish,M., Chalifa-Caspi,V., Shklar,M., Ophir,R., Bar-Even,A., Horn-Saban,S., Safran,M., Domany,E., *et al.* (2005) Genome-wide midrange transcription profiles reveal expression level relationships in human tissue specification. *Bioinformatics*, **21**, 650–9.

23. Bastian,F., Parmentier,G. and Roux,J. (2008) Bgee: integrating and comparing heterogeneous transcriptome data among species. *Data Integr. ….*

24. Brawand,D., Soumillon,M., Necsulea,A., Julien,P., Csárdi,G., Harrigan,P., Weier,M., Liechti,A., Aximu-Petri,A., Kircher,M., *et al.* (2011) The evolution of gene expression levels in mammalian organs. *Nature*, **478**, 343–8.

25. Moretti,S., Laurenczy,B., Gharib,W.H., Castella,B., Kuzniar,A., Schabauer,H., Studer,R. a, Valle,M., Salamin,N., Stockinger,H., *et al.* (2014) Selectome update: quality control and computational improvements to a database of positive selection. *Nucleic Acids Res.*, **42**, D917–21.

26. Rodriguez,J.M., Maietta,P., Ezkurdia,I., Pietrelli,A., Wesselink,J.-J., Lopez,G., Valencia,A. and Tress,M.L. (2013) APPRIS: annotation of principal and alternative splice isoforms. *Nucleic Acids Res.*, **41**, D110–7.

27. Ezkurdia,I., Juan,D., Rodriguez,J.M., Frankish,A., Diekhans,M., Harrow,J., Vazquez,J., Valencia,A. and Tress,M.L. (2014) Multiple evidence strands suggest that there may be as few as 19,000 human protein-coding genes. *Hum. Mol. ….*

28. McKusick-Nathans Institute of Genetic Medicine (2014) Online Mendelian Inheritance in Man, OMIM®. *John Hopkins Univ. (Baltimore MD)*.

29. Blake,J., Bult,C., Eppig,J., Kadin,J., Richardson,J. and Group,T.M.G.D. (2014) The Mouse Genome Database. *Nucleic Acids Res.*

30. Chen,W.-H., Minguez,P., Lercher,M.J. and Bork,P. (2012) OGEE: an online gene essentiality database. *Nucleic Acids Res.*, **40**, D901–6.

31. Canal,L. (2005) A normal approximation for the chi-square distribution. *Comput. Stat. Data Anal.*, **48**, 803–808.

32. Roux,J., Privman,E., Moretti,S., Daub,J.T., Robinson-Rechavi,M. and Keller,L. (2014) Patterns of positive selection in seven ant genomes. *Mol. Biol. Evol.*, **31**, 1661–85.

33. Khaitovich,P., Enard,W., Lachmann,M. and Pääbo,S. (2006) Evolution of primate gene expression. *Nat. Rev. Genet.*, **7**, 693–702.

34. Kuma,K., Iwabe,N. and Miyata,T. (1995) Functional Constraints against Variations on Molecules from the Tissue Level: Slowly Evolving Brain-Specific Genes Demonstrated by Protein Kinase and Immunoglobulin SupergeneFamilies. *Mol. Biol. Evol.*, **12**, 123–130.

35. Roux,J. and Robinson-Rechavi,M. (2011) Age-dependent gain of alternative splice forms and biased duplication explain the relation between splicing and duplication. *Genome Res.*, **21**, 357–63.

36. Albà,M.M. and Castresana,J. (2005) Inverse relationship between evolutionary rate and age of mammalian genes. *Mol. Biol. Evol.*, **22**, 598–606.

37. Albà,M.M. and Castresana,J. (2007) On homology searches by protein Blast and the characterization of the age of genes. *BMC Evol. Biol.*, **7**, 53.

38. Elhaik,E., Sabath,N. and Graur,D. (2006) The “inverse relationship between evolutionary rate and age of mammalian genes” is an artifact of increased genetic distance with rate of evolution and time of divergence. *Mol. Biol. Evol.*, **23**, 1–3.

39. Brunet,F.G., Roest Crollius,H., Paris,M., Aury,J.-M., Gibert,P., Jaillon,O., Laudet,V. and Robinson-Rechavi,M. (2006) Gene loss and evolutionary rates following whole-genome duplication in teleost fishes. *Mol. Biol. Evol.*, **23**, 1808–16.

40. Davis,J.C. and Petrov,D. a (2004) Preferential duplication of conserved proteins in eukaryotic genomes. *PLoS Biol.*, **2**, E55.

41. Jordan,I.K., Wolf,Y.I. and Koonin,E. V (2004) Duplicated genes evolve slower than singletons despite the initial rate increase. *BMC Evol. Biol.*, **4**, 22.

42. Satake,M., Kawata,M., McLysaght,A. and Makino,T. (2012) Evolution of Vertebrate Tissues Driven by Differential Modes of Gene Duplication. *DNA Res.*, 10.1093/dnares/dss012.

43. Urrutia,A. and Hurst,L. (2003) The signature of selection mediated by expression on human genes. *Genome Res.*, 10.1101/gr.641103.

44. Wang,C., Gong,B., Bushel,P.R., Thierry-Mieg,J., Thierry-Mieg,D., Xu,J., Fang,H., Hong,H., Shen,J., Su,Z., *et al.* (2014) The concordance between RNA-seq and microarray data depends on chemical treatment and transcript abundance. *Nat. Biotechnol.*, 10.1038/nbt.3001.

45. Rubinstein,N.D., Doron-Faigenboim,A., Mayrose,I. and Pupko,T. (2011) Evolutionary models accounting for layers of selection in protein-coding genes and their impact on the inference of positive selection. *Mol. Biol. Evol.*, **28**, 3297–308.

46. Macossay-Castillo,M., Kosol,S., Tompa,P. and Pancsa,R. (2014) Synonymous constraint elements show a tendency to encode intrinsically disordered protein segments. *PLoS Comput. Biol.*, **10**, e1003607.

47. Dimitrieva,S. and Anisimova,M. (2014) Unraveling Patterns of Site-to-Site Synonymous Rates Variation and Associated Gene Properties of Protein Domains and Families. *PLoS One*, **9**, e95034.

48. Murrell,B., Wertheim,J.O., Moola,S., Weighill,T., Scheffler,K. and Kosakovsky Pond,S.L. (2012) Detecting individual sites subject to episodic diversifying selection. *PLoS Genet.*, **8**, e1002764.

49. Zaheri,M., Dib,L. and Salamin,N. (2014) A Generalized Mechanistic Codon Model. *Mol. Biol. Evol.*, **31**, 2528–2541.

50. Moyers,B.A. and Zhang,J. (2014) Phylostratigraphic bias creates spurious patterns of genome evolution. *Mol. Biol. Evol.*

51. Montoya-Burgos,J.I., Boursot,P. and Galtier,N. (2003) Recombination explains isochores in mammalian genomes. *Trends Genet.*, **19**, 128–30.

52. Galtier,N., Duret,L., Glémin,S. and Ranwez,V. (2009) GC-biased gene conversion promotes the fixation of deleterious amino acid changes in primates. *Trends Genet.*, **25**, 1–5.

53. Bullaughey,K., Przeworski,M. and Coop,G. (2008) No effect of recombination on the efficacy of natural selection in primates. *Genome Res.*, **18**, 544–54.

54. Glémin,S., Arndt,P.F., Messer,P.W., Petrov,D. and Galtier,N. (2014) Quantification of GC-biased gene conversion in the human genome Quantification of GC-biased gene conversion in the human genome.

55. Duret,L., Mouchiroud,D. and Gautier,C. (1995) Statistical analysis of vertebrate sequences reveals that long genes are scarce in GC-rich isochores. *J. Mol. Evol.*, **40**, 308–17.

56. Subramanian,S. and Kumar,S. (2004) Gene expression intensity shapes evolutionary rates of the proteins encoded by the vertebrate genome. *Genetics*, **168**, 373–81.

57. Liao,B.-Y. and Zhang,J. (2006) Low rates of expression profile divergence in highly expressed genes and tissue-specific genes during mammalian evolution. *Mol. Biol. Evol.*, **23**, 1119–28.

58. Gu,X. and Su,Z. (2007) Tissue-driven hypothesis of genomic evolution and sequence-expression correlations. *Proc. Natl. Acad. Sci. U. S. A.*, **104**, 2779–84.

59. Haygood,R., Babbitt,C.C., Fedrigo,O. and Wray,G. a (2010) Contrasts between adaptive coding and noncoding changes during human evolution. *Proc. Natl. Acad. Sci. U. S. A.*, **107**, 7853–7.

60. Chen,J., Sun,M., Rowley,J.D. and Hurst,L.D. (2005) The small introns of antisense genes are better explained by selection for rapid transcription than by “genomic design”. *Genetics*, **171**, 2151–5.

61. Gonzalez-Porta,M., Frankish,A., Rung,J., Harrow,J. and Brazma,A. (2013) Transcriptome analysis of human tissues and cell lines reveals one dominant transcript per gene. *Genome Biol.*, **14**, R70.

62. Lahens,N.F., Kavakli,I.H., Zhang,R., Hayer,K., Black,M.B., Dueck,H., Pizarro,A., Kim,J., Irizarry,R., Thomas,R.S., *et al.* (2014) IVT-seq reveals extreme bias in RNA-sequencing. *Genome Biol.*, **15**, R86.

63. Cho,H., Davis,J., Li,X., Smith,K.S., Battle,A. and Montgomery,S.B. (2014) High-Resolution Transcriptome Analysis with Long-Read RNA Sequencing. *PLoS One*, **9**, e108095.

64. Necsulea,A. and Kaessmann,H. (2014) Evolutionary dynamics of coding and non-coding transcriptomes. *Nat. Rev. Genet.*, 10.1038/nrg3802.

65. Gilad,Y., Oshlack,A., Smyth,G.K., Speed,T.P. and White,K.P. (2006) Expression profiling in primates reveals a rapid evolution of human transcription factors. *Nature*, **440**, 242–5.

66. Blekhman,R., Marioni,J.C., Zumbo,P., Stephens,M. and Gilad,Y. (2010) Sex-specific and lineage-specific alternative splicing in primates. *Genome Res.*, **20**, 180–9.

67. Enard,W., Khaitovich,P., Klose,J. and Zöllner,S. (2002) Intra-and interspecific variation in primate gene expression patterns. *Science (80-. ).*, **296**, 340–343.

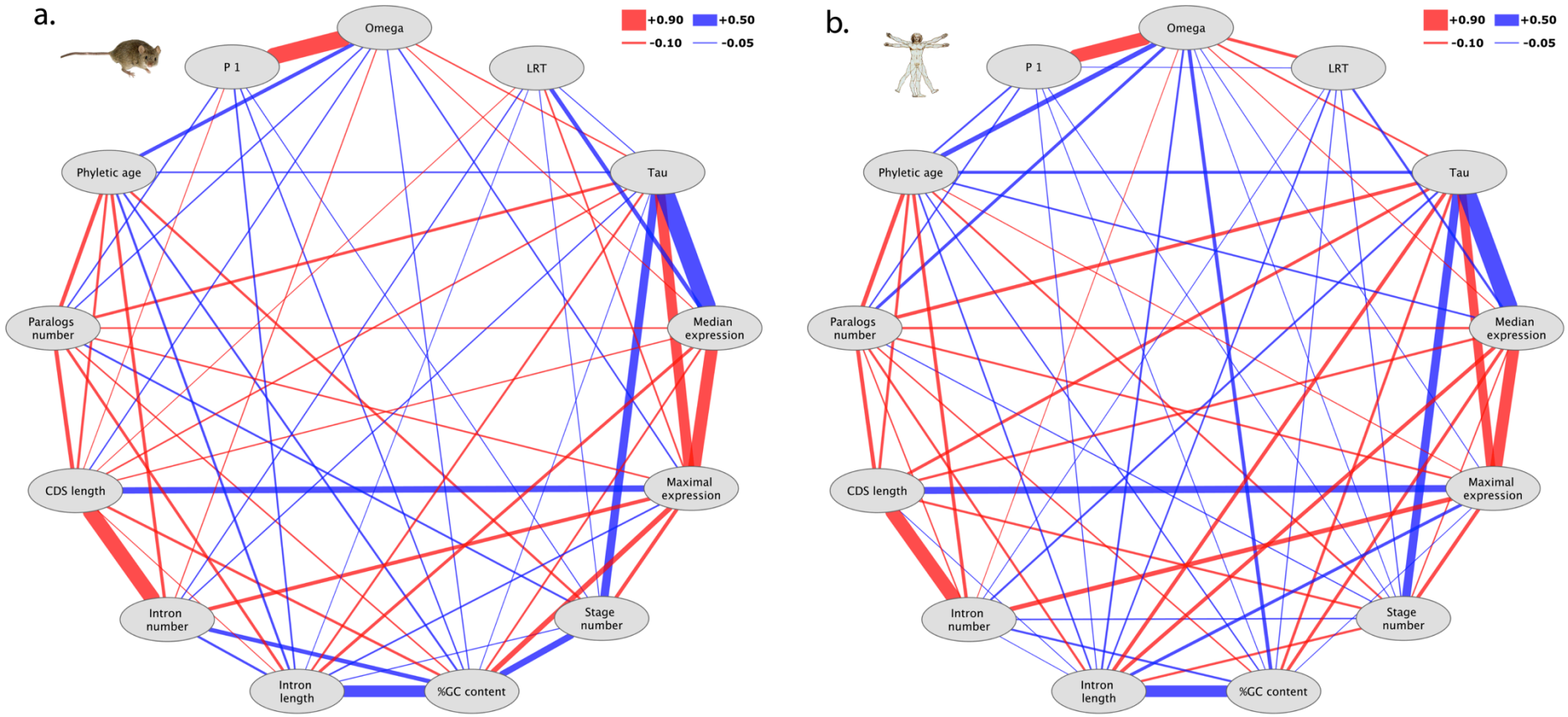
## TABLE AND FIGURES LEGENDS

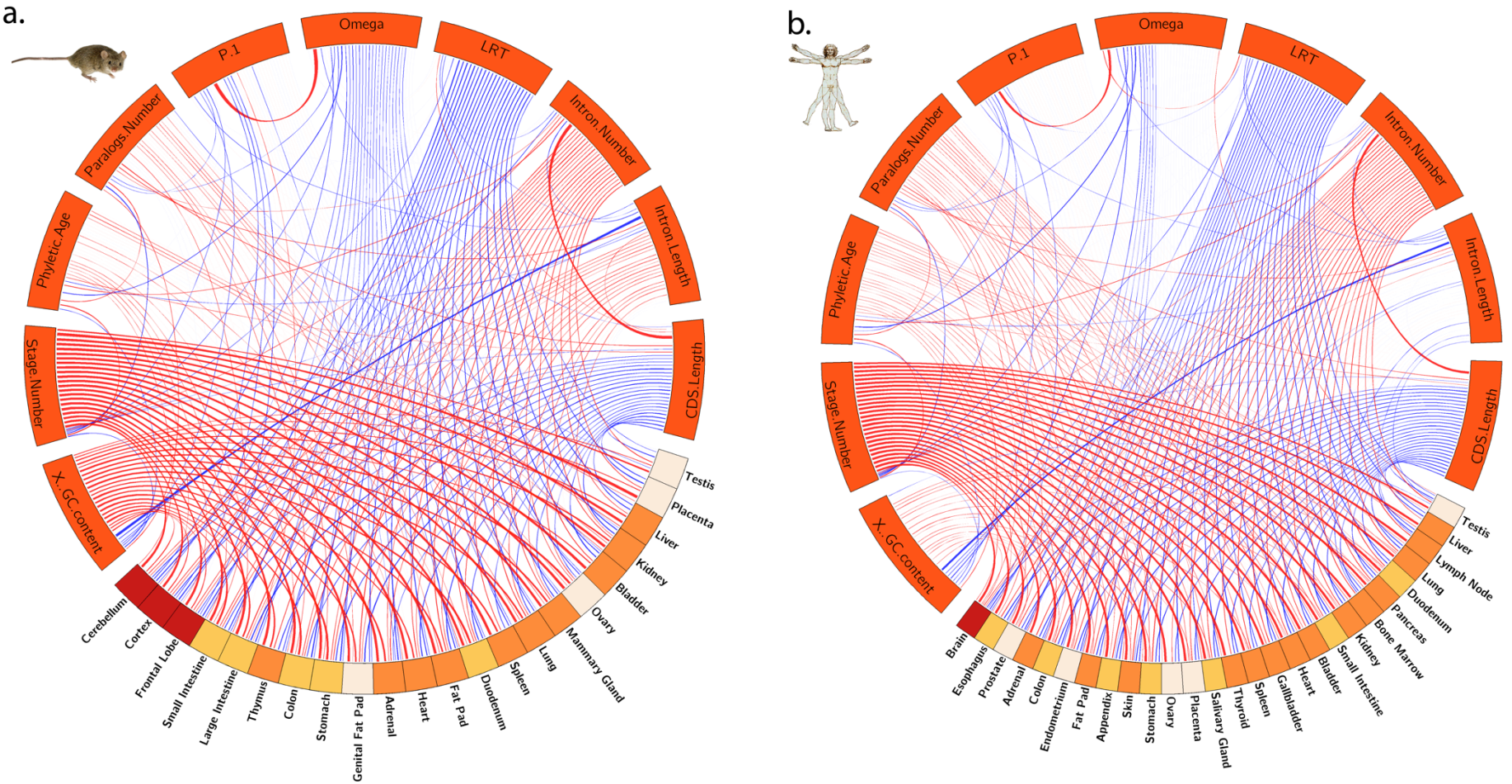
**Table 1.** Values of partial Spearman correlations between parameters, over all tissues. Top right of table: values for mouse (corresponding to Figure 1A); bottom left of table: values for human (corresponding to Figure 1B). Not significant (p-value<0.0005) are in italics.

**Figure 1.** Spearman partial correlations in a) mouse and b) human. The width of the lines shows the strength of correlations. Red lines show positive correlations, blue lines show negative correlations. Only significant correlations (p<0.0005) are shown.

**Figure 2.** Spearman partial correlation with expression values for each tissue separately for a) mouse and b) human. The width of the lines shows the strength of correlations. Red lines show positive correlations, blue shows negative correlations. Only significant correlations (p<0.0005) are shown. Color of the tissue bands represents different groups of tissues (gastrointestinal system, central nervous system, reproductive system and misc).

| mouse / human | $\omega_0$ | ΔlnL | $p_1$ | τ | Median expression | Maximal expression | Stage Number | GC content | Intron length | Intron number | CDS length | Paralogs number | Phyletic age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\omega_0$ | | *-0.031* | 0.748 | 0.067 | 0.051 | -0.074 | *-0.012* | -0.055 | *-0.030* | 0.052 | -0.062 | -0.064 | -0.163 |
| ΔlnL | 0.133 | | *0.014* | -0.048 | -0.187 | 0.079 | -0.043 | *-0.020* | -0.038 | -0.060 | 0.042 | *-0.017* | *0.000* |
| $p_1$ | 0.598 | -0.037 | | *0.024* | *-0.005* | *0.029* | -0.047 | -0.066 | -0.080 | *-0.034* | 0.042 | -0.069 | *-0.017* |
| τ | 0.085 | *-0.006* | *0.018* | | -0.803 | 0.468 | -0.374 | -0.039 | 0.094 | -0.061 | 0.074 | 0.125 | -0.069 |
| Median expression | 0.049 | 0.105 | *0.015* | -0.790 | | 0.553 | 0.012 | 0.088 | 0.135 | *0.002* | 0.061 | 0.070 | *-0.025* |
| Maximal expression | -0.041 | *0.033* | *-0.005* | 0.406 | 0.530 | | 0.164 | 0.231 | -0.076 | 0.168 | -0.283 | 0.075 | *0.010* |
| Stage Number | -0.041 | -0.055 | -0.043 | -0.381 | 0.063 | 0.163 | | -0.287 | -0.048 | *-0.006* | *0.015* | -0.087 | 0.108 |
| GC content | -0.159 | -0.049 | -0.042 | 0.121 | 0.139 | -0.039 | *-0.020* | | -0.518 | -0.190 | 0.113 | 0.071 | -0.090 |
| Intron length | -0.088 | -0.074 | -0.047 | 0.163 | 0.165 | -0.137 | 0.103 | -0.517 | | -0.103 | 0.044 | 0.123 | -0.105 |
| Intron number | 0.039 | -0.039 | *0.002* | -0.091 | *-0.029* | 0.205 | -0.065 | -0.093 | -0.037 | | 0.683 | *0.029* | 0.141 |
| CDS length | *-0.014* | *-0.011* | *0.008* | 0.135 | 0.109 | -0.312 | 0.101 | *-0.011* | -0.041 | 0.629 | | 0.181 | 0.111 |
| Paralogs number | -0.136 | -0.019 | -0.069 | 0.155 | 0.104 | 0.092 | -0.052 | 0.070 | 0.103 | 0.065 | 0.173 | | 0.175 |
| Phyletic age | -0.213 | -0.034 | -0.084 | -0.136 | -0.081 | 0.048 | 0.091 | -0.065 | -0.078 | 0.151 | 0.122 | 0.188 | |


a.
b.
+0.90
+0.50
-0.10
-0.05
Omega
P 1
LRT
Phyletic age
Tau
Paralogs number
Median expression
CDS length
Maximal expression
Intron number
Stage number
Intron length
%GC content


a.
Omega
LRT
Intron.Number
Intron.Length
CDS Length
Testis
Placenta
Liver
Kidney
Bladder
Ovary
Mammary Gland
Lung
Spleen
Duodenum
Fat Pad
Heart
Adrenal
Genital Fat Pad
Stomach
Colon
Thymus
Large Intestine
Small Intestine
Frontal Lobe
Cortex
Cerebellum
X..GC.content
Stage.Number
Phyletic.Age
Paralogs.Number
P.1
b.
Omega
LRT
Intron.Number
Intron.Length
CDS Length
Testis
Liver
Lymph Node
Lung
Duodenum
Pancreas
Bone Marrow
Kidney
Small Intestine
Bladder
Heart
Gallbladder
Spleen
Thyroid
Salivary Gland
Placenta
Ovary
Stomach
Skin
Appendix
Fat Pad
Endometrium
Colon
Adrenal
Prostate
Esophagus
Brain
X..GC.content
Stage.Number
Phyletic.Age
Paralogs.Number
P.1

# Supplementary Material 1

R code is attached at the end of this document.

**Parameters used for analysis:**

Negative selection (Omega, Omega 0, $\omega_0$), neutral evolution ($p_1$), positive selection (ΔlnL), tissue specificity (tau, τ), median expression, maximal expression, stage number, GC content, intron length, intron number, CDS length, paralogs number, phyletic age.

**Supplementary Material online:**

Can be downloaded from: http://dx.doi.org/10.6084/m9.figshare.1221771

Files 1 – 6: Data used for analysis for the 6 different data sets.

Files 7 – 30: Results of correlation analysis.

Files 31 – 36: Data used for analysis with expression for each tissue separately.

Files 37 – 60: Results of correlation analysis for each tissue separately.

Files 61 – 66: Results of correlation analysis for tissue specific genes ($\tau > 0.2$).

Files 67 – 72: Results of correlation analysis for each tissue separately for tissue specific genes ($\tau > 0.2$).

Files 73 – 78: Results of correlation analysis for genes without available evolutionary data.

File 79: Results of correlation analysis with MI score instead of omega 0.

Files 80 – 81: Results of correlation analysis with mean expression, instead of median expression.

Files 82 – 83: Data used for analysis with most expressed transcript instead of the longest transcript.

Files 84 – 85: Results of correlation analysis with most expressed transcript instead of the longest transcript.

Figure 1 – 24: Results of correlation analysis.

Figures 25 – 48: Results of correlation analysis for each tissue separately.

Figures 49 – 54: Results of correlation analysis for tissue specific genes ($\tau > 0.2$).

Figures 55 – 60: Results of correlation analysis for each tissue separately for tissue specific genes ($\tau > 0.2$).

Figures 61 – 66: Results of correlation analysis for genes without available evolutionary data.

Figures 67 – 68: Results of correlation analysis with mean expression, instead of median expression.

Figures 69 – 70: Results of correlation analysis with most expressed transcript instead of the longest transcript.

Key for file names:

Hum = Human; Mus = Mouse

Partial = Partial correlations results; Normal = Standard correlations results

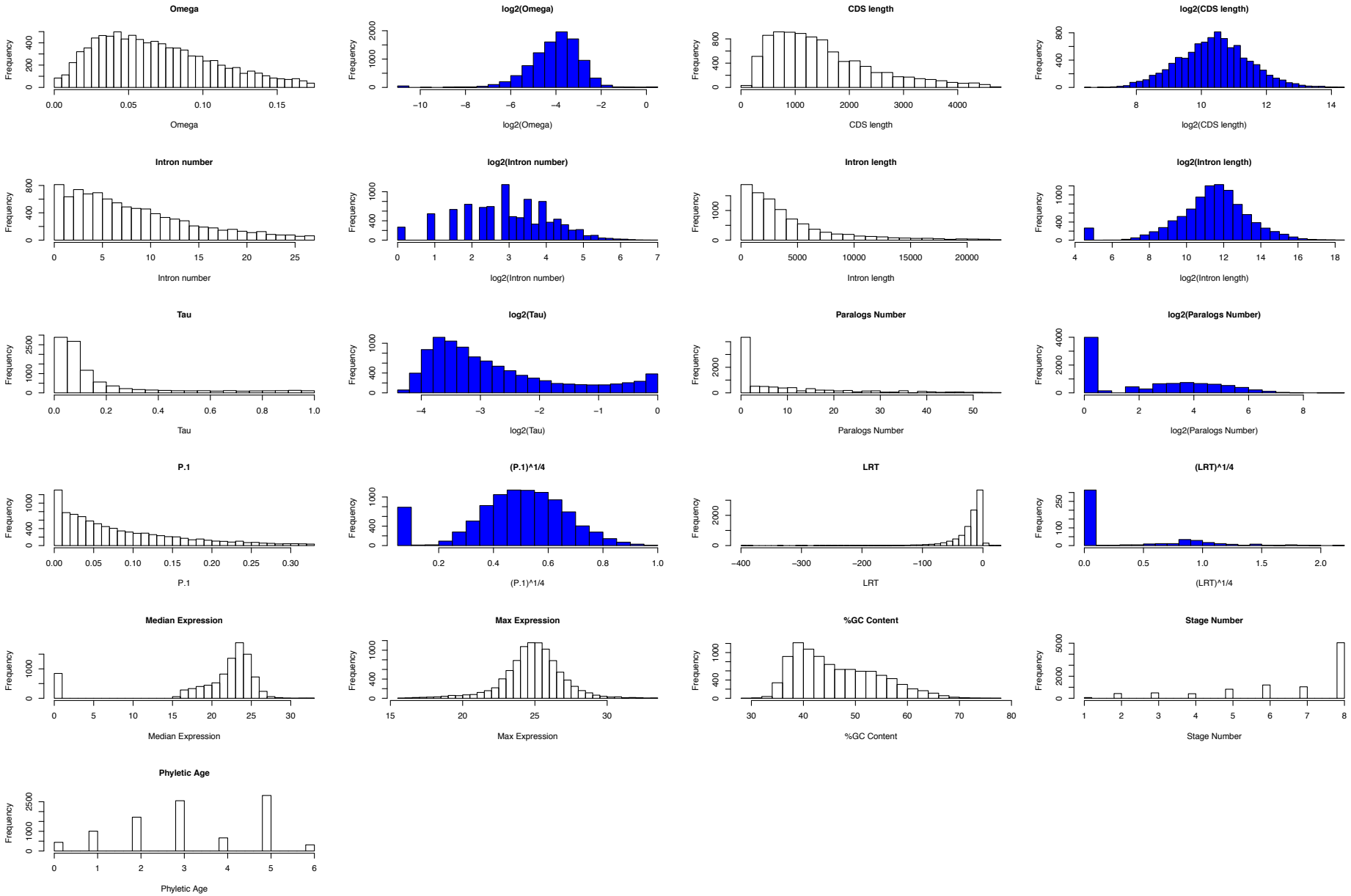
Spearman = Spearman correlations; Pearson = Pearson correlations

Tissues = data for tissues separately

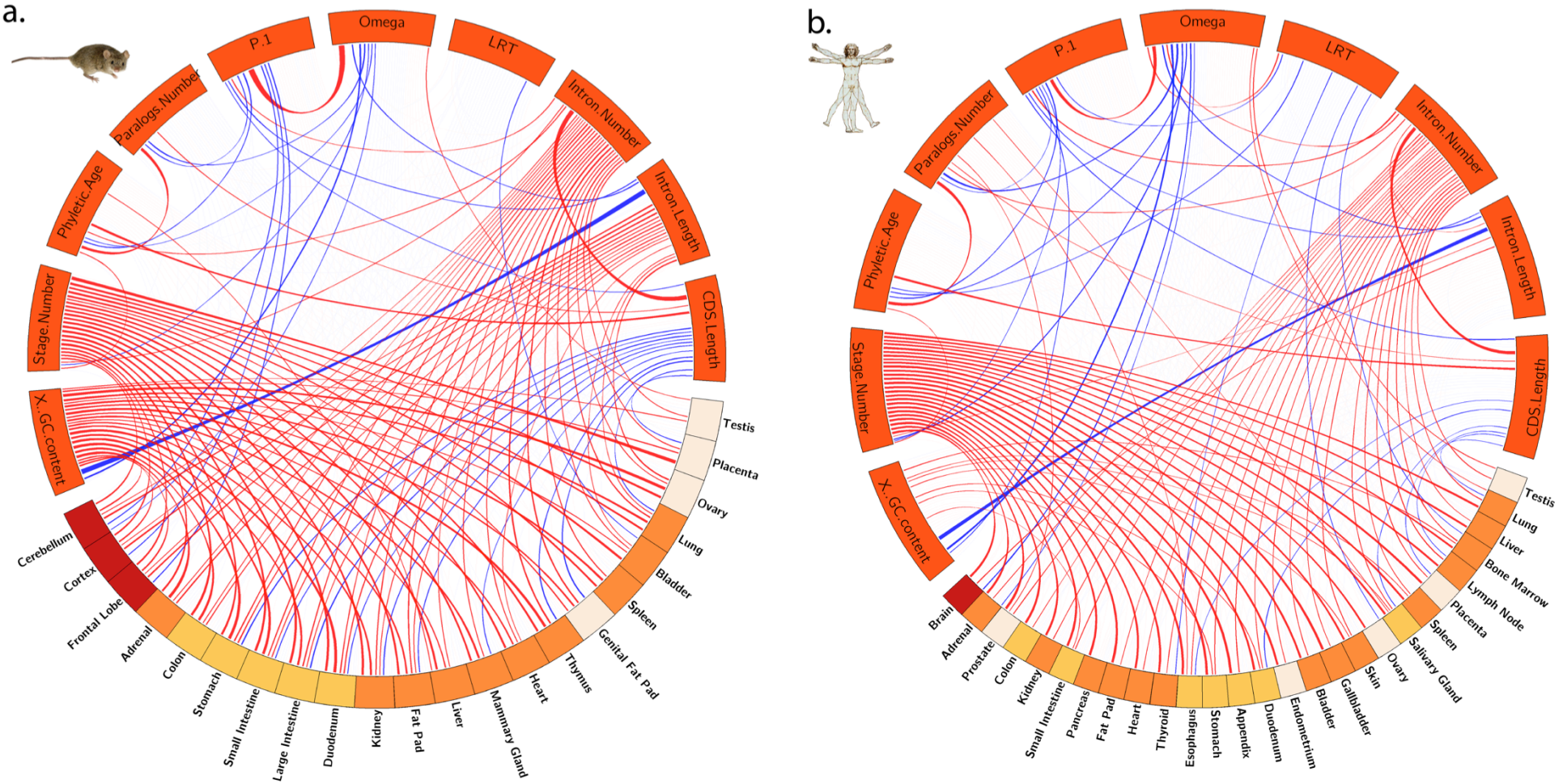
Fagerberg = (Fagerberg et al. 2013); Bgee = microarray Bgee data base (Bastian et al. 2008); Brawand = (Brawand et al. 2011)

**Supplementary table S1**: Example of calculation of partial correlation. The data used are the height of the person (random numbers between 160cm and 190cm) and length of both legs (randomly smaller 1 to 3 cm).

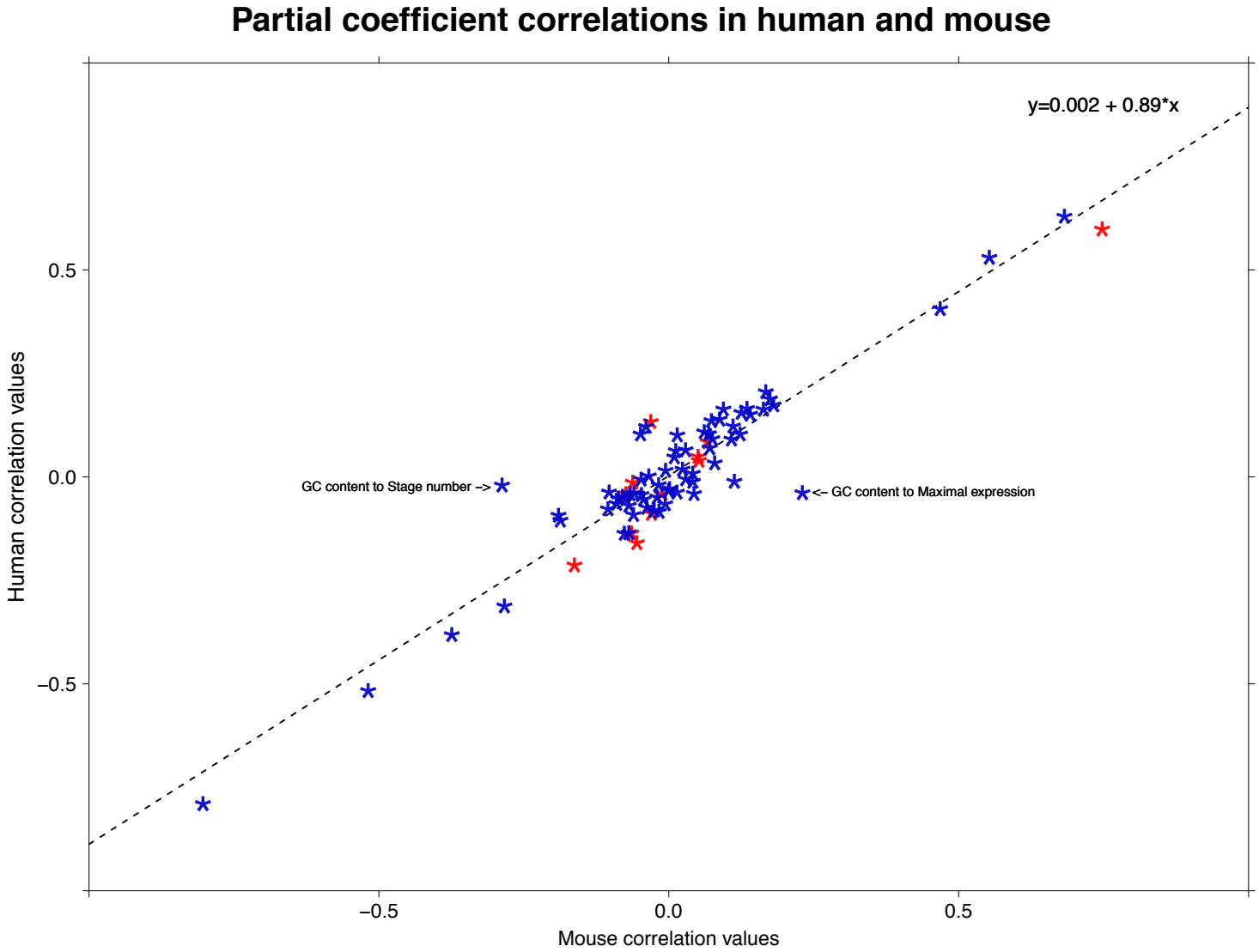
| Data$High <- runif(10 000, 160, 190)<br>Data$LeftLeg <- data$High -100 –runif(10000,1,3)<br>Data$RightLeg <- data$High – 100 – runif(10000, 1, 3) | | |
|---|---|---|
| | Simple correlation | Partial correlation |
| high vs. left leg: | R = 0.9977 | R = 0.7089 |
| high vs. right leg: | R = 0.9977 | R = 0.7031 |
| left leg vs. right leg | R = 0.9955 | R = 0.0008 |
| Data$RightLeg <- data$LeftLeg – runif(10000, 1, 3) | | |
| high vs. left leg: | R = 0.9977 | R = 0.7023 |
| high vs. right leg: | R = 0.9956 | R = 0.0042 |
| left leg vs. right leg | R = 0.9978 | R = 0.7072 |



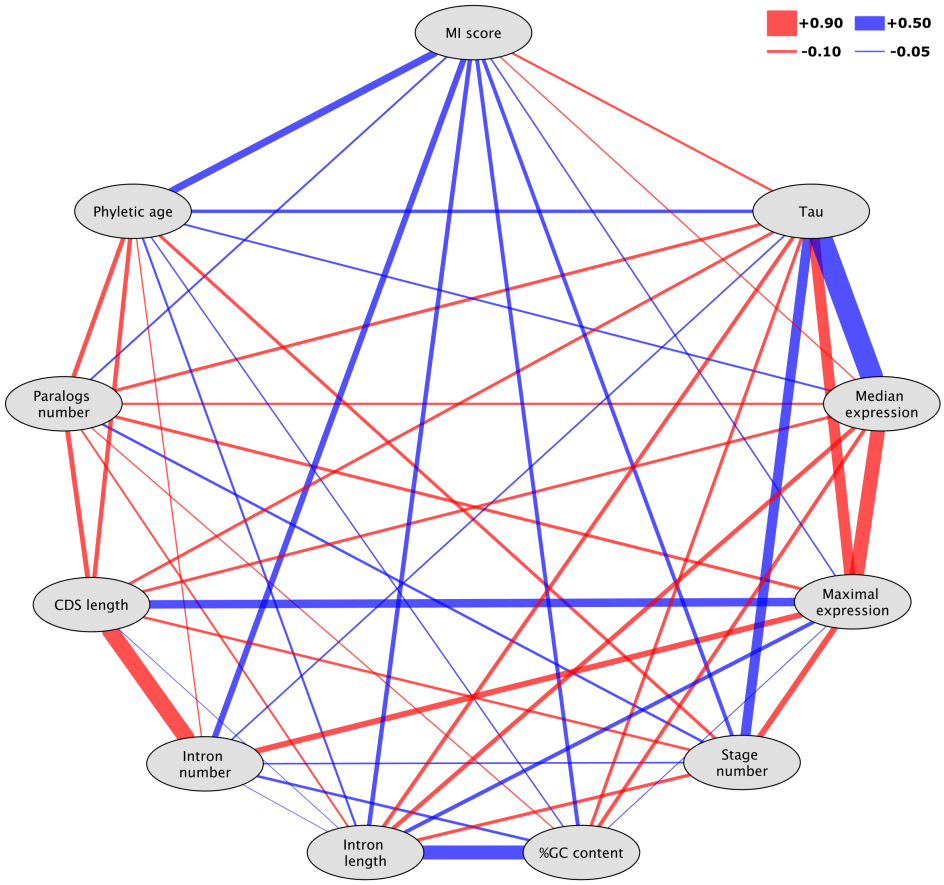

**Supplementary figure S1**: Distribution of all parameter values for human data with expression values from Fagerberg et al (Fagerberg et al. 2013).



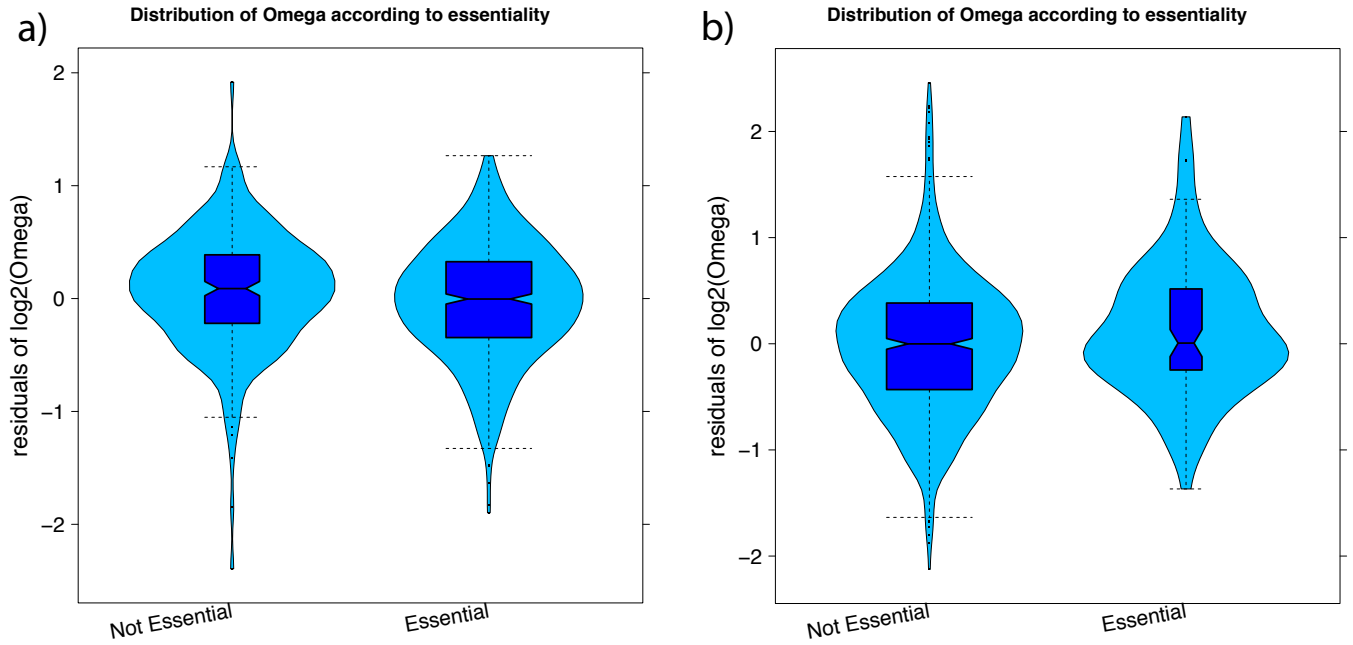

**Supplementary figure S2**: Spearman partial correlation with expression values (only tissue specific genes, $\tau > 0.2$) for each tissue separately for a) mouse and b) human. The width of the lines shows the strength of correlations. Red lines show positive correlations, blue shows negative correlations. Only significant correlations ($p<0.0005$) are shown. Color of the bands represents different groups of tissues (gastrointestinal system, central nervous system, reproductive system and divers).




**Supplementary figure S3**: Comparison of Spearman partial correlation coefficients between human and mouse. In red, correlations involving $\omega_0$.




**Supplementary figure S4**: Spearman partial correlations human using MI score to represent evolutionary rate. The width of the lines shows the strength of correlations. Red lines show positive correlations, blue shows negative correlations. Only significant correlations ($p<0.0005$) are shown.




**Supplementary figure S5**: Distribution of Omega 0 residuals according to essentiality in a) mouse (t-test p = 0.03041) and b) human (t-test p = 0.05607).

## References

Bastian F, Parmentier G, Roux J. 2008. Bgee: integrating and comparing heterogeneous transcriptome data among species. Data Integr. … [Internet]:124–131. Available from: http://link.springer.com/chapter/10.1007/978-3-540-69828-9_12

Brawand D, Soumillon M, Necsulea A, et al. 2011. The evolution of gene expression levels in mammalian organs. Nature [Internet] 478:343–348. Available from: http://www.ncbi.nlm.nih.gov/pubmed/22012392

Fagerberg L, Hallstrom BM, Oksvold P, et al. 2013. Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics. Mol. Cell. Proteomics [Internet]. Available from: http://www.ncbi.nlm.nih.gov/pubmed/24309898

The ENCODE Project Consortium. 2007. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature [Internet] 447:799–816. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2212820&tool=pmcentrez&rendertype=abstract

```
#Script to analyse mouse and human data
#RNA-seq and Microarray
#For Cytoscape: Cytoscape has to be instaled and open with CytoscapeRPC plugin installed and activated.
#For Circos: Only text files are created, circos has to be run separately.

#Used librarys
library("corpcor")
library("plyr")
library("reshape")
library("gplots")
library("lattice")
library("latticeExtra")
library("preprocessCore")
library("xtable")
library("RCytoscape")
    cy <- CytoscapeConnection()
    pluginVersion(cy)

#Color used for graphes
my.col <- colorRampPalette(c("#FFFFFF", "black", "blue", "#FA8072","#00A2FF", "#00CC00", "#E0E0E0"))(7) #1:Backgroundcolor for all graphs, 2: Foregroundcolor for all graphs (E6E6E6), 3:
 Fill for histograms, 4: Red, for boxplots, 5: Blue, for boxplots, 6: Green, for boxplots, 7: Light gray


############
#Parameters#
############
#This section has to be changed according to analysis to run
###Set folders
    setwd("~/Folder with data for analysis")

###Choose organism and data set for analysis
organism <- "Mus" #"Hum" or "Mus"
expDataSource <- "ENCODE" #Brawand, ENCODE, Bgee for Mouse; Brawand, Fagerberg, Bgee for Human;
    add <- ""

#Correlation methods used for Cytoscape and Circos
corMethod <- "spearman" #"pearson" or "spearman"
partial=TRUE #TRUE or FALSE

if (expDataSource == "ENCODE" | expDataSource == "Brawand" | expDataSource == "Fagerberg"| expDataSource == "Yu")
{
    expNorm <- 1000000 #To have all values bigger than 1
} else if (expDataSource == "Bgee" ) {
    expNorm <- 1
}

if (organism == "Mus")
{
    folder <- paste("~/Folder with data for analysis")
    folderAnalysis <- paste("~/Folder for analysis")
    organismName <- "Mouse"
    ortOrganism <- "Rat"
    dataSource <- "EnsV69"

}  else if (organism == "Hum") {
    folder <- paste("~/Folder with data for analysis")
    folderAnalysis <- paste("~/Folder for analysis")
    organismName <- "Human"
    ortOrganism <- "Chp"
    dataSource <- "EnsV69"
}

#Tissue names for different organisms and data sets
if(organism == "Mus")
{
    if (expDataSource == "ENCODE")
    {
        tissuesRPKMNames <- c("Averaged.RPKM.cerebellum","Averaged.RPKM.cortex","Averaged.RPKM.heart","Averaged.RPKM.kidney","Averaged.RPKM.liver",
"Averaged.RPKM.lung","Averaged.RPKM.placenta","Averaged.RPKM.smintestine","Averaged.RPKM.spleen","Averaged.RPKM.testis",
"Averaged.RPKM.thymus","Averaged.RPKM.adrenal","Averaged.RPKM.bladder","Averaged.RPKM.colon","Averaged.RPKM.duodenum",
"Averaged.RPKM.flobe","Averaged.RPKM.gfat","Averaged.RPKM.lgintestine","Averaged.RPKM.mamgland","Averaged.RPKM.ovary", "Averaged.RPKM.sfat","Averaged.RPKM.stomach")
        tissuesNames <- c("cerebellum", "cortex", "heart", "kidney", "liver", "lung", "placenta", "smintestine", "spleen", "testis", "thymus", "adrenal", "bladder", "colon", "duodenum",
"flobe", "gfat", "lgintestine", "mamgland", "ovary", "sfat", "stomach")
        tissuesPrintNames <- c("Cerebellum", "Cortex", "Heart", "Kidney", "Liver", "Lung","Placenta","Small Intestine","Spleen","Testis", "Thymus", "Adrenal", "Bladder", "Colon",
"Duodenum", "Frontal Lobe", "Genital Fat Pad", "Large Intestine", "Mammary Gland", "Ovary", "Subcutaneous Fat Pad", "Stomach")
    }
    else if (expDataSource == "Brawand")
    {
        tissuesRPKMNames <- c("Averaged.RPKM.brain","Averaged.RPKM.cerebellum","Averaged.RPKM.heart","Averaged.RPKM.kidney","Averaged.RPKM.liver", "Averaged.RPKM.testis")
        tissuesNames <- c("brain","cerebellum","heart","kidney","liver", "testis")
        tissuesPrintNames <- c("Brain","Cerebellum","Heart","Kidney","Liver", "Testis")
    }
    else if (expDataSource == "Bgee")
    {
        tissuesRPKMNames <- c("Averaged.RPKM.liver", "Averaged.RPKM.kidney", "Averaged.RPKM.testis", "Averaged.RPKM.blood", "Averaged.RPKM.lung", "Averaged.RPKM.colon",
"Averaged.RPKM.hcampus", "Averaged.RPKM.cortex", "Averaged.RPKM.placenta", "Averaged.RPKM.spleen", "Averaged.RPKM.ovary", "Averaged.RPKM.muscle", "Averaged.RPKM.salivary",
"Averaged.RPKM.marrow", "Averaged.RPKM.skin", "Averaged.RPKM.spinal", "Averaged.RPKM.thymus", "Averaged.RPKM.adrenal", "Averaged.RPKM.hypothalamus", "Averaged.RPKM.pituitary",
"Averaged.RPKM.duodenum", "Averaged.RPKM.cerebellum")
        tissuesNames <- c("liver", "kidney", "testis", "blood", "lung", "colon", "hcampus", "cortex", "placenta", "spleen", "ovary", "muscle", "salivary", "marrow", "skin", "spinal",
"thymus", "adrenal", "hypothalamus", "pituitary", "duodenum", "cerebellum")
        tissuesPrintNames <- c("Liver", "Kidney", "Testis", "Blood", "Lung", "Colon", "Hippocampus", "Cortex", "Placenta", "Spleen", "Ovary", "Muscle", "Salivary Gland", "Bone Marrow",
"Skin", "Spinal Cord", "Thymus", "Adrenal", "Hypothalamus", "Pituitary Gland", "Duodenum", "Cerebellum")
    }
}  else if (organism == "Hum") {
    if (expDataSource == "Fagerberg")
    {
        tissuesRPKMNames <- c("Averaged.RPKM.colon","Averaged.RPKM.kidney", "Averaged.RPKM.liver", "Averaged.RPKM.pancreas", "Averaged.RPKM.lung", "Averaged.RPKM.prostate",
"Averaged.RPKM.brain", "Averaged.RPKM.stomach", "Averaged.RPKM.spleen", "Averaged.RPKM.lymphnode", "Averaged.RPKM.appendix", "Averaged.RPKM.smint", "Averaged.RPKM.adrenal",
"Averaged.RPKM.duodenum", "Averaged.RPKM.fat", "Averaged.RPKM.endometrium", "Averaged.RPKM.placenta", "Averaged.RPKM.testis", "Averaged.RPKM.gbladder", "Averaged.RPKM.ubladder",
"Averaged.RPKM.thyroid", "Averaged.RPKM.esophagus","Averaged.RPKM.heart", "Averaged.RPKM.skin", "Averaged.RPKM.ovary", "Averaged.RPKM.bonem", "Averaged.RPKM.sgland")
        tissuesNames <- c("colon","kidney", "liver", "pancreas", "lung", "prostate", "brain", "stomach", "spleen", "lymphnode", "appendix", "smint", "adrenal", "duodenum", "fat",
"endometrium", "placenta", "testis", "gbladder", "ubladder", "thyroid", "esophagus", "heart", "skin", "ovary", "bonem", "sgland")
        tissuesPrintNames <- c("Colon","Kidney", "Liver", "Pancreas", "Lung", "Prostate", "Brain", "Stomach", "Spleen", "Lymph Node", "Appendix", "Small Intestine", "Adrenal",
"Duodenum", "Fat", "Endometrium", "Placenta", "Testis", "Gallbladder", "Urinal Bladder", "Thyroid", "Esophagus","Heart", "Skin", "Ovary", "Bone Marrow", "Salivary Gland")
    }
    else if (expDataSource == "Brawand")
    {
        tissuesRPKMNames <- c("Averaged.RPKM.fcortex","Averaged.RPKM.pcortex","Averaged.RPKM.tlobe", "Averaged.RPKM.cerebellum", "Averaged.RPKM.heart", "Averaged.RPKM.kidney",
"Averaged.RPKM.liver", "Averaged.RPKM.testis")
        tissuesNames <- c("fcortex","pcortex","tlobe","cerebellum","heart","kidney","liver", "testis")
        tissuesPrintNames <- c("Frontal Cortex","Prefrontal Cortex","Temporal Lobe","Cerebellum","Heart","Kidney","Liver", "Testis")
    }
    else if (expDataSource == "Bgee")
    {
        tissuesRPKMNames <- c("Averaged.RPKM.liver", "Averaged.RPKM.kidney", "Averaged.RPKM.testis", "Averaged.RPKM.blood", "Averaged.RPKM.lung", "Averaged.RPKM.colon",
"Averaged.RPKM.hcampus", "Averaged.RPKM.cortex", "Averaged.RPKM.placenta", "Averaged.RPKM.spleen", "Averaged.RPKM.ovary", "Averaged.RPKM.muscle", "Averaged.RPKM.salivary",
"Averaged.RPKM.marrow", "Averaged.RPKM.skin", "Averaged.RPKM.spinal", "Averaged.RPKM.thymus", "Averaged.RPKM.adrenal", "Averaged.RPKM.hypothalamus", "Averaged.RPKM.pituitary",
"Averaged.RPKM.duodenum", "Averaged.RPKM.cerebellum")
        tissuesNames <- c("liver", "kidney", "testis", "blood", "lung", "colon", "hcampus", "cortex", "placenta", "spleen", "ovary", "muscle", "salivary", "marrow", "skin", "spinal",
"thymus", "adrenal", "hypothalamus", "pituitary", "duodenum", "cerebellum")
        tissuesPrintNames <- c("Liver", "Kidney", "Testis", "Blood", "Lung", "Colon", "Hippocampus", "Cortex", "Placenta", "Spleen", "Ovary", "Muscle", "Salivary Gland", "Bone Marrow",
"Skin", "Spinal Cord", "Thymus", "Adrenal", "Hypothalamus", "Pituitary Gland", "Duodenum", "Cerebellum")
    }
}

#Number of tissues
nTissues <- length(tissuesNames)
```

```
yLim <- 1000 #for Tau comparison

#Correction term for Circos picture representation
correctionTerm <- 8*nTissues

############
#Input data#
############

    ##Filters: in Genes "protein_coding"
    ##Save: in CSV and Unique results only

        #Gene information
        #Ensembl Gene ID, Ensembl Transcript ID, Associated Gene Name, % GC content
            orgGenes = read.delim(paste("~/Gene strucuture data")

        #Ortholog information
        #Ensembl Gene ID, Ensembl Transcript ID, Ortholog Ensembl Gene ID, dN, dS, Homology Type
            orgOrthologs = read.table(paste("~/Ortholog information")
            dataOrthologs <- orgOrthologs

        #Paralog information
        #Ensembl Gene ID, Ensembl Transcript ID, Organism Paralog Ensembl Gene ID, Homology Type
            orgParalogs = read.table(paste("~/Paralog information"), sep="\t", header=TRUE)

        #Developmental information
        #Developmental data from Bgee
            orgDevelopment = read.table(paste("~/Developmental information"), sep="\t", header=TRUE)

        #PPI information
        #The number of direct neighbors of genes in protein-protein network
            orgConnectivity = read.table(paste("~/PPI information"), sep="\t", header=TRUE)

        #Protein gene connection
        #Ensembl Gene ID, Ensembl Transcript ID, Ensembl Protein ID
        orgProtein = read.table(paste("~/Protein information"), sep="\t", header=TRUE)

        #Phyletic ages of genes
        orgPhyleticage = read.table(paste("~/Phyletic age information"), sep="\t", header=TRUE)

        #Essentiality of genes
        if (organism == "Mus" | organism == "Hum") {
            orgEssentiality = read.table(paste("~/Essentiality information"), sep="\t", header=TRUE)

        #GO annotation of genes
        orgGO = read.table(paste("~/GO annotation information"), sep="\t", header=TRUE)

        #Omega from Selectome
        orgSelectome <- read.table(paste("~/Evolutionary rate information"), sep="\t", header=TRUE)

        #Ensembl Gene ID, Ensembl Transcript ID, CDS Length, Exon Rank in Transcript, Exon Chr Start (bp), Exon Chr End (bp)
        orgStructure = read.table(paste("~/Gene structure information"), sep=",", header=TRUE)

        ##Tissue expression
        if (organism == "Mus")
        {
            if (expDataSource == "ENCODE")
            {
                orgExpression <- read.table(paste("~/Expression information"), sep="\t", header=TRUE)
            } else if (expDataSource == "Brawand") {
                orgExpression <- read.table(paste("~/Expression information"), sep="\t", header=TRUE)
            } else if (expDataSource == "Bgee") {
                orgExpression <- read.table(paste("~/Expression information"), sep="\t", header=TRUE)
            }
        }  else if (organism == "Hum") {
            if (expDataSource == "Fagerberg")
            {
                orgExpression <- read.table(paste("~/Expression information"), sep="\t", header=TRUE)
                colnames(orgExpression) <-lapply(colnames(orgExpression),function(x){x <- unlist(strsplit(toString(x), split='_', fixed=TRUE))[1]})
            } else if (expDataSource == "Brawand") {
                orgExpression <- read.table(paste("~/Expression information"), sep="\t", header=TRUE)
            } else if (expDataSource == "Bgee") {
                orgExpression <- read.table(paste("~/Expression information"), sep="\t", header=TRUE)
            }
        }

#####################################
#Merge all the loaded data to one file#
#####################################

cat("\n Analysis is done for ", nTissues," tissues.",sep="")

cat("\n Overall ",nrow(orgOrthologs)," transcripts in orthologs data."," Summary:",sep="")
summary(orgOrthologs)
#Take only one to one ortologs
orgOrthologs <- orgOrthologs[regexpr("one2one",orgOrthologs$Homology.Type)>0,]
cat("\n Overall ",nrow(orgOrthologs)," transcripts with one to one orthologs."," Summary:",sep="")
summary(orgOrthologs)

cat("\n Overall ",nrow(orgSelectome)," transcripts with one to one orthologs."," Summary:",sep="")
summary(orgSelectome)

cat("\n Overall ",nrow(orgParalogs)," transcripts in paralogs data."," Summary:",sep="")
summary(orgParalogs)
#Take only within-species paralogs
orgParalogs <- orgParalogs[regexpr("within_species",orgParalogs$Homology.Type)>0,]
sumParalogs <- count(orgParalogs,"Ensembl.Gene.ID")
    names(sumParalogs) <- c("Ensembl.Gene.ID","Paralogs.Number")
    orgParalogs <- orgParalogs[,c("Ensembl.Gene.ID", "Ensembl.Transcript.ID")]
    orgParalogs <- merge(orgParalogs, sumParalogs, by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE)
    orgParalogs <- orgParalogs[,c("Ensembl.Gene.ID","Paralogs.Number")]
    orgParalogs <- unique(orgParalogs)
cat("\n Overall ",nrow(orgParalogs)," transcripts with within species paralogs."," Summary:",sep="")
summary(orgParalogs)

summary(orgDevelopment)

summary(orgConnectivity)
orgConnectivity <- orgConnectivity[,c("locus", "protein","connectivity")]
colnames(orgConnectivity) <- c("Ensembl.Gene.ID", "Ensembl.Protein.ID","Connectivity")
orgConnectivity <- merge(orgConnectivity, orgProtein, by=c("Ensembl.Gene.ID","Ensembl.Protein.ID"), all.x=TRUE, sort=FALSE)
orgConnectivity <- na.omit(orgConnectivity)
orgConnectivity <- orgConnectivity[,c("Ensembl.Gene.ID", "Ensembl.Transcript.ID","Connectivity")]
summary(orgConnectivity)

summary(orgPhyleticage)
orgPhyleticage <- orgPhyleticage[, c("locus", "phyleticage")]
colnames(orgPhyleticage) <- c("Ensembl.Gene.ID", "Phyletic.Age")
summary(orgPhyleticage)

summary(orgEssentiality)

if(organism == "Mus")
{
    orgEssentiality <- orgEssentiality[,c("ens_mouse_id", "mouse_ontology_essentiality")]
    colnames(orgEssentiality) <- c("Ensembl.Gene.ID", "Essentiality")
    orgEssentiality$Essentiality <- ifelse(orgEssentiality$Essentiality == "yes", 1, 0)
```

```
} else if (organism == "Hum") {
    orgEssentiality <- orgEssentiality[,c("ens_human_id", "human_omim_desc_essentiality")]
    colnames(orgEssentiality) <- c("Ensembl.Gene.ID", "Essentiality")
    orgEssentiality$Essentiality <- ifelse(orgEssentiality$Essentiality == "yes", 1, 0)

}
summary(orgEssentiality)

summary(orgGO)
orgGO <- orgGO[, c("locus", "GOID", "GOTerm")]
colnames(orgGO) <- c("Ensembl.Gene.ID", "GO.ID", "GO.Term")
summary(orgGO)

cat("\n Overall ",nrow(orgGenes)," transcripts", " in ",length(unique(orgGenes$Ensembl.Gene.ID))," genes for ", organismName," (",dataSource,")."," Summary:",sep="")
summary(orgGenes)

#Collect all files in one
total <- merge(orgGenes,orgOrthologs,by=c("Ensembl.Gene.ID","Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE)
total <- merge(total,orgParalogs,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE)
total$Paralogs.Number <- ifelse(is.na(total$Paralogs.Number), 0, total$Paralogs.Number)
total <- merge(total,orgConnectivity, by=c("Ensembl.Gene.ID","Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE)
total <- merge(total,orgPhyleticage,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE)
total <- merge(total,orgEssentiality,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE)
total <- merge(total,orgDevelopment,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE)
total <- merge(total,orgSelectome,by=c("Ensembl.Gene.ID","Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE)

#Add to the file with gene names dN and dS, and calculate Omega
cat("\n Omega is calculated as dN/dS if dS is > 0, otherweis Omega = 0",sep="")
total$Omega <- ifelse(total$dS>0,total$dN/total$dS,total$dS)

cat("\n Overall ",nrow(orgStructure)," exons for ", organismName," (",dataSource,")."," Summary:",sep="")
summary(orgStructure)

structure <- orgStructure
#Calculate the length of each exon
structure$Exon.Length <- ifelse(structure$Exon.Chr.End..bp.>0,structure$Exon.Chr.End..bp. - structure$Exon.Chr.Start..bp.+1,structure$Exon.Chr.End..bp.)

##Calculations for Introns

    #Calculate the summary length of all exons in transcript
    exonLength <- aggregate(Exon.Length ~ Ensembl.Transcript.ID, FUN="sum", data=structure)
    names(exonLength) <- c("Ensembl.Transcript.ID","Exon.Total.Length")

    #Calculate the number of all exons in the transcript
    exonNumber <- aggregate(Exon.Rank.in.Transcript ~ Ensembl.Transcript.ID, FUN="max", data=structure)
    names(exonNumber) <- c("Ensembl.Transcript.ID","Exon.Number")

    #Find the start of the first exon in transcript
    transcriptStart <- aggregate(Exon.Chr.Start..bp. ~ Ensembl.Transcript.ID, FUN="min", data=structure)
    names(transcriptStart) <- c("Ensembl.Transcript.ID","Transcript.Start")

    #Find the end of the last exon in transcript
    transcriptEnd <- aggregate(Exon.Chr.End..bp. ~ Ensembl.Transcript.ID, FUN="max", data=structure)
    names(transcriptEnd) <- c("Ensembl.Transcript.ID","Transcript.End")

    #Calculate the longest Transcript for each gene
    maxCDS <- aggregate(CDS.Length ~ Ensembl.Gene.ID, FUN="max", data=structure)
    names(maxCDS) <- c("Ensembl.Gene.ID","Max.CDS.Length")

    #Put all the calculated data to one table
    structure <- merge(structure,exonLength,by=c("Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE)
    structure <- merge(structure,exonNumber,by=c("Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE)
    structure <- merge(structure,transcriptStart,by=c("Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE)
    structure <- merge(structure,transcriptEnd,by=c("Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE)
    structure <- merge(structure,maxCDS,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE)

#Calculate the length of the transcript
structure$Transcript.Length <- ifelse(structure$Transcript.End>0,structure$Transcript.End - structure$Transcript.Start+1,structure$Transcript.End)

#Calculate the number of introns
structure$Intron.Number <- ifelse(structure$Exon.Number>0,structure$Exon.Number-1,structure$Exon.Number)

#Calculate the length of the Introns
structure$Intron.Length <- ifelse(structure$Transcript.Length>0,(structure$Transcript.Length - structure$Exon.Total.Length),structure$Transcript.Length)

#Intron Length is the mean length of introns in transcript
cat("\n Intron Length is the mean length of introns in the transcript.")
structure$Intron.Length <- ifelse(structure$Intron.Number>0,structure$Intron.Length/structure$Intron.Number,structure$Intron.Number)

#Label if transcript is longest for this gene
structure$Max.CDS.Length <- ifelse(structure$CDS.Length==structure$Max.CDS.Length,TRUE,FALSE)

#Choose only columns and rows that are needed
structure <- structure[,c('Ensembl.Gene.ID','Ensembl.Transcript.ID','CDS.Length','Intron.Length','Intron.Number',"Max.CDS.Length")]
structure <- unique(structure) #Many rows are the same, because before there was a row for each Exon, so just delete duplicates

#Merge the two tables (with gene properties and gene structure)
total <- merge(total,structure,by=c("Ensembl.Gene.ID","Ensembl.Transcript.ID"), all.x=TRUE)

    tempTotal <- total

##Choice of the transcript
#Transcript with avalable Omega.0 (for calculation of Omega.0 the longest transcrip was used)
#If no Omega.0 for the gene, the longest transcript
#If more transcripts have same CDS.Length, then one for which Connectivity is avalable
#If still more transcripts, the one with longest introns.

cat("\n Longest transcript will be chousen.")
#Transcript with avalable Omega.0 values
maxOmega <- aggregate(Omega.0 ~ Ensembl.Gene.ID, FUN="max", data=total)
names(maxOmega) <- c("Ensembl.Gene.ID","Max.Omega")
total <- merge(total, maxOmega, by="Ensembl.Gene.ID", all.x=TRUE)
    total1 <- total[!is.na(total$Max.Omega),] #Avalable Omega.0 for at least one transcript
        total2 <- total[is.na(total$Max.Omega),] #No Omega.0 for all transcripts
    total1 <- total1[!is.na(total1$Omega.0),]
    total1 <- total1[,-(length(colnames(total1)))] #Remove Max.Omega column
        total2 <- total2[,-(length(colnames(total2)))] #Remove Max.Omega column
        #The longest transcript
        total2 <- subset(total2,Max.CDS.Length == TRUE)
        #Connectivity data avalability, if several transcripts the same length
        maxConnectivity <- aggregate(Connectivity ~ Ensembl.Gene.ID, FUN="max", data=total2)
        names(maxConnectivity) <- c("Ensembl.Gene.ID","Max.Connectivity")
        total2 <- merge(total2, maxConnectivity, by="Ensembl.Gene.ID", all.x=TRUE)
            total21 <- total2[!is.na(total2$Max.Connectivity),] #Avalable Connectivity for at list one transcript
            total22 <- total2[is.na(total2$Max.Connectivity),] #No Connectivity for all transcripts
            total21 <- total21[!is.na(total21$Connectivity),]
        total2 <- rbind(total21, total22)
        #Maximal intron length for the rest
        maxIntron <- aggregate(Intron.Length ~ Ensembl.Gene.ID, FUN="max", data=total2)
        names(maxIntron) <- c("Ensembl.Gene.ID","Max.Intron")
        total2 <- merge(total2, maxIntron, by="Ensembl.Gene.ID", all.x=TRUE)
        total2 <- total2[total2$Intron.Length==total2$Max.Intron,]
        #Random one for the rest. 15 for Mouse
        temp <- split(1:nrow(total2),total2$Ensembl.Gene.ID)
        temp2 <- sapply(temp,function(x){x <- x[1]})
        total2 <- total2[temp2,]
        total2 <- total2[!is.na(total2$Ensembl.Gene.ID),]
        total2 <- total2[,-(length(colnames(total2)))] # Remove Max.Intron column
```

```
        total2 <- total2[,-(length(colnames(total2)))] # Remove Max.Connectivity column
        total2 <- total2[,-(length(colnames(total2)))] # Remove Max.CDS.Length column
    total1 <- total1[,-(length(colnames(total1)))] # Remove Max.CDS.Length column
total <- rbind(total1, total2)

#Bgee data were already normalized, so bring them back to FPKM
if(expDataSource == "Bgee")
{
    fmin <- function(x)
            {
                x <- subset(x,x>0)
                res <- min(x, na.rm=TRUE)
                return(res)
            }

    orgExpression[,-1] <- apply(orgExpression[,-1], c(1,2), function(x){x <- 2^x})
    minExp <- apply(orgExpression[,-1],2,fmin)
    minExp <- min(minExp)
    orgExpression[,-1] <- apply(orgExpression[,-1], c(1,2), function(x){x <- x-minExp})
}

#Choose used genes
cat("\n If many replicates for one organ, then the mean of expression is chosen.")
totalExpr <- total[,c("Ensembl.Gene.ID","Ensembl.Transcript.ID")]

if(expDataSource == "Brawand")
{
    orgExpression <- merge(totalExpr,orgExpression,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE,incomparables = NA)
    if(organism == "Hum")
    {
        orgExpression$Averaged.RPKM.fcortex <- rowMeans(orgExpression[,regexpr("_frontal_cortex",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.pcortex <- orgExpression[,regexpr("prefrontal_cortex",colnames(orgExpression))>0]
        orgExpression$Averaged.RPKM.tlobe <- orgExpression[,regexpr("temporal_lobe",colnames(orgExpression))>0]
        orgExpression$Averaged.RPKM.cerebellum <- rowMeans(orgExpression[,regexpr("Cerebellum",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.heart <- rowMeans(orgExpression[,regexpr("Heart",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.kidney <- rowMeans(orgExpression[,regexpr("Kidney",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.liver <- rowMeans(orgExpression[,regexpr("Liver",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.testis <- rowMeans(orgExpression[,regexpr("Testis",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression <- orgExpression[,c("Ensembl.Gene.ID", tissuesRPKMNames)]
    } else if(organism == "Mus"){
        orgExpression$Averaged.RPKM.brain <- rowMeans(orgExpression[,regexpr("Brain",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.cerebellum <- rowMeans(orgExpression[,regexpr("Cerebellum",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.heart <- rowMeans(orgExpression[,regexpr("Heart",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.kidney <- rowMeans(orgExpression[,regexpr("Kidney",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.liver <- rowMeans(orgExpression[,regexpr("Liver",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression$Averaged.RPKM.testis <- rowMeans(orgExpression[,regexpr("Testis",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
        orgExpression <- orgExpression[,c("Ensembl.Gene.ID", tissuesRPKMNames)]
    }
} else if(expDataSource == "ENCODE"){
    #orgExpression <- merge(totalExpr,orgExpression,by=c("Ensembl.Gene.ID","Ensembl.Transcript.ID"), all.x=TRUE, sort=FALSE,incomparables = NA)
    orgExpression <- merge(totalExpr,orgExpression,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE,incomparables = NA)
    orgExpression$Averaged.RPKM.cerebellum <- rowMeans(orgExpression[,regexpr("Cbellum",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.cortex <- rowMeans(orgExpression[,regexpr("Cortex",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.heart <- rowMeans(orgExpression[,regexpr("Heart",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.kidney <- rowMeans(orgExpression[,regexpr("Kidney",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.liver <- rowMeans(orgExpression[,regexpr("Liver",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.lung <- rowMeans(orgExpression[,regexpr("Lung",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.placenta <- rowMeans(orgExpression[,regexpr("Plac",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.smintestine <- rowMeans(orgExpression[,regexpr("Smint",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.spleen <- rowMeans(orgExpression[,regexpr("Spleen",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.testis <- rowMeans(orgExpression[,regexpr("Testis",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.thymus <- rowMeans(orgExpression[,regexpr("Thymus",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.adrenal <- rowMeans(orgExpression[,regexpr("Adrenal",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.bladder <- rowMeans(orgExpression[,regexpr("Bladder",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.colon <- rowMeans(orgExpression[,regexpr("Colon",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.duodenum <- rowMeans(orgExpression[,regexpr("Duod",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.flobe <- rowMeans(orgExpression[,regexpr("Flobe",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.gfat <- rowMeans(orgExpression[,regexpr("Gfat",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.lgintestine <- rowMeans(orgExpression[,regexpr("Lgint",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.mamgland <- rowMeans(orgExpression[,regexpr("Mamg",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.ovary <- rowMeans(orgExpression[,regexpr("Ovary",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.sfat <- rowMeans(orgExpression[,regexpr("Sfat",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.stomach <- rowMeans(orgExpression[,regexpr("Stom",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    # orgExpression <- orgExpression[,c("Ensembl.Gene.ID","Ensembl.Transcript.ID", tissuesRPKMNames)]
    orgExpression <- orgExpression[,c("Ensembl.Gene.ID", tissuesRPKMNames)]
} else if(expDataSource == "Bgee"){
    orgExpression <- merge(totalExpr,orgExpression,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE,incomparables = NA)

    orgExpression$Averaged.RPKM.liver <- rowMeans(orgExpression[,regexpr("liver",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.kidney <- rowMeans(orgExpression[,regexpr("kidney",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.testis <- rowMeans(orgExpression[,regexpr("testis",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.blood <- rowMeans(orgExpression[,regexpr("blood",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.lung <- rowMeans(orgExpression[,regexpr("lung",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.colon <- rowMeans(orgExpression[,regexpr("colon",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.hcampus <- rowMeans(orgExpression[,regexpr("hippocampus",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.cortex <- rowMeans(orgExpression[,regexpr("cortex",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.placenta <- rowMeans(orgExpression[,regexpr("placenta",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.spleen <- rowMeans(orgExpression[,regexpr("spleen",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.ovary <- rowMeans(orgExpression[,regexpr("ovary",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.muscle <- rowMeans(orgExpression[,regexpr("muscle",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.salivary <- rowMeans(orgExpression[,regexpr("salivary",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.marrow <- rowMeans(orgExpression[,regexpr("marrow",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.skin <- rowMeans(orgExpression[,regexpr("skin",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.spinal <- rowMeans(orgExpression[,regexpr("spinal",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.thymus <- rowMeans(orgExpression[,regexpr("thymus",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.adrenal <- rowMeans(orgExpression[,regexpr("adrenal",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.hypothalamus <- rowMeans(orgExpression[,regexpr("hypothalamus",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.pituitary <- rowMeans(orgExpression[,regexpr("pituitary",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.duodenum <- rowMeans(orgExpression[,regexpr("duodenum",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.cerebellum <- rowMeans(orgExpression[,regexpr("cerebellum",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression <- orgExpression[,c("Ensembl.Gene.ID", tissuesRPKMNames)]

} else if(expDataSource == "Fagerberg"){
    orgExpression <- merge(totalExpr,orgExpression,by=c("Ensembl.Gene.ID"), all.x=TRUE, sort=FALSE,incomparables = NA)

    orgExpression$Averaged.RPKM.colon <- rowMeans(orgExpression[,regexpr("colon",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.kidney <- rowMeans(orgExpression[,regexpr("kidney",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.liver <- rowMeans(orgExpression[,regexpr("liver",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.pancreas <- rowMeans(orgExpression[,regexpr("pancreas",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.lung <- rowMeans(orgExpression[,regexpr("lung",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.prostate <- rowMeans(orgExpression[,regexpr("prostate",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.brain <- rowMeans(orgExpression[,regexpr("brain",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.stomach <- rowMeans(orgExpression[,regexpr("stomach",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.spleen <- rowMeans(orgExpression[,regexpr("spleen",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.lymphnode <- rowMeans(orgExpression[,regexpr("lymphnode",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.appendix <- rowMeans(orgExpression[,regexpr("appendix",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.smint <- rowMeans(orgExpression[,regexpr("smallintestine",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.adrenal <- rowMeans(orgExpression[,regexpr("adrenal",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.duodenum <- rowMeans(orgExpression[,regexpr("duodenum",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.fat <- rowMeans(orgExpression[,regexpr("fat",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.endometrium <- rowMeans(orgExpression[,regexpr("endometrium",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.placenta <- rowMeans(orgExpression[,regexpr("placenta",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.testis <- rowMeans(orgExpression[,regexpr("testis",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.gbladder <- rowMeans(orgExpression[,regexpr("gallbladder",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.ubladder <- rowMeans(orgExpression[,regexpr("urinarybladde",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.thyroid <- rowMeans(orgExpression[,regexpr("thyroid",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.esophagus <- rowMeans(orgExpression[,regexpr("esophagus",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.heart <- rowMeans(orgExpression[,regexpr("heart",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
```

```
    orgExpression$Averaged.RPKM.skin <- rowMeans(orgExpression[,regexpr("skin",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.ovary <- rowMeans(orgExpression[,regexpr("ovary",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.bonem <- rowMeans(orgExpression[,regexpr("bonem",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression$Averaged.RPKM.sgland <- rowMeans(orgExpression[,regexpr("salivarygland",colnames(orgExpression))>0], na.rm=TRUE, dim=1)
    orgExpression <- orgExpression[,c("Ensembl.Gene.ID", tissuesRPKMNames)]

}

    #Draw expression distribution
        orgRPKM <- na.omit(orgExpression)

        orgRPKM[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))] <- apply(orgRPKM[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))], c(1,2), function(x){x <-
log2(x)})

        dev.new(height=9, width=12)
        par(cex.main=0.95, bg=my.col[1], fg=my.col[2], col.axis=my.col[2], col.lab=my.col[2], col.main=my.col[2])
        palette(rev(rich.colors(length(tissuesNames)+2)))

        plot(density(orgRPKM[,tissuesRPKMNames[1]],n=1000), main = "Expression values among different tissues",xlab="RPKM",col=(1), lwd=3)
        for(i in c(2:length(tissuesRPKMNames)))
        {
            lines(density(orgRPKM[,tissuesRPKMNames[i]],n = 1000), col=(i), lwd=3)
        }
        legend("topright",tissuesPrintNames,col=(1:length(tissuesRPKMNames)),lty="solid", lwd=3)
        dev.copy2pdf(device=quartz, file=paste(folderAnalysis, organism,"TissuesOriginalExpressionRPKM", expDataSource, add,".pdf", sep=""),onefile=TRUE)
        #dev.off()

        orgRPKM <- na.omit(orgExpression)

        x2 <- orgRPKM[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))]
        x2 <- x2 * expNorm + 1.0001
        x2 <- log2(x2)

        orgRPKM[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))] <- data.frame(x2)

        dev.new(height=9, width=12)
        par(cex.main=0.95, bg=my.col[1], fg=my.col[2], col.axis=my.col[2], col.lab=my.col[2], col.main=my.col[2])
        palette(rev(rich.colors(length(tissuesNames)+2)))

        plot(density(orgRPKM[,tissuesRPKMNames[1]],n=1000), main = "Expression values among different tissues",xlab="Normalized expression",col=(1), lwd=3)
        for(i in c(2:length(tissuesRPKMNames)))
        {
            lines(density(orgRPKM[,tissuesRPKMNames[i]],n = 1000), col=(i), lwd=3)
        }
        legend("topright",tissuesPrintNames,col=(1:length(tissuesRPKMNames)),lty="solid", lwd=3)
        dev.copy2pdf(device=quartz, file=paste(folderAnalysis, organism,"TissuesOriginalExpressionRPKM", expDataSource, "Extended" , add, ".pdf", sep=""),onefile=TRUE)
        #dev.off()

#Quintile normalization
orgExpression <- na.omit(orgExpression)
x <- orgExpression[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))]
x <- x * expNorm
x <- log2(x)
x[x<0] <- -Inf
x_m <- as.matrix(x)
x <- normalize.quantiles(x_m)
x[x == -Inf] <- log2(1.0001)
orgExpression[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))] <- data.frame(x)

    dev.new(height=9, width=12)
        par(cex.main=0.95, bg=my.col[1], fg=my.col[2], col.axis=my.col[2], col.lab=my.col[2], col.main=my.col[2])
        palette(rev(rich.colors(length(tissuesNames)+2)))

        plot(density(x[,1],n=1000), main = "Expression values among different tissues",xlab="Quantile normalized expression",col=(1), lwd=3)
            for(i in c(2:length(tissuesRPKMNames)))
            {
                lines(density(x[,i],n = 1000), col=(i), lwd=3)
            }
            legend("topright",tissuesPrintNames,col=(1:length(tissuesRPKMNames)),lty="solid", lwd=3)
                dev.copy2pdf(device=quartz, file=paste(folderAnalysis, organism,"OriginalExpressionRPKM", expDataSource,"ExtendedQN", add, ".pdf", sep=""),onefile=TRUE)
    #dev.off()

cat("\n Mean is calculated taking in account tissues with 0 expression. 2+0+4=2",sep="")
fmean <- function(x)
{
    #x <- subset(x,x>0)
    if(!all(is.na(x)))
     {
        res <- mean(x, na.rm=TRUE)
     } else {
        res <- NA
     }
     return(res)
}
orgExpression$Mean.Expression <- apply(orgExpression[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))],1,fmean)

cat("\n Median is calculated taking in account tissues with 0 expression. 2+0+4=2",sep="")
fmedian <- function(x)
{
    #x <- subset(x,x>0)
    if(!all(is.na(x)))
     {
        res <- median(x, na.rm=TRUE)
     }else {
        res <- NA
     }
     return(res)
}
orgExpression$Median.Expression <- apply(orgExpression[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))],1,fmedian)

#Maximal expression value over all tissues
fmax <- function(x)
{
    if(!all(is.na(x)))
    {
        res <- max(x, na.rm=TRUE)
     } else {
        res <- NA
     }
    return(res)
}
orgExpression$Max.Expression <- apply(orgExpression[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""))],1,fmax)

#Function to calculate Tau, xi=xi/max(xi), tau=sum(1-xi)/(n-1)
ftau <- function(x)
{
    if(!all(is.na(x)))
    {
        x <- (1-(x/x[length(x)]))
        res <- sum(x, na.rm=TRUE)
        res <- res/(length(x)-1)
     } else {
        res <- NA
     }
     return(res)
}
orgExpression$Tau <- apply(orgExpression[,c(paste("Averaged.RPKM.", tissuesNames[1:nTissues], sep=""), paste("Max.Expression", sep=""))],1,ftau)
```

```
cat("\n Expression data are avalable for: ", nrow(orgExpression)," genes.",sep="")
cat("\n Summary of expression data after normalisation and calculating tau: ")

###Output
write.table(orgExpression, file=paste(folderAnalysis, organism, "Expression", expDataSource, add,".txt",sep=""),row.names = FALSE, col.names=TRUE, quote = FALSE)
summary(orgExpression)
    totalTemp1 <- total
write.table(total, file=paste(folderAnalysis, organism, "TotalSum", expDataSource,"Parameters", add, ".txt",sep=""), row.names = FALSE, col.names=TRUE, quote = FALSE)

# if(expDataSource != "ENCODE")
# {
    #Add calculated expression values to the main table
    total <- merge(total,orgExpression,by=c("Ensembl.Gene.ID"), all.x=TRUE, incomparables = NA, sort=FALSE)
# } else if (expDataSource == "ENCODE") {
    # total <- merge(total,orgExpression,by=c("Ensembl.Gene.ID","Ensembl.Transcript.ID"), all.x=TRUE, incomparables = NA, sort=FALSE)
    # totalTemp2 <- total

    # #The most expressed transcript will be chosen
    # total <- total[!is.na(total$CDS.Length),]

    # #Omega 0 is assumed for gene not for individual transcript
    # maxOmega <- aggregate(Omega.0 ~ Ensembl.Gene.ID, FUN="max", data=total)
    # names(maxOmega) <- c("Ensembl.Gene.ID","Omega.0")
    # total <- total[,-which(names(total) %in% c("Omega.0"))]
    # total <- merge(total, maxOmega, by="Ensembl.Gene.ID", all.x=TRUE)

    # maxLRT <- aggregate(LRT ~ Ensembl.Gene.ID, FUN="max", data=total)
    # names(maxLRT) <- c("Ensembl.Gene.ID","LRT")
    # total <- total[,-which(names(total) %in% c("LRT"))]
    # total <- merge(total, maxLRT, by="Ensembl.Gene.ID", all.x=TRUE)

    # maxP1 <- aggregate(P.1 ~ Ensembl.Gene.ID, FUN="max", data=total)
    # names(maxP1) <- c("Ensembl.Gene.ID","P.1")
    # total <- total[,-which(names(total) %in% c("P.1"))]
    # total <- merge(total, maxP1, by="Ensembl.Gene.ID", all.x=TRUE)

    # maxPS <- aggregate(Positive.Selection ~ Ensembl.Gene.ID, FUN="max", data=total)
    # names(maxPS) <- c("Ensembl.Gene.ID","Positive.Selection")
    # total <- total[,-which(names(total) %in% c("Positive.Selection"))]
    # total <- merge(total, maxPS, by="Ensembl.Gene.ID", all.x=TRUE)

    # #Choice of the transcript with maximal median expression
    # maxMedianE <- aggregate(Median.Expression ~ Ensembl.Gene.ID, FUN="max", data=total)
    # names(maxMedianE) <- c("Ensembl.Gene.ID","Max.Median.Expression")
    # total <- merge(total, maxMedianE, by="Ensembl.Gene.ID", all.x=TRUE)
    # total$Max.Median.Expression <- ifelse(total$Median.Expression == total$Max.Median.Expression,TRUE,FALSE)
    # total <- subset(total,Max.Median.Expression == TRUE)

    # #Choise of transcript with maximal maximal expression
    # maxMaxE <- aggregate(Max.Expression ~ Ensembl.Gene.ID, FUN="max", data=total)
    # names(maxMaxE) <- c("Ensembl.Gene.ID","Max.Max.Expression")
    # total <- merge(total, maxMaxE, by="Ensembl.Gene.ID", all.x=TRUE)
    # total$Max.Max.Expression <- ifelse(total$Max.Expression == total$Max.Max.Expression,TRUE,FALSE)
    # total <- subset(total,Max.Max.Expression == TRUE)

    # #For the rest, the longest transcript is chousen
    # minIntron <- aggregate(Intron.Length ~ Ensembl.Gene.ID, FUN="min", data=total)
    # names(minIntron) <- c("Ensembl.Gene.ID","Min.Intron")
    # total <- merge(total, minIntron, by="Ensembl.Gene.ID", all.x=TRUE)
    # total$Min.Intron <- ifelse(total$Intron.Length == total$Min.Intron,TRUE,FALSE)
    # total <- subset(total,Min.Intron == TRUE)

    # temp <- split(1:nrow(total),total$Ensembl.Gene.ID)
    # temp2 <- sapply(temp,function(x){x <- x[1]})
    # total <- total[temp2,]
    # total <- total[!is.na(total$Ensembl.Gene.ID),]
# }

#File to save the data to all the tissues
totalTissues <- total[,c("Ensembl.Gene.ID", "Ensembl.Transcript.ID","CDS.Length", "Intron.Length", "Intron.Number", "Omega", "Omega.0", "LRT", "P.1", "Positive.Selection",
 "Max.Expression", "Mean.Expression", "Median.Expression", "Tau", "X..GC.content", "Paralogs.Number", "Stage.Number", "Stage.First", "Connectivity", "Phyletic.Age", "Essentiality",
 tissuesRPKMNames)]

#Choose only usefull colums
total <- total[,c("Ensembl.Gene.ID", "Ensembl.Transcript.ID",  "X..GC.content", "Omega.0", "LRT", "P.1", "Positive.Selection", "CDS.Length", "Intron.Length", "Intron.Number", "Omega",
 "Max.Expression", "Mean.Expression", "Median.Expression", "Tau", "Paralogs.Number", "Stage.Number", "Stage.First", "Connectivity", "Phyletic.Age", "Essentiality")]
cat("\n Overall ",nrow(total)," genes."," Summary:",sep="")
summary(total)

cat("\n Overall ",nrow(totalTissues)," genes for each tissue."," Summary:",sep="")
summary(totalTissues)

###Save the results
write.table(total,file=paste(folderAnalysis, organism, "Table", expDataSource, add, ".txt",sep=""),row.names = FALSE, col.names=TRUE, quote = FALSE)
write.table(totalTissues,file=paste(folderAnalysis, organism, "TableTissues", expDataSource, add,".txt",sep=""), row.names = FALSE, col.names=TRUE, quote = FALSE)

######################
######################
######################


#########################################################
#Make partial correlation with glm model for Cytoscape#
#########################################################

#Load the data
data <- read.table(paste(folderAnalysis,organism, "Table", expDataSource, ".txt",sep=""), header=TRUE)

add <- ""
#partial <- FALSE
#corMethod <- "pearson" # #"spearman"

cat("\n Summary of the data in the first step: ", sep="")
summary(data)

        # ##Only essential human-mouse orthologs
        # data <- data[data$Essentiality==0,]
        # data <- data[!is.na(data$Essentiality),]

        # ##Only specific genes
        # data <- data[data$Tau>0.2,] #0.2 used to define tissue specific genes
        # data <- data[!is.na(data$Tau),]

            # geneData <- data.frame(H=runif(10000, 160, 190))
            # #geneData$High <- runif(10000, 160, 190)
            # geneData$LL <- geneData$H - 100 - runif(10000, 1, 3)
            # geneData$RL <- geneData$H - 100 - runif(10000, 1, 3)
            # geneData$RL <- geneData$LL - runif(10000, 1, 3)
            # corMethod="spearman"
            # partial <- TRUE

        # ###Change Omaga 0 to MI score from appris data base, only for human
        # data2 <-  read.table(paste(folder, "APPRIS_DB.csv",sep=""), header=TRUE, sep=";")
        # data2 <- data2[,c("ENSEMBL", "MI.score")]
        # colnames(data2) <- c("Ensembl.Gene.ID", "MI.Score")
        # data2$MI.Score <- ifelse(as.numeric(as.character(data2$MI.Score))<0, NA, as.numeric(as.character(data2$MI.Score)))
        # data <- merge(data, data2, by="Ensembl.Gene.ID", all.x=TRUE,sort=FALSE)
        # ##Leave only selected columns
```

```
        # data <- data[,c("MI.Score", "CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number", "Stage.Number",
"Phyletic.Age")]
        # colnames(data) <- c("Omega", "CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number", "Stage.Number",
"Phyletic.Age")
        # parametersNames <- c("MI score", "CDS length", "Intron \n length", "Intron \n number", "Median \n expression", "Maximal \n expression", "Tau", "%GC content", "Paralogs \n
number", "Stage \n number", "Phyletic age")

        # ###Only genes without Omega 0
        # data <- data[is.na(data$Omega.0),]
        # data <- data[,c("CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number", "Stage.Number",
"Phyletic.Age")] #"Connectivity"
        # colnames(data) <- c("CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number", "Stage.Number",
"Phyletic.Age")#
        # parametersNames <- c("CDS length", "Intron \n length", "Intron \n number", "Median \n expression", "Maximal \n expression", "Tau", "%GC content", "Paralogs \n number", "Stage
\n number", "Phyletic age")


    ###All parameters
    ##Leave only needed columns
    data <- data[,c("Omega.0", "LRT", "P.1","CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number",
"Stage.Number", "Phyletic.Age")]
    colnames(data) <- c("Omega",  "LRT", "P.1", "CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number",
"Stage.Number", "Phyletic.Age")
    parametersNames <- c("Omega",  "LRT", "P 1", "CDS length", "Intron \n length", "Intron \n number", "Median \n expression", "Maximal \n expression", "Tau", "%GC content", "Paralogs \n
number", "Stage \n number", "Phyletic age")

    cat("\n To calculate correlation ", corMethod, " correlation was used.",sep="")
    if (partial)
    {
        cat("\n Partial correlation was performed.",sep="")
        part <- "Partial"
    } else {
        cat("\n Normal correlation was performed.",sep="")
        part <- "Normal"
    }

#Delete all genes with NA and not known parameters
data <- na.omit(data)
data <- data[data$Max.Expression>0.00015,] #Genes that are not expressed in any tissue

cat("\n All the genes with unknown parameters are removed from the analysis. ", nrow(data), " are left for the analysis.", " Summary: ", sep="")
summary(data)

geneData<-data

#Normalization of the data
minOmega <- min(geneData$Omega[geneData$Omega>0])
geneData$Omega <- geneData$Omega + minOmega
geneData$Omega <- log2(geneData$Omega)

minLength <- min(geneData$Intron.Length[geneData$Intron.Length>0])
geneData$Intron.Length <- geneData$Intron.Length + minLength
geneData$Intron.Length <- log2(geneData$Intron.Length)

minP1 <- min(geneData$P.1[geneData$P.1>0])
geneData$P.1 <- geneData$P.1 + minP1
geneData$P.1 <- sqrt(sqrt(geneData$P.1))

geneData$Intron.Number <- geneData$Intron.Number + 1
geneData$Intron.Number <- log2(geneData$Intron.Number)

geneData$Paralogs.Number <- geneData$Paralogs.Number + 1
geneData$Paralogs.Number <- log2(geneData$Paralogs.Number)

geneData$Phyletic.Age <- geneData$Phyletic.Age + 1

minTau <- 1 - max(geneData$Tau)
geneData$Tau <- geneData$Tau + minTau
geneData$Tau <- log2(geneData$Tau)

geneData$LRT <- ifelse(geneData$LRT<0, 0, geneData$LRT)
geneData$LRT <- sqrt(sqrt(geneData$LRT))

geneData$CDS.Length <- log2(geneData$CDS.Length)

cat("\n The data are normalised. Log2 of all paremeters is taken, exept GC content.", sep="")
cat("Summary of the data after normalization:",sep="")
summary(geneData)

cat("\n Graphical representation of the data is saved in the file \"Parameters\".", sep="")

    dev.new(height=12, width=18)
    par(mfrow=c(6,4),cex.main=0.95, bg=my.col[1], fg=my.col[2], col.axis=my.col[2], col.lab=my.col[2], col.main=my.col[2])#

    hist(data$Omega[which(data$Omega < quantile(data$Omega,0.95))], main=paste("Omega"),xlab="Omega",breaks=30)
    hist(geneData$Omega, main=paste("log2(Omega)"),xlab="log2(Omega)", col=my.col[3],breaks=30)

    hist(data$CDS.Length[which(data$CDS.Length < quantile(data$CDS.Length,0.95))], main=paste("CDS length"), xlab="CDS length",breaks=30)
    hist(geneData$CDS.Length, main=paste("log2(CDS length)"), xlab="log2(CDS length)", col=my.col[3],breaks=30)

    hist(data$Intron.Number[which(data$Intron.Number<quantile(data$Intron.Number,0.95))], main=paste("Intron number"), xlab="Intron number",breaks=30)
    hist(geneData$Intron.Number, main=paste("log2(Intron number)"), xlab="log2(Intron number)", col=my.col[3],breaks=30)

    hist(data$Intron.Length[which(data$Intron.Length<quantile(data$Intron.Length,0.95))], main=paste("Intron length"), xlab="Intron length",breaks=30)
    hist(geneData$Intron.Length, main=paste("log2(Intron length)"), xlab="log2(Intron length)", col=my.col[3],breaks=30)

    hist(data$Tau, main=paste("Tau"), xlab="Tau",breaks=30)
    hist(geneData$Tau, main=paste("log2(Tau)"), xlab="log2(Tau)", col=my.col[3],breaks=30)

    hist(data$Paralogs.Number[which(data$Paralogs.Number<quantile(data$Paralogs.Number,0.95))], main=paste("Paralogs Number"), xlab="Paralogs Number",breaks=30)
    hist(geneData$Paralogs.Number, main=paste("log2(Paralogs Number)"), xlab="log2(Paralogs Number)", col=my.col[3],breaks=30)

    hist(data$P.1[which(data$P.1<quantile(data$P.1,0.95))], main=paste("P.1"), xlab="P.1",breaks=30)
    hist(geneData$P.1, main=paste("(P.1)^1/4"), xlab="(P.1)^1/4", col=my.col[3],breaks=30)

    hist(data$LRT, main=paste("LRT"), xlab="LRT",breaks=30)
    hist(geneData$LRT[which(data$LRT>quantile(data$LRT,0.95))], main=paste("(LRT)^1/4"), xlab="(LRT)^1/4", col=my.col[3],breaks=30)

    hist(data$Median.Expression, main=paste("Median Expression"), xlab="Median Expression",breaks=30)

    hist(data$Max.Expression, main=paste("Max Expression"), xlab="Max Expression",breaks=30)

    hist(data$X..GC.content, main=paste("%GC Content"), xlab="%GC Content",breaks=30)
    hist(geneData$X..GC.content, main=paste("log2(X..GC.content)"), xlab="log2(X..GC.content)", col=my.col[3],breaks=50)

    hist(data$Stage.Number, main=paste("Stage Number"), xlab="Stage Number",breaks=30)

    hist(data$Phyletic.Age, main=paste("Phyletic Age"), xlab="Phyletic Age",breaks=30)

    dev.copy2pdf(device=quartz, file=paste(folderAnalysis, organism,"Parameters", expDataSource, add,".pdf", sep=""),onefile=TRUE)#,paper="A4r"
    #dev.off()
########

#Names of the variables used
variableNames <- colnames(geneData)

#Calculating correlations
if (partial==TRUE)
```

```
{
    x <- data.frame(x1=NULL,x2=NULL,corValue=NULL,pValue=NULL,significant=NULL)
    for(j in variableNames) #j is the name of variable for which the correlation is calculated
    {
        variablesToUse <- variableNames[variableNames != j] #all other variables
        t = length(variableNames)-1

        for(n in c(1:t))
        {
            j2 <- variablesToUse[n]
            variablesToUse2 <- variablesToUse[variablesToUse != j2]

            fmodel <-"geneData$"
            fmodel <- paste(fmodel,j,"~",sep="")
            fmodel2 <-"geneData$"
            fmodel2 <- paste(fmodel2,j2,"~",sep="")
            for(i in variablesToUse2)
            {
                fmodel <- paste(fmodel,"geneData$",i,"+",sep="")
                fmodel2 <- paste(fmodel2,"geneData$",i,"+",sep="")
            }
            fmodel <- substr(fmodel, 1, nchar(fmodel)-1) #delet last character "+"
            fmodel2 <- substr(fmodel2, 1, nchar(fmodel2)-1) #delet last character "+"
            fmx <- glm(fmodel, na.action = na.exclude)
            fmy <- glm(fmodel2, na.action = na.exclude)

            xres <- resid(fmx)
            yres <- resid(fmy)
            ct <- cor.test(xres, yres, method=corMethod)

            s <- ct$estimate
            coeff <- ct$p.value
            signCoeff <- ct$p.value < 0.0005 #Treshhold for significance, corrected for 14 parameters
            x <- rbind(x, data.frame(x1=j,x2=j2,corValue=s,pValue=coeff,significant=as.integer(signCoeff)))
        }
    }
} else {
    x <- data.frame(x1=NULL,x2=NULL,corValue=NULL,pValue=NULL,significant=NULL)
    for(j in variableNames) #j is the name of variable for which the correlation is calculated
    {
        variablesToUse <- variableNames[variableNames != j] #all other variables
        t = length(variableNames)-1
        for(n in c(1:t))
        {
            j2 <- variablesToUse[n]
            variablesToUse2 <- variablesToUse[variablesToUse != j2]
            ct <- cor.test(geneData[,j], geneData[,j2], method=corMethod)
            s <- ct$estimate
            coeff <- ct$p.value
            signCoeff <- ct$p.value < 0.0005 #Treshhold for significance, corrected for 14 parameters, 91 correlation
            x <- rbind(x, data.frame(x1=j,x2=j2,corValue=s,pValue=coeff,significant=as.integer(signCoeff)))
        }
    }
}

row.names(x) <- c(1:nrow(x))

sameCorRowNumbers <- vector("numeric")
for(i in rownames(x))
{
    sameCor <- x[with(x,x$x2 == x[i,]$x1 & x$x1 == x[i,]$x2),]
    if(as.integer(rownames(sameCor))>as.integer(i))
    {
        sameCorRowNumbers <- append(sameCorRowNumbers, as.integer(rownames(sameCor)))
    }
}
x <- x[-sameCorRowNumbers,]

cat("\n Correlation table",sep="")
print(x, type="latex",file="")
cat("\n Result of the correlation is saved in \"Cor_Original\".",sep="")
write.table(x,file=paste(folderAnalysis, organism, part, corMethod, "Cor", expDataSource, "Original", add, ".txt", sep=""),row.names = FALSE,quote = FALSE)

xp <- x[x$x1=="Omega",]
xp <- xp[xp$significant >0,]
xp$var <- xp$corValue*xp$corValue
v <- sum(xp$var)*100
print(paste("The variance of Omega is explained to ",v, "% through used parameters",sep=""))

#Making the file for Cytoscape
x$abs <- abs(x$corValue)*20
x$sign <- sign(x$corValue)
x <- x[,c("x1","x2","significant","abs","sign")]

#Data to use with cytoscape
cat("\n Result of the correlation for Cytoscape representation is saved in \"Cor_List\".",sep="")
write.table(x, file=paste(folderAnalysis, organism, part, corMethod, "Cor", expDataSource,"List", add, ".txt",sep=""),row.names = FALSE,quote = FALSE)

###
#Draw graph in cytoscape

graphC <- x
cy <- CytoscapeConnection()

# initialize
g <- new ("graphNEL", edgemode = "undirected")
g <- initNodeAttribute (g, "nodeType", "char", "undefined")
g <- initNodeAttribute (g, "label", "char", "undefined")
g <- initEdgeAttribute (g, "edgeType", "char", "undefined")
g <- initEdgeAttribute (g, "significant", "char", "undefined")
g <- initEdgeAttribute (g, "sign", "char", "undefined")
#g <- initEdgeAttribute (g, "label", "char", "undefined")

#add nodes and edges
    g <- addNode("info.node", g)
    #g <- addNode("title.node", g)
parameters <- unique(levels(graphC$x1))
for (p in parameters){
    g <- addNode(p, g)
}

for (n in 1:length(rownames(graphC))){
    g <- addEdge(as.character(graphC[n,1]), as.character(graphC[n,2]), g)
}

#add node and edge attribues
    nodeData(g, "info.node", "label") = "Information I want"
    #nodeData(g, "title.node", "label") = "Information I want manny majniofjfpwef ndiojfpewfjwe mofpewjkfoeqüwpfkjoweü meowpfjkoü"
for (p in parameters){
    nodeData(g, p, "nodeType") = p
    nodeData(g, p, "label") = p
}
nodeData(g, parameters, "label") = parametersNames

for (n in 1:length(rownames(graphC))){
    edgeData(g, as.character(graphC[n,1]), as.character(graphC[n,2]), "edgeType") = as.character(graphC[n,4])
    edgeData(g, as.character(graphC[n,1]), as.character(graphC[n,2]), "sign") = graphC[n,5]
    edgeData(g, as.character(graphC[n,1]), as.character(graphC[n,2]), "significant") = graphC[n,3]
```

```
    #edgeData(g, as.character(graphC[n,1]), as.character(graphC[n,2]), "label") = round(graphC[n,4]/20*graphC[n,5], digits=2)
}

#create a CytoscapeWindow, after first making sure that no prior window of the same name

cy <- CytoscapeConnection()
setDefaultBackgroundColor(cy, my.col[1])
    window.title = 'Correlation'
    if (window.title %in% as.character (getWindowList(cy)))
      deleteWindow (cy, window.title)
    cw <- new.CytoscapeWindow (window.title, g)

# set window and network sizes
setWindowSize (cw, 1200, 1200)
fitContent (cw)
setZoom (cw, 0.9 * getZoom (cw))

#send graph to Cytoscape
displayGraph (cw)

#Set default settings for the graph
setDefaultEdgeColor(cw, my.col[2])
lockNodeDimensions(cw, FALSE)
setNodeShapeDirect(cw, parameters, "ellipse")
setNodeFontSizeDirect(cw, parameters, 10)
setNodeColorDirect(cw, parameters, my.col[7])
setNodeWidthDirect(cw, parameters, 85)
setNodeHeightDirect(cw, parameters, 40)

    #Legend
    setNodeShapeDirect(cw, "info.node", "rect")
    setNodeFontSizeDirect(cw, "info.node", 10)
    setNodeColorDirect(cw, "info.node", my.col[1])
    setNodeWidthDirect(cw, "info.node", 135)
    setNodeHeightDirect(cw, "info.node", 135)
    setNodeImageDirect(cw,"info.node", "file:/Legend.png")
    setNodeBorderColorDirect(cw, "info.node", my.col[1])
    setNodeOpacityDirect(cw, "info.node", 0)

#ask Cytoscape to layout the graph
layoutNetwork (cw, 'attribute-circle')

#instruct Cytoscape to use each node's 'label' attribute as the value for the visible label it draws on the node
setNodeLabelRule (cw, 'label')

setEdgeLineWidthRule(cw, "edgeType", as.character(graphC$abs), as.numeric(graphC$abs))
setEdgeColorRule(cw, "sign", c("-1", "1"), c("blue", "red"), mode="lookup")
setEdgeOpacityRule(cw, "significant", c("1", "0"), c("175", "0"), mode="lookup")
#setEdgeLabelRule(cw, "label")

# now ask Cytoscape to redraw the graph using these rules
redraw (cw)

#saveLayout(cw,'CorrelationLayout13') #Manually change the order of parameters
restoreLayout(cw, 'CorrelationLayout13')
fitContent (cw)

saveImage(cw, paste(folderAnalysis, organism, part, corMethod, "Cor", expDataSource, "Cyt", add, ".png", sep=""), 'png', 2.0)

#########################
#########################
#########################


#########################################################
#Organism partial correlation with GLM model for circos#
#########################################################

cat("\n Calculating partial correlation with glm model for each tissue separately.",sep="")

#Load the data

dataOrg <- read.table(paste(folderAnalysis, organism, "TableTissues", expDataSource,".txt",sep=""), header=TRUE)

add <- ""

cat("\n Summary of the data (", nrow(dataOrg), " genes) in the first step: ", sep="")
summary(dataOrg)
dataOrg <- dataOrg[dataOrg$Max.Expression>0.00015,]

        # ###Only essential human-mouse orthologs
        # dataOrg <- dataOrg[dataOrg$Essentiality==1,]
        # dataOrg <- dataOrg[!is.na(dataOrg$Essentiality),]

        # ###Only specific genes
        # dataOrg <- dataOrg[dataOrg$Tau>0.2,]
        # dataOrg <- dataOrg[!is.na(dataOrg$Tau),]
        # summary(dataOrg)

cat("\n To calculate correlation ", corMethod, " correlation was used.",sep="")
if (partial)
{
    cat("Partial correlation was performed.",sep="")
    part <- "Partial"
} else {
    cat("Normal correlation was performed.",sep="")
    part <- "Normal"
}

cat("\n The analysis is done for ", length(tissuesRPKMNames), " tissues.", sep="")

###All parameters
#Leave only needed columns
dataOrg <- dataOrg[,c("Omega.0", "LRT", "P.1","CDS.Length", "Intron.Length", "Intron.Number", "X..GC.content", "Paralogs.Number", "Stage.Number", "Phyletic.Age", tissuesRPKMNames)]
parameterNames <- c("Omega", "LRT", "P.1", "CDS.Length", "Intron.Length", "Intron.Number", "X..GC.content", "Paralogs.Number", "Stage.Number", "Phyletic.Age")

colnames(dataOrg) <- c(parameterNames, tissuesRPKMNames)

geneDataOrg <- na.omit(dataOrg)

cat("\n All the genes with unknown parameters are removed from the analysis. ", nrow(geneDataOrg)," are left for the analysis.", " Summary: ", sep="")
summary(geneDataOrg)

minOmega <- min(geneDataOrg$Omega[geneDataOrg$Omega>0])
geneDataOrg$Omega <- geneDataOrg$Omega + minOmega
geneDataOrg$Omega <- log2(geneDataOrg$Omega)

minLength <- min(geneDataOrg$Intron.Length[geneDataOrg$Intron.Length>0])
geneDataOrg$Intron.Length <- geneDataOrg$Intron.Length + minLength
geneDataOrg$Intron.Length <- log2(geneDataOrg$Intron.Length)

geneDataOrg$LRT <- ifelse(geneDataOrg$LRT<0, 0, geneDataOrg$LRT)
geneDataOrg$LRT <- sqrt(sqrt(geneDataOrg$LRT))

minP1 <- min(geneDataOrg$P.1[geneDataOrg$P.1>0])
geneDataOrg$P.1 <- geneDataOrg$P.1 + minP1
geneDataOrg$P.1 <- sqrt(sqrt(geneDataOrg$P.1))
```

```
geneDataOrg$Intron.Number <- geneDataOrg$Intron.Number + 1
geneDataOrg$Intron.Number <- log2(geneDataOrg$Intron.Number)

geneDataOrg$Paralogs.Number <- geneDataOrg$Paralogs.Number + 1
geneDataOrg$Paralogs.Number <- log2(geneDataOrg$Paralogs.Number)

geneDataOrg$Phyletic.Age <- geneDataOrg$Phyletic.Age + 1

geneDataOrg$CDS.Length <- log2(geneDataOrg$CDS.Length)

cat("\n Graphical representation of the expression data is saved in the file \"TissuesExpression\".", sep="")

    #Run separately in R, cannot draw from LaTeX
        dev.new(height=9, width=12)
        par(cex.main=0.95, bg=my.col[1], fg=my.col[2], col.axis=my.col[2], col.lab=my.col[2], col.main=my.col[2])
        palette(rev(rich.colors(length(tissuesNames)+2)))

        plot(density(geneDataOrg[,tissuesRPKMNames[1]],n=1000), main = "Expression values among different tissues",xlab="Normalized RPKM",col=(1), lwd=3)
        for(i in c(2:length(tissuesRPKMNames)))
        {
            lines(density(geneDataOrg[,tissuesRPKMNames[i]],n = 1000), col=(i), lwd=3)
        }
        legend("topright",tissuesPrintNames,col=(1:length(tissuesRPKMNames)),lty="solid", lwd=3)

        dev.copy2pdf(device=quartz, file=paste(folderAnalysis, organism,"TissuesExpression", expDataSource, add,".pdf", sep=""),onefile=TRUE)#,paper="A4r"
        #dev.off()

cat("\n Overall ",nrow(geneDataOrg)," genes were used for analysis."," Summary:",sep="")
summary(geneDataOrg)

variableNames <- tissuesRPKMNames

##############
##Calculation correlation
    x <- data.frame(x1=NULL,x2=NULL,corValue=NULL,pValue=NULL,significant=NULL)
    for(j in variableNames) #j is the name of variable for which the correlation is calculated
    {
        variablesToUse <- parameterNames #Names of other variables
        t = length(variablesToUse)
        for(n in c(1:t))
        {
            j2 <- variablesToUse[n]
            variablesToUse2 <- variablesToUse[variablesToUse != j2]

            if(partial==TRUE)
            {
                fmodel <-"geneDataOrg$"
                fmodel <- paste(fmodel,j,"~",sep="")
                fmodel2 <-"geneDataOrg$"
                fmodel2 <- paste(fmodel2,j2,"~",sep="")
                for(i in variablesToUse2)
                {
                    fmodel <- paste(fmodel,"geneDataOrg$",i,"+",sep="")
                    fmodel2 <- paste(fmodel2,"geneDataOrg$",i,"+",sep="")
                }
                fmodel <- substr(fmodel, 1, nchar(fmodel)-1) #delet last character "+"
                fmodel2 <- substr(fmodel2, 1, nchar(fmodel2)-1) #delet last character "+"
                fmx <- glm(fmodel, na.action = na.exclude)
                fmy <- glm(fmodel2, na.action = na.exclude)

                xres <- resid(fmx)
                yres <- resid(fmy)

                ct <- cor.test(xres, yres, method=corMethod)
            } else {
                ct <- cor.test(geneDataOrg[,j], geneDataOrg[,j2], method=corMethod)
            }

            s <- ct$estimate
            coeff <- ct$p.value
            signCoeff <- ct$p.value < 0.0005
            x <- rbind(x, data.frame(x1=j,x2=j2,corValue=s,pValue=coeff,significant=as.integer(signCoeff)))
        }
    }
    variableNames <- parameterNames
    for(j in variableNames) #j is the name of variable for which the correlation is calculated
    {
        variablesToUse <- variableNames[variableNames != j] #Names of other variables
        t = length(variablesToUse)
        for(n in c(1:t))
        {
            j2 <- variablesToUse[n]
            variablesToUse2 <- variablesToUse[variablesToUse != j2]
            if(partial==TRUE)
            {
                fmodel <-"geneDataOrg$"
                fmodel <- paste(fmodel,j,"~",sep="")
                fmodel2 <-"geneDataOrg$"
                fmodel2 <- paste(fmodel2,j2,"~",sep="")
                for(i in variablesToUse2)
                {
                    fmodel <- paste(fmodel,"geneDataOrg$",i,"+",sep="")
                    fmodel2 <- paste(fmodel2,"geneDataOrg$",i,"+",sep="")
                }
                fmodel <- substr(fmodel, 1, nchar(fmodel)-1) #delet last character "+"
                fmodel2 <- substr(fmodel2, 1, nchar(fmodel2)-1) #delet last character "+"
                fmx <- glm(fmodel, na.action = na.exclude)
                fmy <- glm(fmodel2, na.action = na.exclude)

                xres <- resid(fmx)
                yres <- resid(fmy)
                ct <- cor.test(xres, yres, method=corMethod)
            } else {
                ct <- cor.test(geneDataOrg[,j], geneDataOrg[,j2], method=corMethod)
            }

            s <- ct$estimate
            coeff <- ct$p.value
            signCoeff <- ct$p.value < 0.0005
            x <- rbind(x, data.frame(x1=j,x2=j2,corValue=s,pValue=coeff,significant=as.integer(signCoeff)))
        }
    }
    row.names(x) <- c(1:nrow(x))

    x$x2 <- factor(x$x2,levels=levels(x$x1))
    sameCorRowNumbers <- vector("numeric")
    for(i in rownames(x))
    {
        sameCor <- x[with(x,x$x2 == x[i,]$x1 & x$x1 == x[i,]$x2),]
        if(nrow(sameCor)>0)
        {
            if(as.integer(rownames(sameCor))>as.integer(i))
            {
                sameCorRowNumbers <- append(sameCorRowNumbers, as.integer(rownames(sameCor)))
            }
        }
    }
    x <- x[-sameCorRowNumbers,]
```

```
cat("\n Correlation table",sep="")
print(x, type="latex",file="",append=FALSE)
cat("\n Result of the correlation is saved in \"CorTissues_Original\".",sep="")
write.table(x,file=paste(folderAnalysis, organism, part, corMethod, "CorTissues", expDataSource, "Original", add,".txt",sep=""),row.names = FALSE,quote = FALSE)
############################

nTissues <- length(tissuesNames)

correctionTerm <- (length(parameterNames)-1)*nTissues
cat("\n Expression data are sorted according to correlation with Omega.",sep="")
#Sorting expression parameters according to correlation with Omega
    x.exp <- x[(regexpr("Averaged.RPKM.",x$x1)+regexpr("Averaged.RPKM.",x$x2))==0,]#data frame with expression correlations
    x.exp$abs <- abs(x.exp$corValue)
    x.exp <- x.exp[with(x.exp, order(x.exp$x2, x.exp$corValue, decreasing=TRUE)),]
    row.names(x.exp) <- c(1:nrow(x.exp))
    #x.exp$exp.order <- row.names(x.exp)
    nRow <- nrow(x.exp)
    x.exp <- x.exp[with(x.exp,x.exp$x2=="Omega"),]#Parameter used for sorting expression data#
    x.exp$exp.order <- c((nRow-nTissues+1):nRow)
    #x.exp <- x.exp[(nrow(x.exp)-nTissues+1):nrow(x.exp),]
    x.exp <- x.exp[,c("x1","exp.order")]
    x2 <- merge(x,x.exp,by=c("x1"), all.x=TRUE, sort=FALSE)

    for(i in c(1:nrow(x2)))
    {
        if(regexpr("Averaged.RPKM.",x2$x1[i])>0) #If it is expression in the first row
        {
            resA <- lapply(strsplit(as.character(x2$x1[i]), split=".", fixed=TRUE),function(x){x[1]})
            resR <- lapply(strsplit(as.character(x2$x1[i]), split=".", fixed=TRUE),function(x){x[2]})
            resN <- ifelse(as.integer(x2$exp.order[i])<10, paste("0",x2$exp.order[i],sep=""), x2$exp.order[i])
            resT <- lapply(strsplit(as.character(x2$x1[i]), split=".", fixed=TRUE),function(x){x[3]})

            x2$x1a[i] <- paste(resA,".",resR,".",resN,".",resT,sep="")
        }
    }

    x2$x1 <- x2$x1a
    x2 <- x2[,c("x1","x2","corValue")]
    x3 <- merge(x,x2,by=c("x2","corValue"), all.x=TRUE, sort=FALSE)
    for(i in c(1:nrow(x3)))
    {
        if(regexpr("Averaged.RPKM.",x3$x1.x[i])<0) #If it is not an expression in the first row
        {
            x3$x1.y[i] <- as.character(x3$x1.x[i])
        }
    }
    x3 <- x3[,c("x1.y","x2","corValue","pValue","significant")]
    names(x3) <- c("x1","x2","corValue","pValue","significant")

    x <- x3
    x <- x[with(x, order(x$x1, decreasing=TRUE)),]

    v.temp <- data.frame(values=NULL)
    v.temp <- rbind(v.temp,data.frame(v=x$x1))
    v.temp <- rbind(v.temp,data.frame(v="Omega"))
    variables <- unique(v.temp)#All expression variables
    variables <- variables[with(variables, order(variables$v)),]
    x$x1 <- factor(x$x1,levels=levels(variables))
#Finish sorting

xtemp1 <- x
x$abs <- abs(x$corValue) # was 20 for cytoscype
x$sign <- sign(x$corValue)
x <- x[,c("x1","x2","significant","abs","sign")]

#Table with corralation data
cat("\n Result is saved in \"CorTissues_List\". ",sep="")
write.table(x,file=paste(folderAnalysis, organism, part, corMethod, "CorTissues", expDataSource, "List", add,".txt",sep=""),row.names = FALSE,quote = FALSE)

x <- xtemp1
x <- x[with(x, order(x$x1, decreasing=TRUE)),]
row.names(x) <- c(1:nrow(x))
v.temp <- data.frame(values=NULL)

for(i in c(1:nrow(x)))
{
    if(regexpr("Averaged.RPKM.",x$x1[i])>0) #If it is expression in the first row
    {
        res <- lapply(strsplit(as.character(x$x1[i]), split=".", fixed=TRUE),function(x){x[3]})
        x$order[i] <- as.integer(res)-1-correctionTerm
        v.temp <- rbind(v.temp,data.frame(variables=x$x1[i]))
    }
    else if(regexpr("Averaged.RPKM.",x$x2[i])>0) #If it is expression in the second column
    {
        res <- lapply(strsplit(as.character(x$x1[i]), split=".", fixed=TRUE),function(x){x[3]})
        x$order[i] <- as.integer(res)-1
        v.temp <- rbind(v.temp,data.frame(variables=x$x2[i]))
    }
    else
    {
        x$order[i] <- 0
    }
}

######CORRELATIONS HERE
x$abs <- as.integer(abs(x$corValue)*1000) #Correlation strength, width of the lines
x$sgn <- paste("color=c", sign(x$corValue)+1,x$significant,sep="") #Correlation positive or negative and if significant, "color=c11" which describe the color

variables <- levels(x$x1) #All variables used for correlations
v.temp <- data.frame(values=NULL)
v.temp <- rbind(v.temp,data.frame(v=x$x1))
v.temp <- rbind(v.temp,data.frame(v=x$x2))
variables <- unique(v.temp)#All expression variables
variables <- variables[with(variables, order(variables$v)),]

x$x1x <- 0
x$x1y <- 0
x$x2x <- 0
x$x2y <- 0

x.exp <- x[(regexpr("Averaged.RPKM.",x$x1)+regexpr("Averaged.RPKM.",x$x2))==0,]#data frame with expression correlations
x.ne <- x[(regexpr("Averaged.RPKM.",x$x1)+regexpr("Averaged.RPKM.",x$x2))<0,]#data frame with other correlations

#Names of not exprssion variables
v.temp <- data.frame(values=NULL)
v.temp <- rbind(v.temp,data.frame(variables=x.exp$x1))
variables.exp <- unique(v.temp)#All expression variables
variables.exp <- variables.exp[with(variables.exp, order(variables.exp$variables)),]

variables.ne <- parameterNames

v <- vector(mode="numeric") #Vector with the length for each segment, used later in gaps
#Calculating the maximum needed width for expression segment and sum of all segments
    maxL=0
    for(j in variables)
    {
        s <- (sum(x[x$x1==j,]$abs)+sum(x[x$x2==j,]$abs))*2 #Segment should be twice as long as width of all correlations
```

```
        if(maxL < s)
        {
            maxL <- s
        }
        v <- append(v, s)
    }
#Calclulating the maximum of expression segments and the length for each expression segment
    v.exp <- vector(mode="numeric") #used later in bands, contain weidth of each expression segment
    maxL.exp=0
    for(j in variables.exp)
    {
        s <- (sum(x.exp[x.exp$x1==j,]$abs)+sum(x.exp[x.exp$x2==j,]$abs))*4 #was 2 before, now 4 for Poster #Segment should be twice as long as width of all correlations
        if(maxL.exp < s)
        {
            maxL.exp <- s
        }
        v.exp <- append(v.exp, s)
    }

#Table for Karyotype file
    kar <- data.frame(variables=variables.ne, maxL)#Not expression variables and segment length. data.frame with "variables" and "maxL"

    kar <- rbind(kar,data.frame(variables="Expression",maxL=maxL.exp*length(v.exp))) #Length of the big expression segment. Max expression length * number of
    gaps <- data.frame(variables, maxL, v)#All parameters with maximal length (maxL) and own length (v)

    kar$name <- paste ("chr","\t","-","\t", kar$variables,sep="")#Adding “chr - “ to the variables #Make error in LaTeX
    kar$x <- 0 #Adding x column with 0, starting points
    kar$color <- "chr1" #Adding color parameter
    kar <- kar[, c("name", "variables", "x", "maxL", "color")] #puting colums in the right order
    kar <- kar[with(kar, order(kar$variables, decreasing=TRUE)),] #puting rows in the right order

#Calculating the exact coordinates for each band of expression
    band <- data.frame(variables=NULL, start=NULL, end=NULL)
    a.p = 0
    for(i in c(1:length(v.exp)))
    {
        band <- rbind(band,data.frame(variables=variables.exp[i],start=a.p+1,end=a.p + maxL.exp))#calculate the coordinates for each segment of expression
        a.p=a.p + maxL.exp
    }

    band$start[1]=0
    band$c <- paste("band","\t","Expression","\t",band$variables,"\t",band$variables,sep="") #data.frame with variables, start and end point and c("band Expression"+variable names)
    band$color <- "chr2" #color column

#Names of parameters for labeling the graph
    labels <- data.frame(name="Expression",start=band$start, end=band$end) #data.frame with "Expression" and start and end of each segment
    #Creating labels for each segment
    for(i in c(1:nrow(labels)))
    {
        res <- lapply(strsplit(as.character(band$variables[i]), split=".", fixed=TRUE),function(x){x[4]})
        labels$label[i] <- as.character(res)
    }

###########ONTOLOGY###############

        ##divers >>#chr2
        ##gastrointestinal system >>#chr4
        ##central nervous system >>#chr3
        ##reproductive system >>#chr5

        #salivary(salivary gland) > oral region > gastrointestinal system > visceral organ >>#chr4
            #sgland == salivary
        #stomach > gastrointestinal system > visceral organ >> #chr4
        #duodenum > intestine > gastrointestinal system (first section of the small intestine) > visceral organ >>#chr4
        #smintestine(small intestine) > intestine > gastrointestinal system > visceral organ >>#chr4
        #lgintestine(large intestine) > intestine > gastrointestinal system > visceral organ >>#chr4
        #colon > large intestine > intestine > gastrointestinal system > visceral organ >>#chr4
        #appendix > large intestine > intestine > gastrointestinal system > visceral organ >>#chr4
        #esophagus > gastrointestinal system > visceral organ >> #chr4

        #cerebellum > forebrain > brain > central nervous system >>#chr3
        #flobe(frontal lobe) > central nervous system >>#chr3
        #tlobe(temporal lobe) > central nervous system >>#chr3
        #cortex > forebrain > brain > central nervous system >>#chr3
        #fcortex(frontal cortex) > cortex > farebrain > brain > central nervous system >>#chr3
        #pcortex(prefrontal cortex) > cortex > farebrain > brain > central nervous system >>#chr3
        #brain > central nervous system >>#chr3
        #hcampus(Hippocampus) > telencephalon > forebrain > brain > central nervous system >>#chr3
        #spinal(Spinal Cord) > central nervous system >>#chr3
        #hypothalamus > diencephalon > forebrain > brain > central nervous system >>#chr3
        #pituitary(Pituitary Gland) > diencephalon gland > diencephalon > forebrain > brain > central nervous system >>#chr3

        #gfat (Genital Adipose Tissue) > seminal vesicle > male reproductive system > reproductive system >>#chr5
        #ovary > reproductive system >>#chr5
        #placenta > reproductive system > visceral organ >>#chr5
        #testis > reproductive system > visceral organ >>#chr5
        #prostate > reproductive system > visceral organ >>#chr5
        #endometrium > uterus > reproductive system > visceral organ >>#chr5
        #uterus > reproductive system > visceral organ >>#chr5

        #thymus > haemolymphoid system >>#chr2
        #spleen > haemolymphoid system >>#chr2
        #lymphnode > haemolymphoid system >>#chr2

        #adrenal > endocrine system >>#chr2
        #pancreas > endocrine system >>#chr2

        #kidney > renal-urinal system > visceral organ >>#chr2
        #bladder > renal-urinal system > visceral organ >>#chr2
            #ubladder == bladder

        #lung > respiratory system > visceral organ >>#chr2

        #liver > liver and biliary system > visceral organ >>#chr2
        #gbladder (gallbladder) > liver and biliary system > visceral organ >>#chr2

        #heart > cardiovascular system >>#chr2
        #blood > cardiovascular system >>#chr2

        #mamgland(mamary gland) > integumental system gland > integumental system >>#chr2
        #skin > integumental system >>#chr2

        #sfat (Subcutaneous Fat Pad (Subcutaneous Adipose Tissue)) > Adipose Tissue >>#chr2
            #fat == sfat

        #muscle >>#chr2

        #marrow(Bone Marrow) > bone > sceletal system >>#chr2
            #bonem == marrow

        #thyroid > foregut > gut > alimentary system > visceral organ >>#chr2

    for(i in c(1:nrow(band)))
    {
        if(regexpr("flobe",band$variables[i])>0)
        {
            band$color[i] <- "chr3"
            labels$label[i] <- "Frontal Lobe"
```

```
}
else if(regexpr("cerebellum",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Cerebellum"
}
else if(regexpr("brain",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Brain"
}
else if(regexpr("fcortex",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Frontal Cortex"
}
else if(regexpr("pcortex",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Prefrontal Cortex"
}
else if(regexpr("tlobe",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Temporal Lobe"
}
else if(regexpr(".cortex",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Cortex"
}
else if(regexpr("hcampus",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Hippocampus"
}
else if(regexpr("spinal",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Spinal Cord"
}
else if(regexpr("hypothalamus",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Hypothalamus"
}
else if(regexpr("pituitary",band$variables[i])>0)
{
    band$color[i] <- "chr3"
    labels$label[i] <- "Pituitary Gland"
}
######
else if(regexpr("colon",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Colon"
}
else if(regexpr("stomach",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Stomach"
}
else if(regexpr("smintestine",band$variables[i])>0 | regexpr("smint",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Small Intestine"
}
else if(regexpr("duodenum",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Duodenum"
}
else if(regexpr("lgintestine",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Large Intestine"
}
else if(regexpr("salivary",band$variables[i])>0 | regexpr("sgland",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Salivary Gland"
}
else if(regexpr("appendix",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Appendix"
}
else if(regexpr("esophagus",band$variables[i])>0)
{
    band$color[i] <- "chr4"
    labels$label[i] <- "Esophagus"
}
#####
else if(regexpr("ovary",band$variables[i])>0)
{
    band$color[i] <- "chr5"
    labels$label[i] <- "Ovary"
}
else if(regexpr("gfat",band$variables[i])>0)
{
    band$color[i] <- "chr5"
    labels$label[i] <- "Genital Fat Pad"
}
else if(regexpr("testis",band$variables[i])>0)
{
    band$color[i] <- "chr5"
    labels$label[i] <- "Testis"
}
else if(regexpr("placenta",band$variables[i])>0)
{
    band$color[i] <- "chr5"
    labels$label[i] <- "Placenta"
}
else if(regexpr("prostate",band$variables[i])>0)
{
    band$color[i] <- "chr5"
    labels$label[i] <- "Prostate"
}
else if(regexpr("endometrium",band$variables[i])>0)
{
    band$color[i] <- "chr5"
    labels$label[i] <- "Endometrium"
}
else if(regexpr("uterus",band$variables[i])>0)
{
    band$color[i] <- "chr5"
    labels$label[i] <- "Uterus"
}
```

```
        #####
        else if(regexpr("thymus",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Thymus"
        }
        #####
        else if(regexpr("sfat",band$variables[i])>0 | (regexpr("fat",band$variables[i])>0 & regexpr("gfat",band$variables[i])<0))
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Fat Pad"
        }
        #####
        else if((regexpr("bladder",band$variables[i])>0 & regexpr("gbladder",band$variables[i])<0) | regexpr("ubladder",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Bladder"
        }
        else if(regexpr("gbladder",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Gallbladder"
        }
        #####
        else if(regexpr("mamgland",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Mammary Gland"
        }
        #####
        else if(regexpr("skin",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Skin"
        }
        #####
        else if(regexpr("lung",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Lung"
        }
        #####
        else if(regexpr("adrenal",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Adrenal"
        }
        #####
        else if(regexpr("heart",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Heart"
        }
        #####
        else if(regexpr("blood",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Blood"
        }
        #####
        else if(regexpr("kidney",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Kidney"
        }
        #####
        else if(regexpr("spleen",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Spleen"
        }
        #####
        else if(regexpr("liver",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Liver"
        }
        #####
        else if(regexpr("muscle",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Muscle"
        }
        #####
        else if(regexpr("marrow",band$variables[i])>0 | regexpr("bonem",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Bone Marrow"
        }
        #####
        else if(regexpr("pancreas",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Pancreas"
        }
        #####
        else if(regexpr("thyroid",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Thyroid"
        }
        #####
        else if(regexpr("lymphnode",band$variables[i])>0)
        {
            band$color[i] <- "chr2"
            labels$label[i] <- "Lymph Node"
        }
    }

###########END######ontology############

band <- band[, c("c", "start", "end", "color")] #Puting colums in the right order

#Data to use with circos
cat("\n Correlation result for representation in circos are saved in \"Karyotype\", \"Labels\" and \"Links\".",sep="")
write.table(kar,file=paste(folderAnalysis, organism, part, corMethod, "Cor", expDataSource, "Karyotype", add,".txt",sep=""),row.names = FALSE, col.names=FALSE, quote = FALSE,sep="\t")
 #_GLM_karyotype.txt
write.table(band,file=paste(folderAnalysis, organism, part, corMethod, "Cor", expDataSource, "Karyotype", add,".txt",sep=""), row.names = FALSE, col.names=FALSE, quote = FALSE, append =
 TRUE,sep="\t")#_GLM_karyotype.txt
write.table(labels,file=paste(folderAnalysis, organism, part, corMethod, "Cor", expDataSource, "Labels", add, ".txt",sep=""), row.names = FALSE, col.names=FALSE, quote =
 FALSE,sep="\t")#_GLM_labels.txt

#Puting maximal expression value for the expression segments
for(i in c(1:nrow(gaps)))
{
    if(regexpr("Averaged.RPKM.",gaps$variables[i])>0)
    {
```

```
        gaps$maxL[i] <- maxL.exp
    }
}
temp_x <- x

x <- temp_x
#Calculating the exact possitions of the links inside the segment
for(k in c(1:nrow(gaps)))
{
    j <- gaps[k,]$variables
    if(regexpr("Averaged.RPKM.",gaps[k,]$variables)>0) #for expression segments
    {
        m <- (gaps[k,]$maxL-gaps[k,]$v)/(2*length(parameterNames)) #m is the spacing between correlations
    }
    else #for not expression segments
    {
        m <- (gaps[k,]$maxL-gaps[k,]$v)/(2*(nTissues+length(parameterNames)-1))
    }
    n = 0
    for(i in c(1:nrow(x)))
    {
        if (as.character(x[i,]$x1)==as.character(j))
        {
            n <- n+ x[i,]$abs/2+m #spacing
            x[i,]$x1x <- n  #beginning of the link
            n <- n+ x[i,]$abs
            x[i,]$x1y <- n  #end of the link
            n <- n+ x[i,]$abs/2+m #spacing
        }
        if (as.character(x[i,]$x2)==as.character(j))
        {
            n <- n+ x[i,]$abs/2+m
            x[i,]$x2x <- n
            n <- n+ x[i,]$abs
            x[i,]$x2y <- n
            n <- n+ x[i,]$abs/2+m
        }
    }
}

x$x1x <- as.integer(x$x1x)
x$x1y <- as.integer(x$x1y)
x$x2x <- as.integer(x$x2x)
x$x2y <- as.integer(x$x2y)
#Finish calculating the positions

x.temp <- x

#Recalculating the positions for the links for expression according to the big Expression segment
for(i in c(1:nrow(x)))
{
    if(regexpr("Averaged.RPKM.",x$x1[i])>0) #if expression parameter is in the 1. column
    {
        x$x1x[i] <- x$x1x[i] + maxL.exp*x$order[i]
        x$x1y[i] <- x$x1y[i] + maxL.exp*x$order[i]
    }
    else if(regexpr("Averaged.RPKM.",x$x2[i])>0) #if the expression parameter is in the 2. column
    {
        x$x2x[i] <- x$x2x[i] + maxL.exp*x$order[i]
        x$x2y[i] <- x$x2y[i] + maxL.exp*x$order[i]
    }
}

#Renaming all expression segment into "Expression"
for(i in c(1:nrow(x)))
{
    if(regexpr("Averaged.RPKM.",x$x1[i])>0)
    {
        x$x1a[i] <- "Expression"
    }
    else
    {
        x$x1a[i] <- as.character(x$x1[i])
    }
}
x$x1 <- x$x1a
for(i in c(1:nrow(x)))
{
    if(regexpr("Averaged.RPKM.",x$x2[i])>0)
    {
        x$x2a[i] <- "Expression"
    }
    else
    {
        x$x2a[i] <- as.character(x$x2[i])
    }
}
x$x2 <- x$x2a

x <- x[,c("x1","x1x","x1y","x2","x2x","x2y","sgn")] #Chousing the right columns#Data to use with circos (together with karyotype)


write.table(x,file=paste(folderAnalysis, organism, part, corMethod, "Cor", expDataSource, "Links", add, ".txt",sep=""), row.names = FALSE, col.names=FALSE, quote =
 FALSE,sep="\t")#_GLM_links.txt
################################
################################
################################



################################################
#Essentiality and Omega in Violin Plot & T-test#
################################################

data <- read.table(paste(folderAnalysis, organism, "Table", expDataSource, ".txt",sep=""), header=TRUE)

data <- data[data$Max.Expression>0.00015,]

data <- data[,c("Omega.0", "LRT", "P.1","CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number", "Stage.Number",
 "Phyletic.Age", "Essentiality")]
colnames(data) <- c("Omega",  "LRT", "P.1", "CDS.Length", "Intron.Length", "Intron.Number", "Median.Expression", "Max.Expression", "Tau", "X..GC.content", "Paralogs.Number",
 "Stage.Number", "Phyletic.Age", "Essentiality")

data <- na.omit(data)
geneData<-data

#Normalization of the data
minOmega <- min(geneData$Omega[geneData$Omega>0])
geneData$Omega <- geneData$Omega + minOmega
geneData$Omega <- log2(geneData$Omega)

minLength <- min(geneData$Intron.Length[geneData$Intron.Length>0])
geneData$Intron.Length <- geneData$Intron.Length + minLength
geneData$Intron.Length <- log2(geneData$Intron.Length)

minP1 <- min(geneData$P.1[geneData$P.1>0])
geneData$P.1 <- geneData$P.1 + minP1
geneData$P.1 <- sqrt(sqrt(geneData$P.1))
```

```
geneData$Intron.Number <- geneData$Intron.Number + 1
geneData$Intron.Number <- log2(geneData$Intron.Number)

geneData$Paralogs.Number <- geneData$Paralogs.Number + 1
geneData$Paralogs.Number <- log2(geneData$Paralogs.Number)

geneData$Phyletic.Age <- geneData$Phyletic.Age + 1

minTau <- 1 - max(geneData$Tau)
geneData$Tau <- geneData$Tau + minTau
geneData$Tau <- log2(geneData$Tau)

geneData$LRT <- ifelse(geneData$LRT<0, 0, geneData$LRT)
geneData$LRT <- sqrt(sqrt(geneData$LRT))

geneData$CDS.Length <- log2(geneData$CDS.Length)

summary(geneData)

#Names of the variables used
variableNames <- colnames(geneData)
variablesToUse <- variableNames[variableNames != c("Omega", "Essentiality")] #all other variables
fmodel <-"geneData$"
fmodel <- paste(fmodel,"Omega","~",sep="")
for(i in variablesToUse)
{
    fmodel <- paste(fmodel,"geneData$",i,"+",sep="")
}
fmodel <- substr(fmodel, 1, nchar(fmodel)-1) #delet last character "+"
fmx <- glm(fmodel, na.action = na.exclude)
xres <- resid(fmx)

dataPlot <- data.frame(Omega=xres, Essentiality=geneData$Essentiality)

dataP <- as.matrix(dataPlot[,c("Omega", "Essentiality")])

dev.new(height=8, width=8)
palette(rainbow(9))
trellis.par.set(list(background=list(col=my.col[1]), add.text=list(col=my.col[2], cex=1.5),axis.line=list(col=my.col[2]), axis.text=list(col=my.col[2], cex=1.5),
 par.main.text=list(col=my.col[2], cex=1.3), par.xlab.text=list(col=my.col[2], cex=1.5), par.ylab.text=list(col=my.col[2], cex=1.7), plot.line=list(col=my.col[2]), dot.line=list(lwd=1,
 lty=2, col="#4B4B4B"))) #trellis.par.get()

bwplot(as.numeric(dataP[,1])~dataP[,2], xlab="", ylab="residuals of log2(Omega)", main=paste("Distribution of Omega according to essentiality",sep=""), horizontal=FALSE,
col = c("#00BFFF"),
fill=c("blue"),
panel = function(x,y,..., box.ratio, col, pch){
    panel.violin(x=x, y=y,...,  cut = 0, varwidth = TRUE, box.ratio = 4*box.ratio, col=col)
    panel.bwplot(x=x, y=y, ..., varwidth = TRUE ,box.ratio = .5,  pch='|', notch=TRUE)},
    par.settings = list(box.rectangle=list(col=my.col[2], lwd=2), plot.symbol = list(pch='.', cex = 0.1, col=my.col[2]), box.umbrella=list(col=my.col[2])), scales=list(x=list(rot=10,
 labels=c("Not Essential", "Essential"))))

dev.copy2pdf(device=quartz, file=paste(folderAnalysis, organism,"OmegaEssentialityNewAres.pdf", sep=""),onefile=TRUE)#,paper="A4r"

dataPF <- data.frame(dataPlot)
dataPF$Omega <- as.numeric(dataPF$Omega)

t.test(Omega~Essentiality, data=dataPF)

t.test(Omega~Essentiality, data=dataPF, alternative = "greater")
#################
#################
###################


#####################################
###Correlation between correlations###
#####################################

########################
dataCorME <- read.table(paste(folderAnalysis, "Mouse.txt",sep=""), header=TRUE)

dataCorHF <- read.table(paste(folderAnalysis, "Human.txt",sep=""), header=TRUE)

dataCor <- merge(dataCorME, dataCorHF, by=c("x1","x2"), all.x=TRUE, all.y=TRUE)
dataCor <- dataCor[,c("x1", "x2", "corValue.x", "corValue.y")]
colnames(dataCor) <- c("x1", "x2", "ENCODE", "Fagerberg")

head(dataCor)
dataPlot <- dataCor[,c(-2)]
head(dataPlot)
v <- colnames(dataPlot[,-1])

dataPlot$x1 <- ifelse(dataPlot$x1 == "Omega", "Omega", "Others")

dataPlot <- as.matrix(dataPlot)


dev.new(height=12, width=16)
trellis.par.set(list(background=list(col=my.col[1]), add.text=list(col=my.col[2], cex=1.5),axis.line=list(col=my.col[2]), axis.text=list(col=my.col[2], cex=1.5),
 par.main.text=list(col=my.col[2], cex=2.5), par.xlab.text=list(col=my.col[2], cex=1.5), par.ylab.text=list(col=my.col[2], cex=1.7), plot.line=list(col=my.col[2]), dot.line=list(lwd=1,
 lty=2, col="#4B4B4B"))) #trellis.par.get()

xyplot(as.numeric(dataPlot[,3]) ~ as.numeric(dataPlot[,2]), groups=dataPlot[,1], col=c("#FF0000F0", "#0000CCF0"),
panel = function(x, y, ...){
    panel.superpose(x, y, ...,
    panel.groups=function(x, y, col, col.symbol, ...) {
        panel.xyplot(x, y, col=col, ...)
        #panel.abline(lm(y~x), col.line=col.symbol)
        panel.text(0.75,0.9, "y=0.002 + 0.89*x", col="black")
        panel.text(-0.47,-0.023, "GC content to Stage number ->", col="black", cex=1)
        panel.text(0.44,-0.035, "<- GC content to Maximal expression", col="black", cex=1)
    },
    panel.abline(lm(y~x), col.line="black", lwd=2, lty=2)
    )},
xlab="Mouse correlation values", ylab="Human correlation values", main=paste("Partial coefficient correlations in human and mouse",sep=""), pch="*", cex=6, xlim=c(-1,1), ylim=c(-1,1)
 )


dev.copy2pdf(device=quartz, file=paste(folderAnalysis, "MusHumanCorrelationCorrelationsOmega.pdf", sep=""),onefile=TRUE)
############

###################################
```