METHODOLOGY

Tissue Enrichment Analysis: TEA

David Angeles-Albores[1], Raymond Y Lee[2], Juancarlos Chan[2] and Paul W Sternberg[1*]

[*]Correspondence: pws@caltech.edu
[1]California Institute of Technology, Division of Biology and Biological Engineering, 1200 E California Blvd, 91125 Pasadena, US
Full list of author information is available at the end of the article

Abstract

Over the last ten years, there has been an explosive development in tools capable of measuring gene expression. These tools generate a large number of gene targets, but understanding these datasets, and forming hypotheses based on them remains challenging.

We present a method for detecting tissue enrichment in *C. elegans* using the tissue ontology for this organism. We also present an efficient method for trimming the ontology that results in concise yet useful output.

Our tool, Tissue Enrichment Analysis (TEA), can be found at www.wormbase.org/tea

Keywords: Gene Ontology; Tissue Ontology; Wormbase

Background

RNA-seq and other high-throughput methods in biology have the ability to identify thousands of genes that are altered between conditions. These genes are often correlated in their biological characteristics or functions, but identifying these functions remains challenging. In order to interpret these long lists of genes, biologists need to abstract genes into fewer terms that are biologically relevant in order to form hypotheses about what is happening in the data. One such abstraction method relies on Gene Ontology (GO). GO provides a controlled set of hierarchically ordered terms in the form of an directed acyclic graph[1–3] that provide detailed information about the molecular, cellular or biochemical functions of the gene among others. For a given gene list, certain software programs can query whether a particular gene is enriched[4–6]. However, GO is often difficult to interpret due to the large number of terms associated with a given gene. There exist a number of GO analytic tools for use by the community but a shared complaint for many programs is the very large number of GO terms that are significantly associated with any given gene list.

A common tool for GO analysis, DAVID, clusters terms into broad categories that are amenable to exploration by researchers [7], whereas PANTHER, a different software package [4, 8], attempts to solve this issue by employing a manually reduced ontology, GOslim (pers. comm.).

Here we provide a new framework that analyses user-input list for enrichment of specific tissues. We believe that tissues are physiologically relevant units with broad, relatively well-understood functionalities amenable to hypothesis formation. As such, we believe that identification of tissues is likely to provide researchers with enough information to be able to form hypotheses about the physiological responses of an organism to a specified condition. Our analysis also cuts down on result verbosity by filtering the ontology before testing using a small set of well-defined criteria to remove terms that don't contribute extra information. To our knowledge, such filtering has never been performed in an algorithmic fashion for an ontology before — indeed, tools such as DAVID do not employ term trimming *a priori* of testing, but rather fuzzy clustering *post* testing to reduce the number of ontology terms. We believe our trimming methodology strikes a good balance between detailed tissue calling and conservative testing.

We built our software using a pre-established tissue ontology for the worm, C. elegans [9]. The C. elegans database, Wormbase[10], maintains a carefully curated list of gene expression data from GFP-reporters. We use this gold-standard list to develop a tissue enrichment analysis that reliably identifies even small tissues and show that we can reliably discriminate between embryonic and larval tissues. Our tool is available in Wormbase at the address http://mangolassi.caltech.edu/ azurebrd/cgi-bin/testing/amigo/getWithPost.cgi and provides users with a text-based file of the enrichment results as well as a simple and clear graph of the results that exhibit the largest fold-change enrichment. Although we present results here for the worm, we note that our software is species agnostic, and we are working to integrate tissue ontologies from other databases to provide a broader service to the community.

Methods

Generating a Useful Dictionary

Reducing term redundancy through a similarity metric

As a first step to generate our tissue enrichment software, we wished to select tissue terms that were reasonably well-annotated, yet specific enough to provide insight and not redundant with other terms. We also wanted to avoid testing tissues at levels where redundancy becomes problematic. For example, several left and right neurons have at least 25 annotating genes and we may want to include them for enrichment testing. However, many left/right neuronal sisters have almost entirely the same annotations, with at most one or two gene differences between them. We reasoned that when two tissues have almost identical annotations, we cannot have statistical confidence in differentiating between them. As a result, testing these sister tissues provides no additional information compared with testing only the parent node to these sisters. We refer to such sisters as 'redundant'. In order to identify redundancy, we defined a similarity metric

$$s_i = \frac{|g_i|}{|\bigcup_{i=0}^{k} g_i|} \quad (1)$$

Where s_i is the similarity for a tissue i eith k sisters; g_i refers to the set of tissues associated with tissue i and $|g|$ refers to the cardinality of set g. For a given set of sisters, we called them redundant if they exceeded a given similarity threshold. We envisioned two possible criteria and built different dictionaries using each one. Under a threshold criteron 'any' with parameter S between $(0, 1)$, a given set of sisters j was considered redundant if the condition

$$s_{i,j} > S \quad (2)$$

was true for any sister i in set j. Under a threshold criterion 'avg' with parameter S, a given set of sisters j was considered redundant if the condition

$$\mathrm{E}[s_i]_j > S \quad (3)$$

was true for the set of sisters j (see figure 1).

Terminal branch terms and parent terms can be safely removed in an algorithmic fashion

Another problem arises from the fact that the tissue ontology is scarcely populated at this point in time. Many nodes have 0-10 annotations, which we consider too few to accurately test. To solve this issue, we implemented a straightforward trimming algorithm. For a given terminal node, we test whether the node has more than a threshold number of annotations. If it does not, the node is removed. The next node in the branch is tested and removed recursively until a node which satisfies the condition is found. At that point, no more nodes can be removed from that branch. This is guaranteed by the structure of the ontology: Parent nodes inherit all of the annotations of all of their descendants, so the number of annotating terms monotonically increases with increasing term hierarchy (see figure 2). In this way, we ensure that our term dictionary includes only those tissues that are considered sufficiently well annotated for statistical purposes.

Finally, we also wanted to remove as many terms as possible from the dictionary with the goals of reducing covariance between terms, decreasing multiple testing and removing as many non-informative terms as possible. Decreasing covariance between terms is important because we employ a frequentist approach that assumes all terms are independent. Large covariation coefficients between some terms means that if one of these tissues tests significant, the other terms are much more likely to pass significance testing as well. This makes adequate correction for false positive rates considerably more difficult. Moreover, from a data analysis perspective, we reasoned that, for any parent node, if all its daughters were selected for testing, there was no additional benefit to test the parent. In other words, if all the daughter nodes are tested, there is little additional information to be gained by including the parent node. To address this issue we removed parent nodes from the analysis if all their daughter nodes passed the annotation threshold (see figure 3).

Filtering greatly reduces the number of nodes used for analysis

By itself, each of these filters can reduce the number of nodes employed for analysis. Notably, these filters are not all commutative – while trimming and redundancy filtering are commutative, applying the ceiling filter is not commutative with either

the trimming or the redundancy filter. If the ceiling filter is applied before any other filter, only terminal nodes will remain, since all the parents have complete daughter sets. Since terminal nodes are the most poorly annotated, after applying the remaining filters very few nodes will be left behind if any. On the other hand, applying the ceiling operator after trimming and redundancy filtering will result in greater numbers of nodes. We always applied the ceiling at the end. For validation (see below) we made a number of different dictionaries. The original ontology has 1675 terms with more than 5 gene annotations. After filtering, dictionary sizes ranged from 21 to a maximum of 400 terms, which shows the number of terms in a scarcely annotated ontology can be reduced by tenfold by application of a few simple filters.

These filters were used to compile a static dictionary that we employ for all analyses. Because we have integrated our scripts to draw on the WormBase databases, our dictionary will remain up to date as tissue expression data improves. Our completed static trimmed dictionary is available for download at the following ftp URL: XX. The final dictionary includes XX tissues for testing, and has XX annotating genes. All code was implemented in Python.

Tissue enrichment testing via a hypergeometric model

Having built a static dictionary, we generated a Python script that implements a significance testing algorithm based on the hypergeometric model. Briefly, the hypergeometric model assumes the existence of an urn with a pre-determined number of balls inside it. The balls can be painted one of several colors. The hypergeometric model provides an answer to the question: If an individual removes N balls, what is the probability of observing n_i balls of color i, if the balls are selected without replacement? Mathematically, this is expressed as:

$$\mathrm{P}(n_i|N, m_1, \ldots, m_k, M) = \frac{\binom{m_i}{n_i}\binom{M-m_i}{N-n_i}}{\binom{N}{n_i}} \quad (4)$$

Here, n_i is the number of balls of type i drawn, N is the total number of draws, m_i is tissue i and $M = \sum_i m_i$ is the total number of balls in the urn. In our specific case, M_i is equal to the total number of annotations in our dictionary. N is found

by taking the user-input list and removing any genes that are not in our annotation dictionary. The remaining genes are then associated with their annotation profiles — if a tissue is associated with s tissues, it generates s balls of s colors. Our program counts the number of times each tissue appears in the user list, and calculates the probability of having withdrawn as many or more balls for each tissue in the user list. Due to the discrete nature of the hypergeometric distribution, this algorithm can generate artifacts when the list is small. To avoid spurious results, a tissue is never considered significant if there are no annotations for it in the user-provided list.

Once the probability of drawing the labels has been quantified, we apply a standard FDR correction using a Benjamini-Hochberg step-up algorithm[11]. Genes that have a q-value less than a given alpha are considered significant. Our default setting is to set the alpha threshold at 0.1, but users will be able to modify this value either in batch or in our web application. The program returns a text-based table showing the tissues that tested significant, along with their associated q-value, the expected number of hits for a list of that size, the observed number of hits and the enrichment fold change (observed hits / expected hits). Finally, the program can also return a bar chart of the enrichment fold change for the fifteen tissues with the largest enrichment fold change. Our software relies heavily on the Pandas, Numpy, Seaborn and SciPy modules to perform all statistical testing and data handling[12–14].

Our software is implemented in an easy to use GUI within WormBase. Users input a gene-list (see figure 4) using any valid gene name for *C. elegans*. These names are processed into standard WBIDs and the result is displayed in the same window in an easy to read format containing all the relevant information, and a graph of the results is also displayed (see figure 5).

Validation of the algorithm and parameter selection

In order to select an appropriate dictionary and validate our tool, we found a set of 30 gold standards based on microarray and RNA-seq literature which are believed to be enriched in specific tissues[]. Some of these studies went on to use GFP to identify expression patterns and for this reason we generated a clean Since the expression data is curated from GFP expression at this time and does not include RNA-seq data, these gold standards are statistically independent from the dataset. We wanted

to select a dictionary which included enough terms to be specific beyond the largest C. elegans tissues, yet would minimize the number of spurious results and which had a good dynamic range in terms of enrichment fold-change. Selection of a dictionary based only on minimization of spurious results would result in a dictionary with a large number of annotations per tissue, and would therefore include only the major tissues. On the other hand, selecting a dictionary that can detect smaller tissues will bias us towards tissues with lesser annotations. To our knowledge there is no good method for assessing false-positive or false-negative results for annotations.

As a first attempt to select a good dictionary, we generated all the possible combinations of dictionaries with minimal annotations of 10, 25, 50 and 100 genes and similarity cutoffs of 0.9, 0.95 and 1, using 'average' or 'any' thresholding criteria for the latter (see table 1). For these dictionaries, the number of tissues tested ranged from 97 to 676. The number of tissues was inversely correlated to the minimum annotation, as expected, and was largely insensitive to the redundancy threshold, at least in the range we explored (0.9-1). Next, we analyzed all 30 datasets using each dictionary. Because of the large number of results, instead of analyzing each set of terms individually, we pooled all results for a given dictionary into histograms. When we analyzed the distribution of significant q-values for the dictionaries, we found that the similarity threshold mattered relatively little for any dictionary. We also noticed that the 'any' thresholding method resulted in tighter histograms with a mode closer to 0 (data not shown). For this reason, we chose the 'any' method for dictionary generation. The average q-value increased with decreasing annotation cut-off (see figure 6), which reflects the decreasing statistical power associated with fewer annotations per term, but we remained agnostic as to how significant the trade-off between power and term specificity is. Based on these observations, we ruled out the dictionary with the 100 annotation cut-off - it had the fewest terms and its q-values were not low enough to compensate the trade-off in specificity.

To select between dictionaries generated between 50, 33 and 25 annotation cut-offs, and also to ensure the terms that are selected as enriched by our algorithm are reasonable, we looked in detail at the enrichment analysis results. Most results were highly comparable and in line with what was expected. For some sets, all dictionaries seemed to perform well. For example, in our 'all neuron enriched set' **??** the result was an amalgamation of neuron related terms including mechanosensory neurons,

thermosensitive neurons, interneurons, ganglions and male rays regardless of the dictionary used. On the other hand, when we looked at a gene set enriched for germline precursor expression in the embryo **??**, the dictionary with the 50 cutoff was only able to identify 'oocyte WBbt:006797'; whereas the two smaller dictionaries were able to single out cells germline precursor cells – at the 33-cutoff, our tool identified 'Z2' and 'Z3' as being five-fold enriched; whereas at the 25 gene-cutoff the terms 'Psub4','Psub3' and 'Psub2' were identified in addition to 'Z2' and 'Z3'. We queried an embryonic stage intestine precursor associate geneset **??**. Notably, this gene set yielded no enrichment when using the 25 cutoff dictionary, nor when using the 50 cutoff dictionary. However, the 33 cutoff dictionary suggested, probably correctly, that the E lineage was heavily enriched in this set. Not all queries worked equally well. For example, a number of intestinal enriched genes sets **??** were not enriched in intestine in any dictionary, but they were enriched for pharynx- and hypodermis-related terms. We were somewhat surprised that intestinal gene sets performed poorly, since the intestine is a relatively well-annotated tissue. We also assessed the internal agreement of our tool by using independent gene-sets that we expected to be enriched in the same tissues. We had two independent pan-neuronal sets **??**; two independent PVD enriched sets **??**; two independent GABAergic gene sets **??**; two independent pharyngeal gene sets; and two independent intestinal gene sets **??**. Overall, the tool seems to have good internal agreement. On most sets, the same terms were enriched, although order was somewhat variable. However, most high-scoring terms were preserved between gene sets. The intestinal gene-sets and pharyngeal gene sets comparisons were exceptions, since at least one gene set was missing each for intestine and pharynx in every dictionary, so we didn't consider them as informative for assessing internal agreement.

All comparisons can be found online at: www.XXX.com. Overall, the dictionary generated by a 33 gene annotation cutoff with 0.95 redundancy threshold using the 'any' criterion. seemed to perform well, with a good balance between specificity, verbosity and accuracy, so we selected this parameter set to generate our static dictionary.

Results

We applied our tool to the RNA-seq datasets developed by Engelmann et al. [15] in order to attempt to gain further understanding of the biology underlying these datasets. Engelmann et al. exposed young adult worms to 5 different pathogenic bacteria or fungi for 24 hours, after which mRNA was extracted from the worms for sequencing. We obtained the genes that Engelmann et al identified as up- or down- regulated in their assay, and ran TEA using these lists. Initially we noticed that genes that are down-regulated tend to be twice better annotated on average than genes that were up-regulated, suggesting that our understanding of the worm immune system is scarce, in spite of important advances made over the last decade. Strikingly, 4 out of the five samples showed enrichment of neuronal tissues or neuronal precursor tissues (in the case of Harposporium sp) amongst the down-regulated genes. A possible explanation for this might be that the infected worms are sick and the neurons are beginning to shut down; an alternative hypothesis would be that the worm is down-regulating specific neuronal pathways as a behavioural response against the pathogen. Indeed, several studies[16, 17] have provided evidence that *C. elegans* uses chemosensory neurons to identify pathogens Interestingly, one bacterium did not exhibit the same pattern of down-regulation of neuronal-associated genes. *E. faecalis* showed increased expression of genes associated with neuronal tissues, hinting that *E. faecalis* may have a different pathogenic profile. Up-regulated tissues, when detected, included the hypodermis and excretory duct. Our results highlight the involvement of various *C. elegans* neuronal tissues in pathogen defense and/or illness.

Discussion

We have presented a tissue enrichment analysis tool that employs a standard hypergeometric model to test the *C. elegans* tissue ontology. We have also presented the first, to our knowledge, ontology trimming algorithm. This algorithm, which is very easy to execute, places strong limits on the number of terms selected for testing. Due to the nature of all ontologies as hierarchical, acyclical graphs with term inheritance, term annotations are correlated along any given branch. This correlation reduces the benefits of including all terms for statistical analysis - for any given term along a branch, if that term passes significance, there is a high probability

that many other terms along that branch will also pass significant. If the branch is enriched by random chance, error propagation along a branch means that many more false positives will follow. Thus, a researcher might be misled by the number of terms of correlated function and assign importance to this finding; the fact that the branching structure of GO amplifies false positive signals is a powerful argument for either reducing branch length or branch intracorrelation, or both. On the other hand, if a term is actually enriched, we argue that there is little benefit to presenting the user with additional terms along that branch. Instead, a user will benefit most from testing sparsely along the tree at a suitable specificity for hypothesis formation. Related terms of the same level should only be tested when there is sufficient annotation to differentiate, with statistical confidence, whether one term is enriched above the other (see SI for a back-of-the-envelope calculation of when this can be the case). Our algorithm reduces branch length by identifying and removing nodes that are insufficiently annotated and parents that are likely to include sparse information.

It is important to note that our tool is not the first tissue enrichment model for the worm that has been reported. Chikina *et al* [18] report a tissue enrichment model based on an SVM classifier that has been trained on microarray studies. SVM classifiers are powerful tools capable of great sensitivity, but they require continuous retraining as tissue expression data widens. Our tool benefits from the fact that it will be integrated in WormBase and will therefore be updated continuously as new data is integrated.

We have tried hard to benchmark our tool well. However, our analysis suffers from the drawback that is very hard to benchmark negative controls. Even for our set of positive controls, the statistical analysis sometimes throws out unexpected results. For example, the embryonic germline precursor gene set had the term 'AB' as the most enriched term in the dictionaries with cut off of 25 and 33. Is this an error, or does this hint at new biology? Although we were unable to determine false-positive and false-negative rates, we don't believe this should deter scientists from using our tool. Rather, we encourage researchers to use our tool carefully as a guide, integrating evidence from multiple sources to inform the most likely hypotheses. As with any other tool based on statistical sampling, our analysis is most vulnerable to bias in the data collection stage. For example, we know that

tissue expression reports are negatively biased against germline expression due to the difficulty associated with extrachromosomal array expression in that tissue. Support from the community will be crucial in correcting these flaws going forward; indeed, without the community reports of tissue expression this tool would not be possible.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

DA and PWS conceived of the project; DA developed algorithm; RYL made intellectual contributions to the project; RYL and JC developed the web GUI.

Acknowledgements

We would like to acknowledge all members of the Sternberg lab for helpful discussion.

Author details

[1]California Institute of Technology, Division of Biology and Biological Engineering, 1200 E California Blvd, 91125 Pasadena, US. [2]California Institute of Technology, 1200 E California Blvd, 91125 Pasadena, US.

References

1. The Gene Ontology Consortium: Gene Ontology: tool for the unification of biology. Nature Genetics **25**(may), 25–29 (2000). doi:10.1038/75556. 10614036
2. Ontology, G.: Gene Ontology. Nature Reviews Genetics **2009**, 1–13 (2009)
3. The Gene Ontology Consortium: Gene Ontology Consortium: going forward. Nucleic Acids Research **43**(D1), 1049–1056 (2015). doi:10.1093/nar/gku1179
4. Mi, H., Dong, Q., Muruganujan, A., Gaudet, P., Lewis, S., Thomas, P.D.: PANTHER version 7: Improved phylogenetic trees, orthologs and collaboration with the Gene Ontology Consortium. Nucleic Acids Research **38**(SUPPL.1) (2009). doi:10.1093/nar/gkp1019
5. McLean, C.Y., Bristor, D., Hiller, M., Clarke, S.L., Schaar, B.T., Lowe, C.B., Wenger, A.M., Bejerano, G.: GREAT improves functional interpretation of cis-regulatory regions. Nature biotechnology **28**(5), 495–501 (2010). doi:10.1038/nbt.1630
6. Huang, D.W., Lempicki, R.a., Sherman, B.T.: Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nature Protocols **4**(1), 44–57 (2009). doi:10.1038/nprot.2008.211
7. Huang, D.W., Sherman, B.T., Tan, Q., Kir, J., Liu, D., Bryant, D., Guo, Y., Stephens, R., Baseler, M.W., Lane, H.C., Lempicki, R.A.: DAVID Bioinformatics Resources: Expanded annotation database and novel algorithms to better extract biology from large gene lists. Nucleic Acids Research **35**(SUPPL.2) (2007). doi:10.1093/nar/gkm415
8. Mi, H., Muruganujan, A., Thomas, P.D.: PANTHER in 2013: Modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. Nucleic Acids Research **41**(D1) (2013). doi:10.1093/nar/gks1118
9. Lee, R.Y.N., Sternberg, P.W.: Building a cell and anatomy ontology of Caenorhabditis elegans (2003). doi:10.1002/cfg.248
10. Harris, T.W., Baran, J., Bieri, T., Cabunoc, A., Chan, J., Chen, W.J., Davis, P., Done, J., Grove, C., Howe, K., Kishore, R., Lee, R., Li, Y., Muller, H.M., Nakamura, C., Ozersky, P., Paulini, M., Raciti, D., Schindelman, G., Tuli, M.A., Auken, K.V., Wang, D., Wang, X., Williams, G., Wong, J.D., Yook, K., Schedl, T., Hodgkin, J., Berriman, M., Kersey, P., Spieth, J., Stein, L., Sternberg, P.W.: WormBase 2014: New views of curated biology. Nucleic Acids Research **42**(D1) (2014). doi:10.1093/nar/gkt1063
11. Benjamini, Y., Hochberg, Y.: Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing (1995). 95/57289. doi:10.2307/2346101. http://www.jstor.org/stable/2346101

12. McKinney, W.: pandas: a Foundational Python Library for Data Analysis and Statistics. Python for High Performance and Scientific Computing, 1–9 (2011)
13. Van Der Walt, S., Colbert, S.C., Varoquaux, G.: The NumPy array: A structure for efficient numerical computation. Computing in Science and Engineering **13**(2), 22–30 (2011). doi:10.1109/MCSE.2011.37. 1102.1523
14. Oliphant, T.E.: SciPy: Open source scientific tools for Python. Computing in Science and Engineering **9**, 10–20 (2007)
15. Engelmann, I., Pujol, N.: Innate Immunity in C . Elegans. Inverrtebrate Immunity, 105–121 (2010)
16. Meisel, J.D., Kim, D.H.: Behavioral avoidance of pathogenic bacteria by Caenorhabditis elegans. Trends in Immunology **35**(10), 465–470 (2014). doi:10.1016/j.it.2014.08.008
17. Zhang, Y., Lu, H., Bargmann, C.I.: Pathogenic bacteria induce aversive olfactory learning in Caenorhabditis elegans. Nature **438**(7065), 179–184 (2005). doi:10.1038/nature04216
18. Chikina, M.D., Huttenhower, C., Murphy, C.T., Troyanskaya, O.G.: Global prediction of tissue-specific gene expression and context-dependent gene networks in Caenorhabditis elegans. PLoS Computational Biology **5**(6) (2009). doi:10.1371/journal.pcbi.1000417

Figures

Figure 1 Schematic diagram of annotations for two sisters. The parent node (green) contains at least as many annotations as the union of the two sisters. These two sisters share annotations extensively. Therefore they are too similar and should be removed.

Figure 2 Schematic showing terminal node removal. Nodes with less than a threshold number of genes are trimmed (light red) and discarded from the dictionary. Here, the threshold is 25 genes.

Figure 3 Schematic showing root node removal. We trim parent nodes (light red) if all their daughter nodes have more than the threshold number of annotations. Here, the threshold is 25 genes.

Figure 4 Screenshot of the web GUI.

Figure 5 Screenshot of results from web GUI.

Figure 6 Kernel density estimates for 30 gold standard datasets. We ran TEA on 30 datasets we believed to be enriched in particulae tissues and pooled all the results to observe the distribution of q-values. The mode of the distribution for dictionaries with annotation cut-offs of 100 and 50 genes are very similar; however, when the cut-off is lowered to 25 genes, the mode of the distribution shifts to the left, potentially signalling a decrease in measurement power.

Figure 7 Comparison of Enrichment Results for dictionary size 50 (left) and 25 (right) for a PVD-OLL enriched gene set. Left, at 50 annotation cut-off, TEA singles PVD as highly enriched. Other mechanosensory neurons are also enriched . Right, when the dictionary cut-off is set to 25, TEA shows embryonic tissues that are unrelated to the PVD and OLL lineages.

Figure 8 Genes altered in *C. elegans* after 24hr exposure to *D. coniospora* (fungus) Figure legend text.

Figure 9 Genes altered in *C. elegans* after 24hr exposure to *Harposporium sp.* (fungus) Figure legend text.

Figure 10 Genes altered in *C. elegans* after 24hr exposure to *Serratia marcescens* (bacteria) Figure legend text.

Figure 11 Genes altered in *C. elegans* after 24hr exposure to *E. faecalis* (bacteria) Figure legend text.

Table 1 Parameter specifications and number of tissues for all dictionaries.

Tables

Additional Files

Additional file 1 — Supplementary Information
Complete results from benchmarking analysis

Additional file 2 — Supplementary Information
Complete results from re-analysis of Engelmann et al

Additional file 3 — IPython Notebook
Tutorial for users interested in batch script generation using our software.