DeepIsoFun: A deep domain adaptation approach to predict isoform functions (Supplementary Materials)

Dipan Lal Shaw[1], Hao Chen[1], and Tao Jiang[1,2]

[1] Department of Computer Science and Engineering, University of California, Riverside, CA
[2] Bioinformatics Division, BNRIST / Department of Computer Science and Technology, Tsinghua University, Beijing, China

1 Some supplementary figures

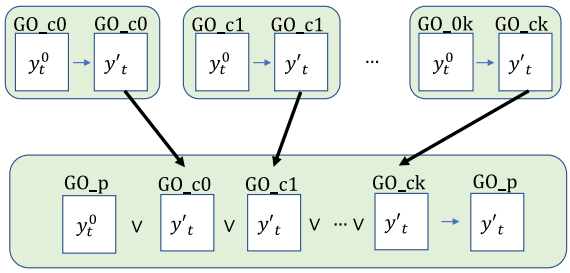


Fig. S1: A schematic illustration of how the training for child nodes may help the training for parent nodes. The GO terms are sorted in topological order and the NN model is trained for each GO term separately in the reverse order. At a parent node p, the predicated class labels of each isoform for its children are simply added to the initial isoform class labels during its training process.

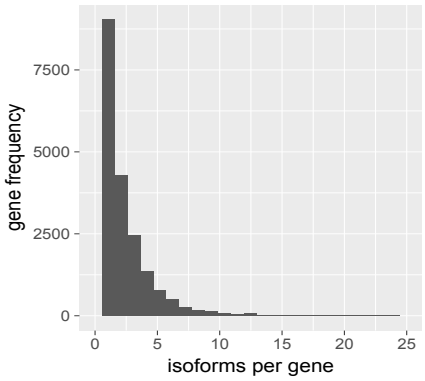


Fig. S2: The distribution of isoforms over genes. Out of the 19532 genes in RefSeq, 9039 have only one isoform (called single isoform genes or SIGs) and 10313 have more than one isoform (called multiple isoform genes or MIGs).

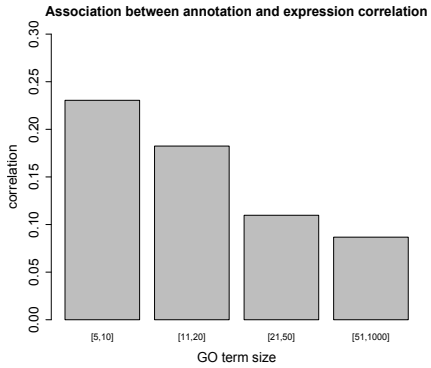


Fig. S3: Correlation between expression similarity and functional similarity with respect to different GO term sizes. The GO terms were again divided into four groups based on sizes. For each group, all genes that have been annotated with at least one term in the group were collected. For these genes, we generated two matrices to represent their (pairwise) similarity in terms of expression profiles or GO functions. Then, using a standard tool (cor.test) in the R Stats package, we estimated the Pearson correlation between these two similarity matrices. Clearly, the stronger the correlation, the more likely we are able to predict GO functions based on expression. As shown in the histogram, the correlation decreases as the GO term size increases although the correlation is weak in all groups. This trend might indicate that there is more annotation noise in GO terms with larger sizes, which in turn might help explain why the performance of DeepIsoDun decreases with respect to GO terms with larger sizes.

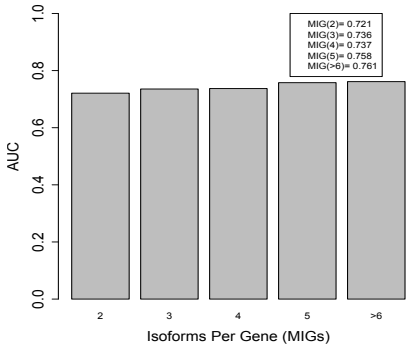
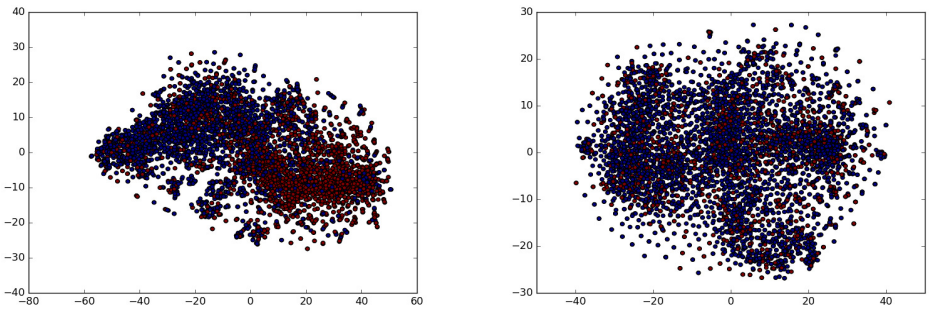


Fig. S4: Performance of DeepIsoFun on MIGs with different numbers of isoforms. We divided MIGs into five groups: MIGs with 2 isoforms, 3 isoforms, 4 isoforms, 5 isoforms, or more than 5 isoforms. The average AUC performace of DeepIsoFun is shown in the histogram. It increases slightly as the number of isoforms per gene increases.



(a) Without DA (b) With DA

Fig. S5: Effectiveness of DA in mixing the two domains. Red dots represent samples from the source domain after feature extraction and blue dots represent samples from the target domain. The tool t-SNE (1) was used to perform dimensionality reduction and visualization. The DA technique clearly makes the two distributions harder to distinguish.

2 Comparison of the divergence of isoform functions predicted by the methods

It would be interesting to compare the average dissimilarity among the isoform functions of the same gene predicted by different methods, although it would probably be difficult to rank the methods based on this measure. We applied the same divergence analysis procedure for DeepIsoFun to the three methods in (2; 3; 4; 5), mi-SVM, MILP (*i.e.*, the non-iterative version of iMILP) and WLRM. The result is shown in Figure S6. The average dissimilarity scores achieved by DeepIsoFun, mi-SVM, MILP, and WLRM are respectively 0.162, 0.322, 0.197 and 0.204. Interestingly, the first three methods all reported the highest divergence on the branch CC.

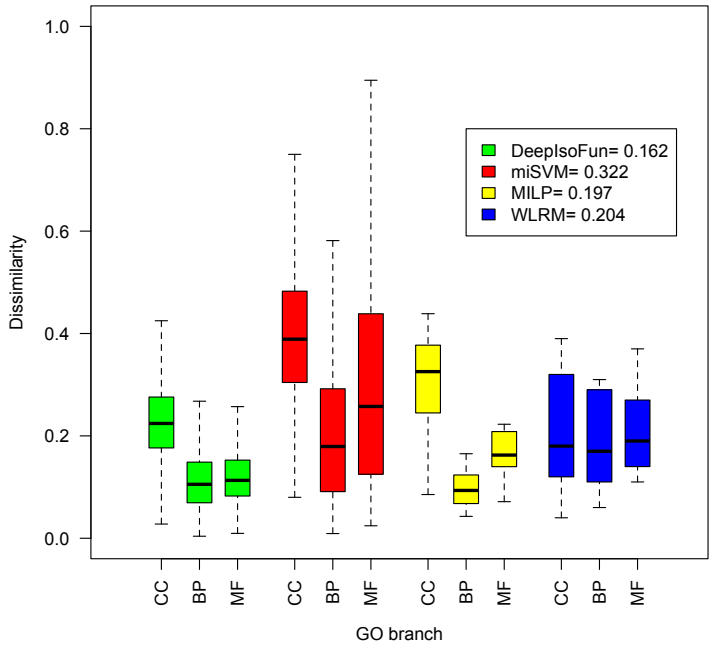


Fig. S6: Functional dissimilarity distributions on the three main branches of GO achieved by DeepIsoFun, mi-SVM, MILP, and WLRM.

3 Time efficiency of the methods

When the programs are run sequentially on a single CPU, DeepIsoFun is faster than the other three methods. More precisely, it takes DeepIsoFun 12.09 minutes to process one GO term on Dataset1 on the average (on a standard compute server node), which is 1.42 times faster than mi-SVM, 0.37 times faster than MILP, 2.49 times faster than iMILP, and 0.21 times faster than WLRM.

Table S1: Average computation time (in minute) per GO term.

	DeepIsoFun	mi-SVM	MILP	iMILP	WLRM
Time	12.09	29.26	16.48	42.17	14.59

4 Some more supplementary tables

Table S2: To check if DeepIsoFun consistently outperforms the other methods on data from more organisms, we tested them on two more expression datasets concerning *Arabidopsis thaliana* and *Drosophila melanogaster* (*i.e.*, fruit fly), respectively named as Dataset#4 and Dataset#5. The data generation procedure is similar as described in Section 3.2. Dataset#4 contains the expression profiles of 24315 genes and 31811 isoforms derived from 13 SRA arabidopsis studies consisting of 101 experiments and Dataset#5 contains the expression profiles of 13022 genes and 28419 isoforms derived from from 11 SRA fruit fly studies consisting of 128 experiments, with the requirement that each study contains at least 6 experiments. The transcript annotations for these two organism were collected from TAIR (https://www.arabidopsis.org/) and FlyBase (http://flybase.org/). The results in the table show that DeepIsoFun consistently performs better than MILP and iMILP. Specifically, in terms of AUC, DeepIsoFun is 97.2% and 58.1% better than MILP and iMILP on Dataset#4 (45.6% and 29.4% better on Dataset#5), respectively, against the baseline 0.5. In terms of AUPRC, DeepIsoFun performs 38.7% and 15.2% better than MILP and iMILP on Dataset#4 (33.6% and 16.1% better on Dataset#5), respectively, against the baseline 0.1.

	AUC			AUPRC		
Method / Dataset	DeepIsoFun	MILP	iMILP	DeepIsoFun	MILP	iMILP
Dataset#4	0.674	0.588	0.610	0.229	0.193	0.212
Dataset#5	0.698	0.636	0.653	0.259	0.219	0.237

Table S3: Comparison of DeepIsoFun, mi-SVM and WLRM on Dataset#4 and Dataset#5. Again, DeepIsoFun consistently outperforms the other two methods. In terms of AUC, DeepIsoFun is 30.6% and 22.7% better than mi-SVM and WLRM, respectively, on Dataset#4 (12.2% and 15.7% better on Dataset#5). In terms of AUPRC, DeepIsoFun is 31.1% and 40.6% better than mi-SVM and WLRM, respectively, on Dataset#4 (25.9% and 18.1% better on Dataset#5).

	AUC			AUPRC		
Method / Dataset	DeepIsoFun	mi-SVM	WLRM	DeepIsoFun	mi-SVM	WLRM
Dataset#4	0.662	0.624	0.632	0.197	0.174	0.169
Dataset#5	0.684	0.664	0.659	0.231	0.204	0.211

Table S4: Comparison of DeepIsoFun, MILP and iMILP in AUC performance with respect to GO terms of different sizes. Again, we divided GO terms into four groups based on size: [5,10], [11,20], [21, 50], and [51, 1000]. As shown in the table, the performance of all methods deceases as the GO term size increases in most cases. This is consistent with the pattern observed in Section 3.3.2 and Figure 3(a).

Method / Dataset	DeepIsoFun				MILP				iMILP			
	[5,10]	[11,20]	[21,50]	[51,1000]	[5,10]	[11,20]	[21,50]	[51,1000]	[5,10]	[11,20]	[21,50]	[51,1000]
Dataset#1	0.761	0.744	0.734	0.731	0.631	0.629	0.612	0.608	0.662	0.657	0.64	0.631
Dataset#2	0.754	0.742	0.726	0.717	0.589	0.579	0.566	0.561	0.661	0.631	0.626	0.62
Dataset#3	0.741	0.726	0.721	0.692	0.558	0.536	0.532	0.534	0.701	0.679	0.665	0.651
Dataset#4	0.701	0.682	0.664	0.647	0.609	0.584	0.582	0.577	0.623	0.608	0.611	0.598
Dataset#5	0.714	0.705	0.695	0.681	0.637	0.641	0.63	0.638	0.668	0.662	0.642	0.638

Table S5: Comparison of DeepIsoFun, mi-SVM and WLRM on Dataset#4 and Dataset#5 in AUC performance with respect to different GO term sizes. As shown in the table, the performance of all methods deceases as the GO term size increases in most cases. Again, this is consistent with the pattern observed in Section 3.3.2 and Figure 3(a).

Method / Dataset	DeepIsoFun				mi-SVM				WLRM			
	[5,10]	[11,20]	[21,50]	[51,1000]	[5,10]	[11,20]	[21,50]	[51,1000]	[5,10]	[11,20]	[21,50]	[51,1000]
Dataset#1	0.756	0.739	0.736	0.708	0.705	0.683	0.664	0.662	0.714	0.696	0.694	0.661
Dataset#2	0.732	0.728	0.725	0.729	0.674	0.665	0.671	0.659	0.672	0.667	0.649	0.641
Dataset#3	0.732	0.715	0.707	0.667	0.657	0.656	0.642	0.616	0.681	0.671	0.67	0.637
Dataset#4	0.687	0.661	0.657	0.643	0.635	0.622	0.623	0.617	0.643	0.636	0.628	0.619
Dataset#5	0.699	0.683	0.682	0.671	0.676	0.669	0.661	0.649	0.666	0.658	0.655	0.656

Table S6: Performance of DeepIsoFun in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. The 18 genes that have multiple isoforms and are annotated with both pro-apoptosis and anti-apoptosis functions are listed in the first two columns. Here, the ID of a gene is extracted from the NCBI database. The numbers of isoforms of the genes are shown in the third column. DeepIsoFun was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark. The prediction results concerning the three functions are shown in the next three columns, where an "Y" means that the concerned function is predicted for at least one of the isoforms of the concerned gene. The last column shows for each gene, if some isoforms of the gene are predicted to be pro-apoptosis but not anti-apoptosis while some other isoforms of the gene are predicted to be anti-apoptosis but not pro-apoptosis.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	Y	Y	Y	Y
BARD1	580	5	Y	Y	Y	N
BMP4	652	9	Y	Y	Y	Y
DDX3X	1654	3	Y	N	N	N
DNAJA1	3301	2	Y	Y	N	N
IL6	3569	2	Y	Y	Y	N
MAPK8	5599	17	Y	Y	Y	Y
PSEN2	5664	2	N	N	N	N
RPS27A	6233	3	Y	N	Y	Y
SNCA	6622	4	Y	Y	N	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	Y	Y	Y	N
UBA52	7311	8	Y	Y	Y	N
UBB	7314	6	Y	Y	Y	Y
HMGA2	8091	5	Y	N	Y	N
SQSYM1	8878	3	Y	N	Y	N
ZNF268	10795	9	Y	Y	Y	Y
YNFAIP8	25816	6	Y	Y	Y	Y

Table S7: Performance of iMILP in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. Here, iMILP was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	N	N	N	N
BARD1	580	5	Y	Y	N	N
BMP4	652	9	Y	Y	Y	N
DDX3X	1654	3	Y	N	Y	N
DNAJA1	3301	2	N	N	N	N
IL6	3569	2	Y	N	Y	N
MAPK8	5599	17	Y	Y	Y	N
PSEN2	5664	2	Y	Y	N	N
RPS27A	6233	3	Y	Y	Y	Y
SNCA	6622	4	Y	Y	N	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	N	N	N	N
UBA52	7311	8	Y	N	Y	N
UBB	7314	6	Y	Y	Y	Y
HMGA2	8091	5	Y	N	Y	N
SQSYM1	8878	3	N	N	N	N
ZNF268	10795	9	Y	Y	Y	Y
YNFAIP8	25816	6	Y	Y	Y	Y

Table S8: Performance of mi-SVM in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. Here, mi-SVM was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	Y	N	Y	N
BARD1	580	5	Y	Y	Y	Y
BMP4	652	9	Y	Y	Y	N
DDX3X	1654	3	Y	N	Y	N
DNAJA1	3301	2	N	N	N	N
IL6	3569	2	Y	Y	N	N
MAPK8	5599	17	Y	Y	Y	Y
PSEN2	5664	2	N	N	N	N
RPS27A	6233	3	Y	Y	Y	Y
SNCA	6622	4	Y	Y	N	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	N	N	N	N
UBA52	7311	8	Y	Y	Y	N
UBB	7314	6	Y	Y	Y	N
HMGA2	8091	5	Y	Y	N	N
SQSYM1	8878	3	Y	N	Y	N
ZNF268	10795	9	Y	Y	Y	N
YNFAIP8	25816	6	Y	Y	N	N

Table S9: Performance of WLRM in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. Here, WLRM was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	Y	Y	Y	N
BARD1	580	5	Y	Y	Y	N
BMP4	652	9	Y	Y	Y	Y
DDX3X	1654	3	N	N	N	N
DNAJA1	3301	2	Y	Y	N	N
IL6	3569	2	Y	N	Y	N
MAPK8	5599	17	Y	Y	Y	N
PSEN2	5664	2	N	N	N	N
RPS27A	6233	3	N	N	N	N
SNCA	6622	4	Y	Y	Y	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	N	N	N	N
UBA52	7311	8	Y	Y	Y	N
UBB	7314	6	Y	N	Y	Y
HMGA2	8091	5	Y	Y	Y	N
SQSYM1	8878	3	Y	Y	Y	N
ZNF268	10795	9	Y	N	Y	N
YNFAIP8	25816	6	Y	Y	Y	N

Bibliography

[1] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[2] R. Eksi, H.-D. Li, R. Menon, Y. Wen, G. S. Omenn, M. Kretzler, and Y. Guan, "Systematically differentiating functions for alternatively spliced isoforms through integrating rna-seq data," *PLoS computational biology*, vol. 9, no. 11, p. e1003314, 2013.

[3] W. Li, S. Kang, C.-C. Liu, S. Zhang, Y. Shi, Y. Liu, and X. J. Zhou, "High-resolution functional annotation of human transcriptome: predicting isoform functions by a novel multiple instance-based label propagation method," *Nucleic acids research*, vol. 42, no. 6, pp. e39–e39, 2013.

[4] T. Luo, W. Zhang, S. Qiu, Y. Yang, D. Yi, G. Wang, J. Ye, and J. Wang, "Functional annotation of human protein coding isoforms via non-convex multi-instance learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 345–354, ACM, 2017.

[5] B. Panwar, R. Menon, R. Eksi, H.-D. Li, G. S. Omenn, and Y. Guan, "Genome-wide functional annotation of human protein-coding splice variants using multiple instance learning," *Journal of proteome research*, vol. 15, no. 6, pp. 1747–1753, 2016.