Disjunctive Interconnection Graph Integration for Disease Gene Prioritization

Dinh Tran Van
University of Padova
Padova, Italy
dinh@math.unipd.it

Alessandro Sperduti
University of Padova
Padova, Italy
sperduti@math.unipd.it

Fabrizio Costa
University of Exeter
Exeter, United Kingdom
f.costa@exeter.ac.uk

ABSTRACT

Disease-Gene associations are normally inferred through gene relations which are best represented in form of graphs. Many research are conducted to find out gene relation network by focusing on different aspect of gene expressions. As a consequence, there exist different gene networks. This brings scientists a great opportunity to have more unified views of genes. Graph integration, therefore, is a logical problem which needs to be effectively solved if we desire to build high performance of disease-gene association inferred systems. In this paper, we propose a novel method for disease gene prioritization. The method owns its strength from an efficient graph integration paradigm which allows to simultaneously capture topological features from all graphs to define gene similarities. Results from two experiments conducted on real datasets prove that our method is the state of the art for disease gene prioritization.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Graph integration, disease gene prioritization, graph node kernels, graph decomposition

ACM Reference Format:
Dinh Tran Van, Alessandro Sperduti, and Fabrizio Costa. 2017. Disjunctive Interconnection Graph Integration for Disease Gene Prioritization. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 5 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

A powerful approach to process large heterogeneous sources of data is to use graph encodings [1], [8] and then use graph-based learning systems. In these systems the notion of node similarity is key. A common approach is to resort to graph node kernels such as diffusion-based kernels [2] where the graph node kernel measures the proximity between any pair of nodes by taking into account the paths that connect them. However, when the graph structure is affected by noise in the form of missing links, node similarities are distorted proportionally to the sparsity of the graph and to the fraction of missing links. Two of the main reasons for this are that 1) the lower the average node degree is, the smaller the number of paths through which information can travel, and 2) missing links can end up separating a graph into multiple disconnected components. In this case, since information cannot travel across disconnected components, the similarity of nodes belonging to different components is null. To address these problems we propose to solve a link prediction task prior to the node similarity computation and start studying the question: how can we improve node similarity using link prediction? In this work we review both the link prediction literature and the diffusion kernel literature, select a subset of approaches in both categories that seem well suited, focus on a set of node predicting problems in the bioinformatics domain and empirically investigate the effectiveness of the combination of these approaches on the given predictive tasks.

2 RELATED WORK

3 METHODS

3.1 Definitions and notation

We represent a problem instance as a graph $G = (V, E)$ where V is the set of nodes and E is the set of links. The set E is partitioned into the subset of observed links (O) and the subset of unobserved links (U). Like other approaches we assume that all unobserved links are indeed "non-links" and we therefore define the link prediction problem as the task of ranking candidate links from the most to the least probable to recover links in O but not in U exploiting only the network topology.

We define the *distance* $\mathcal{D}(u, v)$ between two nodes u and v, as the number of edges on the shortest path between them. The *neighborhood* of a node u with radius r, $N_r(u) = \{v \mid \mathcal{D}(u, v) \leq r\}$, is the set of nodes at distance no greater than r from u. The corresponding *neighborhood subgraph* $\mathcal{N}_r^u$ is the subgraph induced by the neighborhood (i.e. considering all the edges with endpoints in $N_r(u)$). The *degree* of a node u, $deg(u) = |\mathcal{N}_1^u|$, is the cardinality of its *neighborhood.* The maximum node degree in the graph G is $deg(G)$.

3.2 Disjunctive interconnection graph integration

Definitions. A graph $G = (V, E)$ is a structure that consists of a node set $V(G)$ and an edge set $E(G)$. The *distance* between two nodes u and v, notated as $\mathcal{D}(u, v)$, is the length of the shortest path between them. The *neighborhood* with radius r of a node v is the set of nodes at a distance no greater than r from v and denoted by $N_r(v)$. The *neighborhood subgraph* with radius r of node v, denoted by $\mathcal{N}_r^v$, is

Conference'17, July 2017, Washington, DC, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

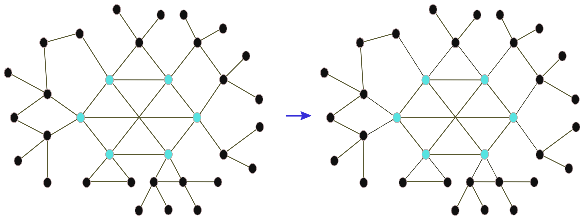

Figure 1: K-core decomposition

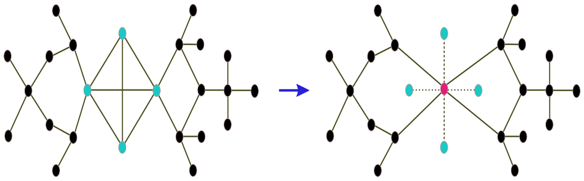

Figure 2: Clique decomposition

the subgraph formed by the nodes in the neighborhood with radius r of v and the relative edges with endpoints in $N_r(v)$.

Network Decomposition. In gene-disease associations networks it is not uncommon to find nodes with high degrees. Unfortunately these cases cannot be effectively processed by decomposition kernels based on exact neighborhood matches because of the high number of neighborhood subgraphs. As an alternative, we propose to decompose the network in a linked collection of sparse sub-networks where each node has a reduced connectivity. More precisely we distinguish two types of edges: *conjunctive* and *disjunctive* edges. Nodes linked by conjunctive edges are going to be used jointly to define the notion of context. Nodes linked by disjunctive edges are instead used to define features based only on the pairwise co-occurrence of the genes at the endpoints. The aim of the following decompositions is to link sparse sub-networks (which comprise only conjunctive edges) via disjunctive edges.

Iterative k-core decomposition [?]: The node set is partitioned in two groups on the basis of the degree of each node w.r.t. a threshold degree D. The node partition is used to induce the conjunctive vs disjunctive edge partition: edges that have endpoints in the same part are marked as conjunctive, while edges with endpoints in different parts are marked as disjunctive. We apply the k-core decomposition iteratively considering only the graph induced by the conjunctive edges until no node has a degree [1]greater than D.

Clique decomposition [?]: To model the notion that nodes in a clique are tightly related, we summarize the whole clique with a new 'representative' node. All the cliques (completely connected subgraphs) with a number of nodes greater than a threshold size C are identified. The endpoints of all edges incident on the clique's nodes are transferred to the representative node. Disjunctive edges are introduced to connect each node in the clique to the representative. Finally all edges with both endpoints in the clique are removed.

In our work a network is transformed by applying first the iterative k-core decomposition and then the clique decomposition.

[1] The degree is defined by only considering incident conjunctive edges.

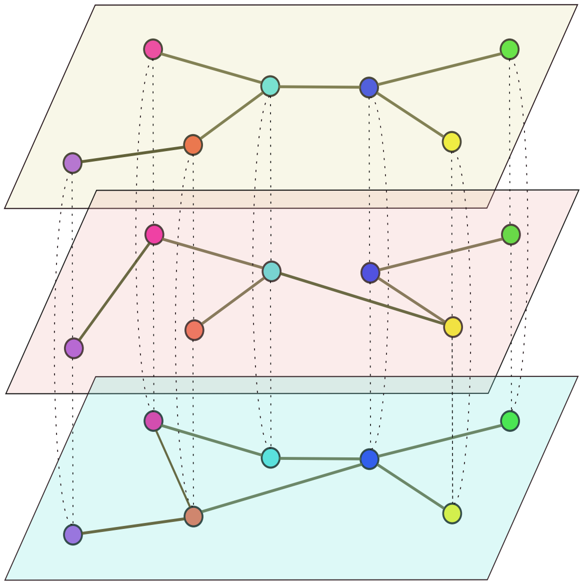

Figure 3: Graph Disjunctive Interconnection Illustration

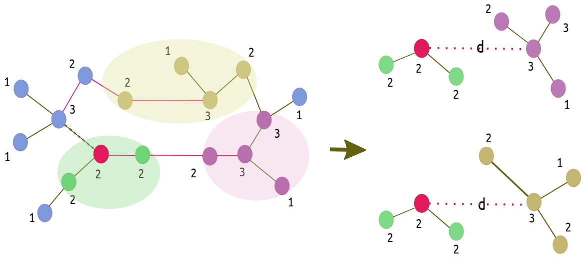

Figure 4: CDNK

Disjuctive interconnection graph integration. Given a set of graphs $\mathcal{G} = \{g_1, g, \ldots, g_n\}$, we call each graph as a layer. Consider a layer tuple g_i, g_j ($g_i, g_j \in \mathcal{G}$) and two nodes $u \in g_i$, $v \in g_j$ such that u and v are both represent for a same gene, we connect u and v by a disjuctive link. As the consequence, nodes represeting same genes in all graph layers are linked by disjuctive links. Figure **??** is a visualization for our graph integration idea.

3.2.1 The Conjunctive Disjunctive Node Kernel. We extend NSPDK and define a node kernel $K(G_u, G_{u'})$ between two copies of the same network G where we distinguish the nodes u and u' respectively. The idea is to define the features of a node u as the subset of NSPDK features that always have the node u as one of the roots. In addition we distinguish between two types of edges, called *conjunctive* and *disjunctive* edges. When computing distances to induce neighborhood subgraphs, only conjunctive edges are considered. When choosing the pair of neighborhoods to form a single feature, we additionally consider roots u and v that are not at distance d but such that u is connected to w via a disjunctive edge and such that w is at distance d from v. In this way disjunctive edges can still allow an *information flow* even if their endpoints are only considered in a pairwise fashion and not jointly.

Formally, we define two relations: the *conjunctive relation* $R^{\wedge}_{r,d}(A_u, B_v, G_u)$ identical to the NSPDK relation $R_{r,d}(A_u, B_v, G)$, and (ii) $\mathcal{D}(u, v) = d$; the *disjunctive relation* $R^{\vee}_{r,d}(A_u, B_v, G_u)$ is true *iff* (i) $A_u \cong \mathcal{N}^u_r$ and $B_v \cong \mathcal{N}^u_r$ are true, (ii) $\exists w$ s.t. $\mathcal{D}(w, v) = d$, and (iii) (u, w) is a disjunctive edge. We define $\kappa_{r,d}$ on the inverse relations ${R^{\wedge}_{r,d}}^{-1}$ and ${R^{\vee}_{r,d}}^{-1}$

$$\kappa_{r,d}(G_u, G_{u'}) =$$
$$\sum_{\substack{A_u, B_v \in {R^{\wedge}_{r,d}}^{-1}(G_u) \\ A'_{u'}, B'_{v'} \in {R^{\wedge}_{r,d}}^{-1}(G_{u'})}} \mathbf{1}_{A_u \cong A'_{u'}} \cdot \mathbf{1}_{B_v \cong B'_{v'}} + \sum_{\substack{A_u, B_v \in {R^{\vee}_{r,d}}^{-1}(G_u) \\ A'_{u'}, B'_{v'} \in {R^{\vee}_{r,d}}^{-1}(G_{u'})}} \mathbf{1}_{A_u \cong A'_{u'}} \cdot \mathbf{1}_{B_v \cong B'_{v'}}.$$

The CDNK is finally defined as $K(G_u, G_v) = \sum_r \sum_d \kappa_{r,d}(G_u, G_v)$, where once again for efficiency reasons, the values of r and d are upper bounded to a given maximal r^* and d^*.

3.3 Conjuctive disjuctive graph node kernel

Here we briefly describe an efficient graph kernel called the Neighborhood Subgraph Pairs Distance kernel (NSPDK) introduced in [?]. NSPDK is an instance of "decompositional" kernels [?] based on the idea of counting the number of common small subgraphs between two graphs. The subgraphs are pairs of neighborhoods whose roots are at a short distance.

Given a labeled graph $G \in \mathcal{G}$ and two rooted graphs A_u, B_v, we first define the relation $R_{r,d}(A_u, B_v, G)$ to be true *iff* $A_u \cong \mathcal{N}^u_r$ is (up to isomorphism $\cong$) a neighborhood subgraph with radius r of G and so is $B_v \cong \mathcal{N}^v_r$, such that v is at distance d from u: $\mathcal{D}(u, v) = d$. We then define the inverse relation R^{-1} that returns all pairs of neighborhoods of radius r at distance d in G, $R^{-1}_{r,d}(G) = \{A_u, B_v | R_{r,d}(A_u, B_v, G) = true\}$. The kernel $\kappa_{r,d}$ over $\mathcal{G} \times \mathcal{G}$ is the number of such fragments in common in two input graphs:

$$\kappa_{r,d}(G, G') = \sum_{\substack{A_u, B_v \in R^{-1}_{r,d}(G) \\ A'_{u'}, B'_{v'} \in R^{-1}_{r,d}(G')}} \mathbf{1}_{A_u \cong A'_{u'}} \cdot \mathbf{1}_{B_v \cong B'_{v'}},$$

where $\mathbf{1}_{A \cong B}$ is the *exact matching function* that returns 1 if A is isomorphic to B and 0 otherwise. Finally, the NSPDK is defined as $K(G, G') = \sum_r \sum_d \kappa_{r,d}(G, G')$, where for efficiency reasons, the values of r and d are upper bounded to a given maximal r^* and d^*, respectively.

4 EXPERIMENTS

In order to evaluate the performance of our proposed method, we conduct two separate experiments.

The first experiment is carried out following the experimental setting in MRF. The aim of this experiment is to compare the performance of the proposed method with other existing methods for disease gene prioritization. We employ three datasets: BioGPS, Pathways and HPRD. To perform the experiments, we employed known gene-disease associations from OMIM, grouped into 20 classes on the basis of disease relatedness by Goh *et al* [?]. Among those classes we selected the 12 with at least 30 confirmed genes. We then built a training set consisting of a positive set P and a reliably negative set N (still unlabelled in practice) for each of them. P contains all its disease gene members. N is constructed by randomly picking genes from known disease genes such that $|N| = \frac{1}{2}|P|$. The unknown genes relate to at least one disease class, but do not relate to the current class. We chose the genes in N from the other disease genes because we assume that they were less likely to be associated to the considered class. In fact, disease genes are generally more studied and a potential association has more chances to have already been identified.

After that, leave-one-out cross validation was used to evaluate the performance of the algorithm. Iteratively, every gene in the training set was selected to be the test gene and the remaining genes in P and N were used to train the model. Once the model was trained, a score list for the test gene and the candidate genes was computed. Then, we computed a decision score for each test gene representing the percentage of candidate genes ranked lower than it. We collected all decision scores for every gene in all disease classes to form a global decision score list. The performance of Scuba was measured by calculating the area under the curve (AUC) in the receiver-operating-characteristic plot obtained from the decision score list. The AUC expresses the probability that a randomly chosen disease gene is ranked above a randomly picked non-disease gene for any disease class.

The second experiment (unbiased evaluation), we follow the setting used in *Unbiased evaluation* with the aim at comparing our method with different web tools for disease gene prioritization. Although the first evaluation is useful to compare our method with others, predictive performance in cross-validation experiments may be inflated compared to real applications. Indeed, the retrieval of known disease genes can be facilitated by various means. One mean is the crosstalk between data repositories: for example, KEGG [?] draws its information also from medical literature. Moreover, often the discovery of the link between a gene and a disease coincides with the discovery of a functional annotation or of a molecular interaction. In practice, instead, researchers are interested in novel associations, which in most cases are harder to find due to a lack of information around them.

In order to achieve a thorough evaluation of Scuba, we tested it in a more realistic setting, following the work of [?]. In this study, several gene prioritization tools were benchmarked as follows. Newly discovered disease-gene associations were collected over a timespan of six months, gathering 42 associations. As soon as a new association was discovered, eight pre-selected gene prioritization web tools were queried with a proper training set in order to mimic the discovery through each of them. These 42 predictions were used to assess the ability of the tools to successfully prioritize disease genes. The idea behind this procedure is to anticipate the integration of the associations in the data sources and so avoid biased predictions.

In order to test Scuba in this setting, we backdated our data to a time prior to May 15, 2010 by employing String v8.2 data [?]. The candidate sets were constructed by considering all genes with Ensembl [?] gene identifier within the chromosomal regions around the test genes, in order to get on average 100 candidates for each trial. Then, we performed prioritizations for each test gene in two distinct cases - genome-wide and candidate set-based prioritizations. In genome-wide prioritizations all coding genes in the genome were prioritized, while in candidate set-based tests only the genes belonging to the candidate groups were ranked. In both cases, we normalized ranking positions over the total number of genes in order to get the median and the standard deviation of the normalized ranks for test genes. We also computed the true positive rate (TPR) relatively to some representative thresholds (5%, 10% and 30% of the ranking) and the AUC obtained by averaging over the 42 prioritizations.

Data sources

- **Human Protein Reference Database (HPRD)**: The HPRD resource provides protein interaction data which we implement as an unweighted graph, where genes are linked if their corresponding proteins interact.
- **BioGPS**: It contains expression profiles for 79 human tissues, which are measured by using the Affymetrix U133A array. Gene co-expression, defined by pairwise Pearson correlation coefficients (PCC), is used to build an unweighted graph. A pair of genes are linked by an edge if the PCC value is larger than 0.5.
- **Pathways**: Pathway datasets are obtained from the database of KEGG, Reactome, PharmGKB and PID, which contain 280, 1469, 99 and 2679 pathways, respectively. A pathway coparticipation network is constructed by connecting genes that co-participate in any pathway.
- **String**: The String database gathers protein information covering seven levels of evidence: genomic proximity in procaryotes, fused genes, co-occurrence in organisms, coexpression, experimentally validated physical interactions, external databases and text mining. Overall, these aspects focus on functional relationships that can be seen as edges of a weighted graph, where the weight is given by the reliability of that relationship. To perform the unbiased evaluation we employed the version 8.2 of String, from which we extracted functional links among 17078 human genes.
- **Omim**: OMIM is a public database of disease-gene association. Genes implicated in the same disease are more likely to be involved in other similar diseases as well. Therefore, Omim network is formed by connecting genes which are involved in common disease(s).

5 RESULTS AND DISCUSSION

Table 1 and 2 show the performance of our proposed method together with different methods and tools in two experimental settings. It can be seen from tables that DIGI outperforms all compared methods in all cases. In the first experiment, it shows a bit higher than the second best one, Scuba and much higher than the rest of methods. In particular, DIGI shows around 5% higher than MRF and more than 15% compared to DIR and GeneWanderer. In the table 2, it presents impressive results in both Genome wide and Candidate-based settings. In disease gene prioritization, we pay more attention to the top genes in the ranking since we normally consider them to have actual medical tests to determin the relation of genes with a given genetic disease due to the time and financial constraints. The performance of DIGI on top 5% and 10% is far more higher than all tools and methods. Its average medians are only 6.56 and 8.00 for Genome wide and Candidate genes settings, respectively. In Genome wide setting, DIGI's average AUC performance is about 6% higher than the second best one, Scuba, meanwhile in candidate genes setting, it together with GeneDistiller are ranked first with 0.85.

6 CONCLUSION

We have proposed an efficient method for disease gene prioritization.

ACKNOWLEDGMENTS

This work was supported by the University of Padova, Strategic Project BIOINFOGEN

REFERENCES

[1] Rafal Ablamowicz and Bertfried Fauser. 2007. CLIFFORD: a Maple 11 Package for Clifford Algebra Computations, version 11. (2007). Retrieved February 28, 2008 from http://math.tntech.edu/rafal/cliff11/index.html

[2] Patricia S. Abril and Robert Plant. 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan. 2007), 36–44. https://doi.org/10.1145/1188913.1188915

[3] American Mathematical Society 2015. *Using the amsthm Package*. American Mathematical Society. http://www.ctan.org/pkg/amsthm.

[4] Sten Andler. 1979. Predicate Path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226–236. https://doi.org/10.1145/567752.567774

[5] David A. Anisi. 2003. *Optimal Motion Control of a Ground Vehicle*. Master's thesis. Royal Institute of Technology (KTH), Stockholm, Sweden.

[6] Mic Bowman, Saumya K. Debray, and Larry L. Peterson. 1993. Reasoning About Naming Systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (November 1993), 795–825. https://doi.org/10.1145/161468.161471

[7] Johannes Braams. 1991. Babel, a Multilingual Style-Option System for Use with LaTeX's Standard Document Styles. *TUGboat* 12, 2 (June 1991), 291–301.

[8] Malcolm Clark. 1991. Post Congress Tristesse. In *TeX90 Conference Proceedings*. TeX Users Group, 84–89.

[9] Kenneth L. Clarkson. 1985. *Algorithms for Closest-Point Problems (Computational Geometry)*. Ph.D. Dissertation. Stanford University, Palo Alto, CA. UMI Order Number: AAT 8506171.

[10] Jacques Cohen (Ed.). 1996. Special issue: Digital Libraries. *Commun. ACM* 39, 11 (Nov. 1996).

[11] Sarah Cohen, Werner Nutt, and Yehoshua Sagic. 2007. Deciding equivalances among conjunctive aggregate queries. *J. ACM* 54, 2, Article 5 (April 2007), 50 pages. https://doi.org/10.1145/1219092.1219093

[12] Bruce P. Douglass, David Harel, and Mark B. Trakhtenbrot. 1998. Statecarts in use: structured analysis and object-orientation. In *Lectures on Embedded Systems*, Grzegorz Rozenberg and Frits W. Vaandrager (Eds.). Lecture Notes in Computer Science, Vol. 1494. Springer-Verlag, London, 368–394. https://doi.org/10.1007/3-540-65193-4_29

[13] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. https://doi.org/10.1007/3-540-09237-4

[14] Ian Editor (Ed.). 2008. *The title of book two* (2nd. ed.). University of Chicago Press, Chicago, Chapter 100. https://doi.org/10.1007/3-540-09237-4

[15] Simon Fear. 2005. *Publication quality tables in LaTeX*. http://www.ctan.org/pkg/booktabs.

[16] Matthew Van Gundy, Davide Balzarotti, and Giovanni Vigna. 2007. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies (WOOT '07)*. USENIX Association, Berkley, CA, Article 7, 9 pages.

[17] David Harel. 1978. *LOGICS of Programs: AXIOMATICS and DESCRIPTIVE POWER*. MIT Research Lab Technical Report TR-200. Massachusetts Institute of Technology, Cambridge, MA.

[18] David Harel. 1979. *First-Order Dynamic Logic*. Lecture Notes in Computer Science, Vol. 68. Springer-Verlag, New York, NY. https://doi.org/10.1007/3-540-09237-4

[19] Maurice Herlihy. 1993. A Methodology for Implementing Highly Concurrent Data Objects. *ACM Trans. Program. Lang. Syst.* 15, 5 (November 1993), 745–770. https://doi.org/10.1145/161468.161469

[20] Lars Hörmander. 1985. *The analysis of linear partial differential operators. III*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. viii+525 pages. Pseudodifferential operators.

[21] Lars Hörmander. 1985. *The analysis of linear partial differential operators. IV*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. vii+352 pages. Fourier integral operators.

Table 1: Performance of different methods on MRF experimental setting

Method	**DIGI**	DIR	F3PC	MRF	Scuba
AUC	**0.881**	0.716	0.830	0.731	0.876

Table 2: Performance of DIGI comparing with Scuba and other web tools on unbiased setting

Tool/Method	Response rate	Rank median	TPR in top 5% (%)	TPR in top 10% (%)	TPR in top 30% (%)	AUC
Genome-Wide						
DIGI	100	**6.56**	**42.8**	**63.3**	**83.3**	**86.1**
Scuba	100	10.55	33.3	47.6	78.6	0.80
Candid	100	18.10	21.4	33.3	64.3	0.73
Endeavour	100	15.49	28.6	38.1	71.4	0.79
Pinta	100	19.03	26.2	31.0	71.4	0.77
Candidate-Genes						
DIGI	100	**8.00**	**38.1**	**59.5**	83.3	**0.85**
Scuba	100	12.95	28.6	45.2	73.8	0.78
Suspects	88.9[a]	12.77[a]	33.3[a]	33.3[a]	63.0[a]	0.76[a]
ToppGene	97.6	16.80	35.7	42.9	52.4	0.66
GeneWanderer-RW	88.1	22.10	16.7	26.2	61.9	0.71
Posmed-KS	47.6	31.44	4.7	7.1	23.8	0.58
GeneDistiller	97.6	11.11	26.2	47.6	78.6	**0.85**
Endeavour	100	11.16	26.2	42.9	**90.5**	0.82
Pinta	100	18.87	28.6	31.0	71.4	0.75

[22] IEEE 2004. IEEE TCSC Executive Committee. In *Proceedings of the IEEE International Conference on Web Services (ICWS '04)*. IEEE Computer Society, Washington, DC, USA, 21–22. https://doi.org/10.1109/ICWS.2004.64

[23] Markus Kirschmer and John Voight. 2010. Algorithmic Enumeration of Ideal Classes for Quaternion Orders. *SIAM J. Comput.* 39, 5 (Jan. 2010), 1714–1747. https://doi.org/10.1137/080734467

[24] Donald E. Knuth. 1997. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.)*. Addison Wesley Longman Publishing Co., Inc.

[25] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY.

[26] Leslie Lamport. 1986. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, MA.

[27] Newton Lee. 2005. Interview with Bill Kinder: January 13, 2005. Video. *Comput. Entertain.* 3, 1, Article 4 (Jan.-March 2005). https://doi.org/10.1145/1057270.1057278

[28] Dave Novak. 2003. Solder man. Video. In *ACM SIGGRAPH 2003 Video Review on Animation theater Program: Part I - Vol. 145 (July 27–27, 2003)*. ACM Press, New York, NY, 4. https://doi.org/99.9999/woot07-S422

[29] Barack Obama. 2008. A more perfect union. Video. (5 March 2008). Retrieved March 21, 2008 from http://video.google.com/videoplay?docid=6528042696351994555

[30] Poker-Edge.Com. 2006. Stats and Analysis. (March 2006). Retrieved June 7, 2006 from http://www.poker-edge.com/stats.php

[31] Bernard Rous. 2008. The Enabling of Digital Libraries. *Digital Libraries* 12, 3, Article 5 (July 2008). To appear.

[32] Mehdi Saeedi, Morteza Saheb Zamani, and Mehdi Sedighi. 2010. A library-based synthesis methodology for reversible logic. *Microelectron. J.* 41, 4 (April 2010), 185–194.

[33] Mehdi Saeedi, Morteza Saheb Zamani, Mehdi Sedighi, and Zahra Sasanian. 2010. Synthesis of Reversible Circuit Using Cycle-Based Approach. *J. Emerg. Technol. Comput. Syst.* 6, 4 (Dec. 2010).

[34] S.L. Salas and Einar Hille. 1978. *Calculus: One and Several Variable*. John Wiley and Sons, New York.

[35] Joseph Scientist. 2009. The fountain of youth. (Aug. 2009). Patent No. 12345, Filed July 1st., 2008, Issued Aug. 9th., 2009.

[36] Stan W. Smith. 2010. An experiment in bibliographic mark-up: Parsing metadata for XML export. In *Proceedings of the 3rd. annual workshop on Librarians and Computers (LAC '10)*, Reginald N. Smythe and Alexander Noble (Eds.), Vol. 3. Paparazzi Press, Milan Italy, 422–431. https://doi.org/99.9999/woot07-S422

[37] Asad Z. Spector. 1990. Achieving application requirements. In *Distributed Systems* (2nd. ed.), Sape Mullender (Ed.). ACM Press, New York, NY, 19–33. https://doi.org/10.1145/90417.90738

[38] Harry Thornburg. 2001. Introduction to Bayesian Statistics. (March 2001). Retrieved March 2, 2005 from http://ccrma.stanford.edu/~jos/bayes/bayes.html

[39] TUG 2017. Institutional members of the TeX Users Group. (2017). Retrieved May 27, 2017 from http://wwtug.org/instmem.html

[40] Boris Veytsman. [n. d.]. acmart—Class for typesetting publications of ACM. ([n. d.]). Retrieved May 27, 2017 from http://www.ctan.org/pkg/acmart