Computational Approaches to Combining Predictive Biological Models*

Douglas J. Slotta† Lenwood S. Heath Naren Ramakrishnan
Department of Computer Science

Rich Helm
Department of Wood Science and Forest Products

Malcolm Potts
Department of Biochemistry
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061, USA

March 11, 2002

Keywords: Bioinformatics, genomics, proteomics, cyanobacteria, order statistics.

Abstract

Predictive models of gene expression are evaluated against actual expression levels of genes under various cell conditions. The results are evaluated for *when* a model is a good predictor and for *which* genes the model is a good predictor; the individual models are then combined into a multi-model having better predictive power, under more conditions, than the standalone models.

INTRODUCTION

We are interested in developing strategies that ultimately lead to desiccation-tolerant human cells. The approach is biomimetic in nature, using the strategies employed by anhydrophilic cyanobacteria on various human cell lines. Desiccation tolerant cells would enable long-term storage at ambient temperatures, reducing costs for storage, while also providing a platform for biosensor development. Building computational models of the response of cyanobacteria to desiccation and rehydration is an important step in understanding the key physiological factors involved in this stress and can subsequently lead to the development of similar models for human cells. To permit such analyses we first selected *Escherichia coli* to develop the necessary tools for our long-term studies; the genome of *E. coli* is sequenced and there is a wealth of physiological and metabolic studies available that provide the requisite database. The physiological conditions available from one such study [1, 2] are shown in Table 1.

*Funding generously provided by Department of Defense (MURI) grant N00014-01-1-0852.

†Address all correspondence to slotta@csgrad.cs.vt.edu

When designing a computer system to model response of an organism to stress, it is necessary to employ biological models at different levels of abstraction. In the most abstract models, groups of genes that exhibit coordinated expression under experimental conditions are mined, represented, and visualized as a network. At lower levels of abstraction, the expression of individual genes (genomics) combined with the determination of *in vivo* quantities of thousands of proteins (proteomics) are evaluated and analyzed using clustering algorithms or inductive logic programming. A complete set of accurate models is far beyond current computational or modeling technology.

However, the models in this project have limited and tractable goals: to identify the genes and proteins important for desiccation tolerance and to predict gene and protein expression under combinations of conditions that can be inferred from actual experimental data. The experimental data come from numerous sources of diverse types, including known DNA sequences taken from a number of genome sequencing efforts, protein abundance at several different levels of desiccation, gene expression data, and *a priori* biological knowledge. From these data, models are created using a variety of methods — inductive logic programming, sequence and genome analysis, and statistical pattern recognition. For example, we have demonstrated that inductive logic programming is well suited to analyzing gene expression data in the presence of *a priori* information about gene function, while whole genome analysis enhances our ability to detect patterns in biological processes at the level of an entire cell [3].

All of these models give a useful, yet incomplete view of the processes taking place within the cell. To gain a more complete understanding, one should combine the results of these models into a consistent system-wide model by the use of information integration techniques. The end result is a graphical network with visual interpretations to aid in the human under-

Name	Description
AB	Abundance of proteins from cultures grown in glucose minimal MOPS at 7°C in alpha prime (a') units X 103. These units multiplied by 0.1 give the percent of total protein.
ACE	Relative level of proteins from cultures grown in acetate minimal MOPS compared to glucose minimal MOPS at 37°C
GLY	Relative level of proteins from cultures grown in glycerol minimal MOPS compared to glucose minimal MOPS at 37°C
RIC	Relative level of proteins from cultures grown in glucose rich (amino acid, bases and vitamins) MOPS compared to glucose minimal MOPS at 37°C
T13.5	Relative level of proteins from cultures grown in glucose rich (amino acid, bases and vitamins) MOPS at 13.5°C compared to 37°C
T15	Relative level of proteins from cultures grown in glucose rich (amino acid, bases and vitamins) MOPS at 15°C compared to 37°C
T23	Relative level of proteins from cultures grown in glucose rich (amino acid, bases and vitamins) MOPS at 23°C compared to 37°C
T30	Relative level of proteins from cultures grown in glucose rich (amino acid, bases and vitamins) MOPS at 30°C compared to 37°C
T42	Relative level of proteins from cultures grown in glucose rich (amino acid, bases and vitamins) MOPS at 42°C compared to 37°C
T46	Relative level of proteins from cultures grown in glucose rich (amino acid, bases and vitamins) MOPS at 46°C compared to 37°C

Table 1: Description of conditions.

Information storage and processing	
J	Translation, ribosomal structure and biogenesis
K	Transcription
L	DNA replication, recombination and repair
Cellular processes	
D	Cell division and chromosome partitioning
O	Posttranslational modification, protein turnover, chaperones
M	Cell envelope biogenesis, outer membrane
N	Cell motility and secretion
P	Inorganic ion transport and metabolism
T	Signal transduction mechanisms
Metabolism	
C	Energy production and conversion
G	Carbohydrate transport and metabolism
E	Amino acid transport and metabolism
F	Nucleotide transport and metabolism
H	Coenzyme metabolism
I	Lipid metabolism
Poorly characterized	
R	General function prediction only
S	Function unknown

Table 2: COG functional groups.

standing of the concepts, and which will, in turn, identify the promising areas of future data driven exploration.

A combined model should have predictive and even explanatory power in the context of desiccation tolerance in cyanobacteria. In particular, the model can be used as a substitute for costly biological experiments in isolating the essential biological processes for successful dessication tolerance and in conferring analogous tolerance in other cells or tissues through biological engineering. This model integration approach has the potential to contribute in related areas of bioinformatics.

Ideally, when combining models, one wants to take the best elements of each model to form the new model. Therefore, there needs to be some objective assessment of when a model performs well. Typically, two kinds of models are used for predictive gene expression. The first kind will, given a gene, say yes or no to it being expressed under certain conditions. The second kind assigns a score to the expression possibility of the gene. This is then usually matched against some arbitrary cutoff value with those scores exceeding the value considered to be significant. This practice misses important latent information in the data.

Consider the predicted highly expressed (PHX) method of Karlin and Mrázak [4, 5, 6]. PHX uses the codon bias of genes to determine the extent to which they are expressed. A score for each gene is computed and genes with a score above an arbitrary level are considered to be PHX. But the expression of genes in a cell is not an absolute. Some genes will be highly expressed, some will be somewhat expressed, and some will be expressed only under certain conditions. Since there is a ranking to the expression levels of genes within a cell, and there is a ranking of PHX values for a gene, it is natural that a comparison be made between the two ordered lists to see how close they are to each other.

Cells are not a static environment; the expression levels of genes change depending upon external conditions, or the current stage of the cell cycle. By comparing the model expression level list with actual expression levels of the genes under various physiological conditions, the validity of the model can be assessed for its applicability under those conditions. An automated method of combining models could be used by systems like *Expresso* [7] which refine and design microarray experiments.

In addition, genes can be broken down into meaningful categories and those subgroups can be compared to assess the predictive power of models (within the subgroup). A natural classification of proteins is evolving. *COG* stands for Cluster of Orthologous Groups of proteins [8, 9]. The proteins that comprise each COG are assumed to have evolved from an ancestral protein, and are therefore either orthologs or paralogs. Orthologs are proteins from different species that evolved by vertical descent (speciation), and typically retain the same function as the original. Paralogs are proteins from within a given species that are derived from gene duplication and may evolve new functions that are related to the original. Clusters of Orthologous Groups (COGs) were delineated by comparing protein sequences encoded in 21 complete genomes, representing 17 major phylogenetic lineages. Each COG consists of individual proteins or groups of paralogs from at least 3 lineages and thus corresponds to an ancient conserved domain. COG's are assigned to different functional groups as shown in Table 2.

METHODS

Model Prediction Evaluation

Let L be a list of items where $L = \{1, 2, \ldots, n\}$ and P be a permutation of list L where $P = \{p_1, p_2, \ldots, p_n\}$. If $i < j$ and $p_i > p_j$ then the pair (p_i, p_j) is an inversion [10]. The total number of inversions between two lists provides a measure of how sorted one list is with respect to another. A permutation with no inversions with respect to L has the exact same order as list L. A permutation with only one inversion has an adjacent pair of items switched. A permutation that has the reverse order of L has the maximum number of inversions possible.

The probability of a given permutation having an inversion count x can be determined by knowing the distribution of inversion counts for all possible permutations. The total number of permutations of a list is $n!$ and the minimum inversion count is 0, while the maximum inversion count is

$$\frac{n(n-1)}{2}.$$

The distribution of counts is approximately normal [10] with a mean (μ) of

$$\mu = \frac{n(n-1)}{4},$$

and a standard deviation (σ) of

$$\sigma = \sqrt{\frac{n(2n+5)(n-1)}{72}}.$$

Given the mean and the variance, the number of standard deviations (z-score) a permutation with inversion count x is away from the norm can be easily computed by

$$z = \frac{\mu - x}{\sigma}.$$

The higher the z score, the less likely that a particular ordering could have been created by random chance. The probability of a given count occurring is interpreted as the percentage of inversion counts that are equal to, or less than the current count.

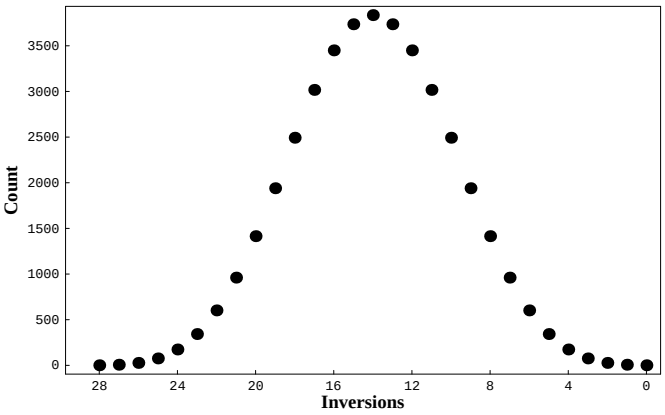

Figure 1: Distribution of inversion counts for a list of length 8.

Figure 1 shows the distribution of inversion counts for lists of length 8. Out of 40,320 permutations, 3,836 have 14 inversions, while only 7 have just a single inversion. If a permutation has a inversion count of 3, then 111 permutations have an equal or lesser score, which is only 0.275%.

There are two ways to interpret a model that gives a permutation with a negative z value (i.e., those whose inversion counts are higher than the mean). The first is that the model is just a really bad predictor. The second is to declare that the model is predicting the opposite of what actually occurs, so to find a better model, we can reverse the output of that model. For example, if model M_1 has a high, negative z-score under condition x, then create a new model M_2, where $M_2 = \text{Reverse}(M_1)$.

Standard	Complement
M_1	`Reverse(`M_1`)`
M_2	`Reverse(`M_2`)`

Table 3: Model possibilities.

Combining Models

Consider a list L and two of its permutations P_1 and P_2, posited by models M_1 and M_2 respectively. Let $l = \{l_1, l_2, \ldots, l_m\}$ be a subgroup of list L, $A = \{a_1, a_2, \ldots, a_m\}$ be the positions of the elements of l in permutation P_1, and $B = \{b_1, b_2, \ldots, b_m\}$ be the positions of the elements of l in permutation P_2. If the number of inversions of A is greater than the number of inversions of B then we say that $M_1 < M_2$ for that subgroup order. Let $(P_1 - A)$ be those elements in P_1 not in A. If the probability associated with $(P_1 - A)$ is greater than that associated with P_1, then that group is important to the ordering of P_1; if the probability associated with $(P_1 - A)$ is less than that associated with P_1, then that group is detrimental

Condition	Subgroup Order	Subgroup Support	Method of Combining
A	$M_1 < M_2$	$M_1 < M_2$	Use subgroup order and positions from M_2 in M_1
B	$M_1 < M_2$	$M_1 > M_2$	Use subgroup order from M_2 in M_1
C	$M_1 > M_2$	$M_1 < M_2$	Use subgroup positions from M_2 in M_1, but keep in the same order as M_1
D	$M_1 \geq M_2$	$M_1 \geq M_2$	Do nothing

Table 5: Possibilities for comparing subgroups in M_1 to M_2.

n	Condition A	B	C
4	94.7%	100.0%	90.9%
5	90.4%	100.0%	68.9%
6	92.4%	100.0%	86.5%
7	89.7%	100.0%	72.4%
8	92.6%	94.6%	81.6%
9	90.7%	94.9%	70.3%
10	92.9%	90.3%	77.7%
20	94.3%	85.2%	68.7%
30	95.1%	82.4%	64.9%

Table 4: Percentage of permutations of length $(1 \ldots n)$ whose order was improved using the subgroup of even numbers from a different permutation.

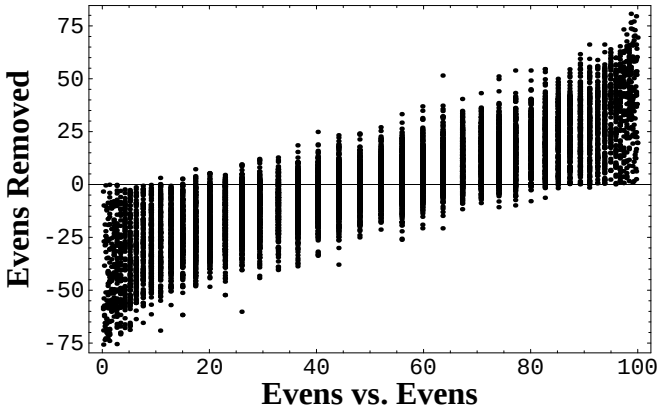

Figure 2: Distribution of the subgroup of even numbers for permutations of a list $\{1, \ldots, 30\}$.

to the ordering of P_1. There are thus four possible outcomes when comparing the output of one model to another for each subgroup, as shown in Table 5.

With the current model of evaluating the predictions of a model against known results, it is possible to combine models in a primitive fashion if the goal is well defined. For example, if there are two models, `ModelA` and `ModelB`, and `ModelA` performs better under physiological condition `X`, then the new combination model could be:

```
if (PhysCond = X)
  use ModelA;
else
  use ModelB;
```

Or the question could be posed, "What are the expression levels of genes for functional group K (transcription), under physiological condition GLY?" The results from the model that best predicts results for that functional group could then be chosen.

A better method involves improving the overall order by combining the outputs of different models, using knowledge of the subgroups that are better predicted under the individual models. For example, under physiological condition `X`, `ModelA` is a better predictor, with the exception of group E, which is better predicted by `ModelB`. The new combination model would be:

```
if (PhysCond = X)
  use ModelA(~E) + ModelB(E);
```

where $\sim$`E` captures the effect of removing subgroup E before `ModelA` is applied.

This solution to combining models in Table 5 is heuristic and is not always guaranteed to give results that are better then individual models. Table 4 shows the probability of the method of combination actually improving the overall score of a model given two random permutations meeting the criteria for the conditions. The list used was $\{1, \ldots, n\}$ and the subgroup was comprised of all of the even numbers in the list.

Figure 2 shows the distribution of the comparison between order of the subgroup of even numbers in P_1 vs. the order of the subgroup of even numbers in P_2 on the x-axis. The y-axis shows the difference in probabilities of P_1 and P_2 with the even numbers removed. The plot shows 10,000 random comparisons of permutations of a list of length 30 to another random permutation of the list. The total number of possible comparisons is $(n!)^2$, which is far too many to show here.

ANALYSIS

The following experiments take the results of two predictive biological models and compare the results with the actual expression levels of known genes from *E. coli* under various con-

ditions [1, 2]. At the time of this writing, Genbank listed 3907 genes for *E. coli*, of which approximately 46 to 80 genes had known expression levels under various conditions as shown in Table 1.

The first model used for these experiments was the PHX model mentioned previously [4, 5, 6]. The data were not taken from the paper, but were computed from the current known genomic data in Genbank. The second model (F25) was a modified version of PHX created by the authors based upon an article by Chen and Inouye [11] which stated that the bias of the first 25 codons of an open reading frame (ORF) was important to gene expression. Therefore the PHX algorithm was applied to only the first 25 codons in each gene.

The genes were assigned into functional groups shown in Table 2 using Clusters of Orthologous Groups (COGs) as defined in Tatusov et al. [8, 9]. Table 6 shows the relative orders of expression levels from the AB column of the *E. coli* data, and the orders predicted by the PHX and F25 models. The numbers following the gene name are either the expression levels (for column AB), or the scores predicted by the models.

AB	**PHX**	**F25**
aceE(8.5)	*atpA*(1.56)	*aceE*(1.06)
atpA(6.9)	*aceE*(1.53)	*sucA*(1.05)
atpD(5.6)	*pta*(1.53)	*aceF*(1.05)
lpdA(4.6)	*lpdA*(1.45)	*lpdA*(1.03)
ppc(4.3)	*ackA*(1.44)	*gltA*(1.02)
mdh(2.6)	*aceF*(1.34)	*atpA*(1.01)
sucA(1.9)	*atpD*(1.32)	*atpD*(1.01)
sucB(1.7)	*sucC*(1.29)	*sucB*(0.99)
sucC(1.6)	*mdh*(1.15)	*sdhA*(0.97)
pta(1.5)	*sucB*(1.11)	*sucC*(0.94)
sdhA(1.1)	*gltA*(1.05)	*ackA*(0.92)
gltA(1.1)	*sucA*(1.04)	*pta*(0.90)
ackA(1.0)	*sdhA*(0.90)	*gor*(0.89)
aceF(0.9)	*ppc*(0.74)	*mdh*(0.87)
gor(0.5)	*gor*(0.63)	*ppc*(0.86)

Table 6: Selected lists of gene expression levels for functional group C (energy production and conversion).

RESULTS

Table 7 shows the evaluation of the predictions made by the two models under various conditions. Note that, with the exception of condition T46, the PHX model is superior to the F25 model.

Figures 3 and 4 show the relationship between how well the model predicts the ordering within the group vs. how much the group supports the overall ordering. For the group to group ordering (y-axis), the value is the probability that an ordering this close could be random chance. So the lower the score, the better. Note that for Figure 3, some of the values exceed

	Model	
Conditions	PHX	F25
AB	0.0017%	13.098%
ACE	0.0073%	30.904%
GLY	0.0002%	17.602%
RIC	$<$0.0000%	7.1560%
T13.5	2.4240%	5.4816%
T15	0.8738%	2.0893%
T23	0.6733%	1.1250%
T30	0.4725%	1.2552%
T42	0.3266%	1.4391%
T46	21.894%	15.454%

Table 7: Likelihood of correspondence between model prediction and actual abundance levels.

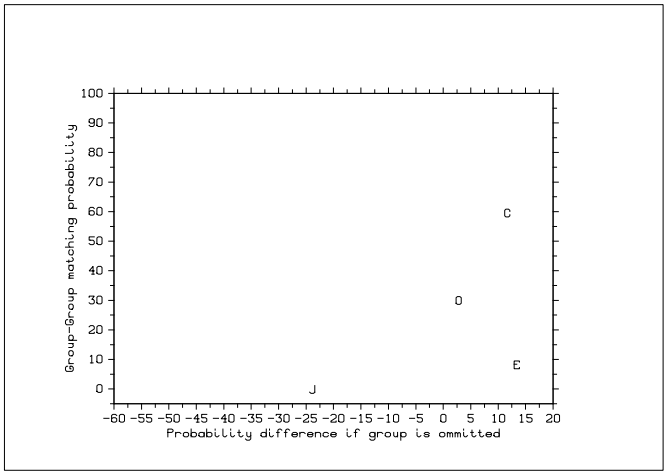

Figure 3: Group expression level prediction scores for Model PHX under physiological condition T46.

50%, so the other interpretation is possible, that this model is predicting the reverse order under those conditions. For the x-axis, the value is the difference in probability if this entire group is removed from the computation. For a negative value, the probability has gone down, so that group was important to the results. For a positive value, the probability has increased, so the inclusion of that group had a detrimental effect on the outcome.

Overall, model F25 is a better predictor under physiological condition T46 than PHX; however, group C under PHX is better predicted under physiological condition T46 than group C under F25. As can been seen from Figures 3 and 4, group C meets the criteria for condition C as shown in Table 5, therefore the positions of group C in PHX will be used in the combined model, but the order of group C from F25 will be preserved. The results are shown in Figure 5. The overall score of F25 was 15.454%, the score of F25 with the addition of group

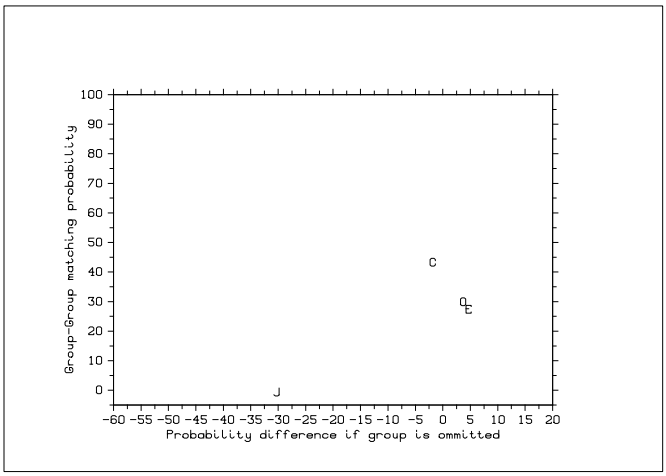

Figure 4: Group expression level prediction scores for Model F25 under physiological condition T46.

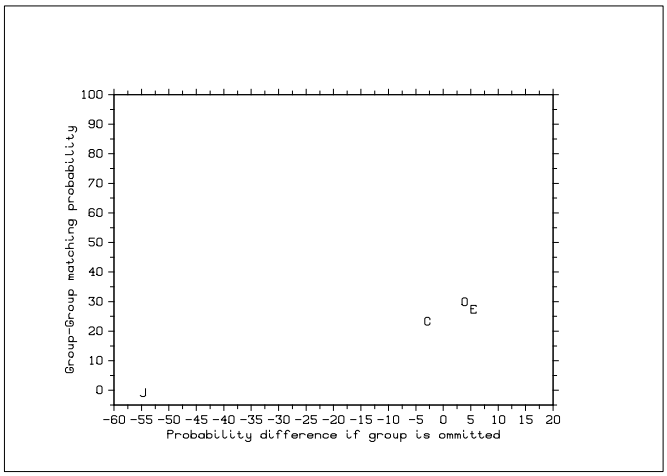

Figure 5: Group expression level prediction scores for the improved Model F25 under physiological condition T46.

C from PHX is 7.873%, a significant improvement.

FUTURE WORK

The heuristic used for combining models can be driven by a mathematical model of (expected) performance improvement. The first step in this direction is the assessment of where the current heuristic model fails and why. This knowledge can be used to refine the heuristic, or to develop an entirely new framework for combining predictive models. Especially relevant are formal frameworks to study model selection and combination, such as the Bayesian setting, risk minimization, and the minimum description length (MDL) principle [12].

References

[1] R. A. VanBogelen, K. Z. Abshire, A. Pertsemlidis, R. L. Clark, and F. C. Neidhardt., editors. *Escherichia coli and Salmonella: cellular and molecular biology*, chapter Gene-protein database of Escherichia coli K-12, edition 6, pages 2067–2117. ASM Press, Washington, D.C., 2nd edition, 1996.

[2] Han Tao, Christoph Bausch, Craig Richmond, Frederick R. Blattner, and Tyrrell Conway. Function genomics: Expression analysis of *Escherichia coli* growing on minimal and rich media. *Journal of Bacteriology*, 181(20):6425–6440, October 1999.

[3] Naren Ramakrishnan and Ananth Y. Grama. Data mining applications in bioinformatics. In Robert L. Grossman, Chandrika Kamath, Philip Kegelmeyer, Vipin Kumar, and Raju R. Namburu, editors, *Data Mining for Scientific and Engineering Applications*, pages 125–140. Kluwer Acedemic Publishers, 2001.

[4] Jan Mrázek, Devaki Bhaya, Arthur R. Grossman, and Samuel Karlin. Highly expressed and alien genes of the *synechocystis* genome. *Nucleic Acids Research*, 29(7):1590–1601, 2001.

[5] Samuel Karlin and Jan Mrázek. Predicted highly expressed genes of diverse prokaryotic genomes. *Journal of Bacteriology*, 182(18):5238–5250, September 2000.

[6] Samuel Karlin, Jan Mrázek, Allan Cambell, and Dale Kaiser. Characterizations of highly expressed genes of four fast-growing bacteria. *Journal of Bacteriology*, 183(17):5025–5040, August 2001.

[7] Ruth G. Alscher, Boris I. Chevone, Lenwood S. Heath, and Naren Ramakrishnan. Finding answers with microarray technology. In *Proceedings of the High Performance Computing Symposium, Advanced Simulation Technologies Conference*, pages 64–69, 2001.

[8] Roman L. Tatusov, Eugene V. Koonin, and David J. Lipman. A genomic perspective on protein families. *Science*, 278(5338):631–637, 1997.

[9] Roman L. Tatusov, Michael Y. Galperin, Darren A. Natale, and Eugene V. Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research*, 28(1):33–36, 2000.

[10] Donald E. Knuth. *The Art of Computer Programming*, volume 3 - Sorting and Searching, pages 11–22. Addison Wesley, second edition, 1998.

[11] G. F. Chen and M. Inouye. Suppression of the negative effect of minor arginine codons on gene expression; preferential usage of minor codons within the first 25 codons of the *Escherichia coli* genes. *Nucleic Acids Research*, 18(6):1465–1473, 1990.

[12] Vladmir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, 1995.