The German Conference on Bioinformatics (GCB) is an annual, international conference devoted to all areas of bioinformatics. Recent meetings attracted a multinational audience with 250 – 300 participants each year.

In 2014, the GCB was hosted by the bioinformatics groups of Bielefeld University. It is organized by the German Society for Chemical Engineering and Biotechnology (DEHEMA), and supported by the Special Interest Group on Informatics in Biology of the German Society of Computer Science (GI) and the Society for Biochemistry and Molecular Biology (GBM).

Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into
• seminars
• proceedings
• dissertations
• thematics

current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISBN 978-3-88579-629-9

GI-Edition

Lecture Notes in Informatics

Robert Giegerich, Ralf Hofestädter, Tim W. Nattkemper (Eds.)

German Conference on Bioinformatics 2014

September 28 – October 1
Bielefeld, Germany

Proceedings
Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-235

ISSN 1617-5468

Volume Editors
Prof. Dr. Robert Giegerich, Universität Bielefeld
E-Mail: robert@techfak.uni-bielefeld.de
Prof. Dr. Ralf Hofestäd, Universität Bielefeld
E-Mail: ralf.hofestaedt@uni-bielefeld.de
Dr.-Ing. Benjamin Kormeier, Universität Bielefeld
E-Mail: bkormeie@techfak.uni-bielefeld.de
apl. Prof. Dr.-Ing. Tim W. Nattkemper, Universität Bielefeld
E-Mail: tim.nattkemper@uni-bielefeld.de

Series Editorial Board
Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria
(Chairman, mayr@ifit.uni-klu.ac.at)
Dieter Fellner, Technische Universität Darmstadt, Germany
Ulrich Flegel, Hochschule für Technik, Stuttgart, Germany
Ulrich Frank, Universität Duisburg-Essen, Germany
Johann-Christoph Freytag, Humboldt-Universität zu Berlin, Germany
Michael Goedicke, Universität Duisburg-Essen, Germany
Ralf Hofestäd, Universität Bielefeld, Germany
Michael Koch, Universität der Bundeswehr München, Germany
Axel Lehmann, Universität der Bundeswehr München, Germany
Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany
Sigrid Schubert, Universität Siegen, Germany
Ingo Timm, Universität Trier, Germany
Karin Vosseberg, Hochschule Bremerhaven, Germany
Maria Wimmer, Universität Koblenz-Landau, Germany

Dissertations
Steffen Hölldobler, Technische Universität Dresden, Germany

Seminars
Reinhard Wilhelm, Universität des Saarlandes, Germany

Thematics
Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2014
printed by Köllen Druck+Verlag GmbH, Bonn
Preface

The German Conference on Bioinformatics is an annual, international conference, organized by the German bioinformatics community. Among the scientific meetings in the field, it is one of those with the longest tradition. Its roots go back to a workshop “Computeranwendungen in den Biowissenschaften” organized by Dietmar Schomburg, starting as early as 1986. Aside from exchange of scientific results, the GCB is also a place where joint strategies with respect to funding programs are discussed, where PIs go head hunting and fresh PhDs make contacts looking for their first postdoc position. Finally, a slot in one afternoon is reserved for the group meetings of the scientific societies that carry GCB: DECHEMI, GBM, and GI. In 2014, GCB comes to Bielefeld University for the second time, after visiting Bielefeld in 2004. Let us use the time span of a decade to look at the major change.

In 2004, “the” human genome sequence had been completed, further model organism genomes were nearing completion, and a good part of the community felt that genomics and the bioinformatics directly associated with genome sequencing would play a less prominent role in the next decade. “Next generation” sequencing technology, although already under development, left no mark at GCB 2004, which presented a balanced set of topics from basic sequence analysis, proteomics, regulatory systems, and data base work. The “1000 genomes” project was only to be defined in 2008! At the GCB 2004, there was but one paper pointing in the direction of dealing with more than one genome at a time: “Syntenic Layout of Two Assemblies of Related Genomes”.

Today, we find that next generation sequencing technologies have made a major impact on the questions we ask and the algorithmic problems we have to solve. Meta-genomics explores microbial communities in an unlimited number of habitats. In combination with single cell sequencing, we can venture deeply into the uncultivated microbial “dark matter”. RNA sequencing is used not only for expression analysis, but also for exploring alternative splicing, improving gene annotations and hunting for non-coding RNA genes. Two invited talks and three contributed papers at GCB 2014 address these topics. Classical topics such as regulation, proteomics, visualization and others still constitute the major part of the program. ”Big Data” production by sequencing projects calls for new data processing strategies – contrary to our expectation, we had almost no paper submissions on cloud computing; however, we could add a workshop on this topic.

While the program of GCB 2014 reflects the above shift of interest, the format of conference sticks to the GCB tradition, presenting a combination of keynote presentations, novel research and highlight papers, workshop and industry presentations, and a poster session, which we feel is particularly strong this year.

The program committee had been growing over the past few issues of the conference, approaching a point where almost all German groups that could possibly submit a contribution to the conference also had a representative in the program committee. This makes anonymous peer reviewing cumbersome, and we have decided to call for a generation change in the committee. Many of the GCB “founding fathers” have stepped back, putting
responsibility on the shoulders of the next generation.

To end with some statistics: GCB 2014 had 29 paper submissions, 12 of which were accepted for presentation. There were 6 invited talks, and 89 posters. The large number of posters caused us to add poster flashlight presentations, and to organize posters around thematic groups. Workshops/tutorials were given on “CELLmicrocosmos neXt”, “Information Visualization with Omix”, and “Cloud Computing”. Looking back at GCB 2004 once more, we find that it included a few contributions in the category “Software Demo”. This nice idea has gone lost over the past decade, and maybe it should be reconsidered in the future.

Robert Giegerich
Ralf Hofestdt
Tim W. Nattkemper
Organizers

Conference Chair

Jens Stoye (Universität Bielefeld)

Local Organizers

Susanne Konermann (Universität Bielefeld)
Tim W. Nattkemper (Universität Bielefeld)
Heike Samuel (Universität Bielefeld)
Alexander Sczyrba (Universität Bielefeld)
Roland Wittler (Universität Bielefeld)

Program committee

Mario Albrecht, Saarbrücken
Rolf Backofen, Freiburg
Jan Baumbach, Odense, Dänemark
Niko Beerwinkel, Zürich
Tim Beißbarth, Göttingen
Sebastian Böcker, Jena
Dmitrij Frishman, München
Robert Giegerich, Bielefeld
Alexander Goesmann, Gießen
Volker Heun, München
Ralf Hofestäd, Bielefeld
Ina Koch, Frankfurt
Oliver Kohlbacher, Tübingen
Stefan Kurtz, Hamburg
Hans-Peter Lenhof, Saarbrücken

Manja Marz, Jena
Alice McHardy, Braunschweig
Burkhard Morgenstern, Göttingen
Axel Mosig, Bochum
Tim Nattkemper, Bielefeld
Sven Rahmann, Essen/Duisburg
Matthias Rarey, Hamburg
Knut Reinert, Berlin
Falk Schreiber, Gatersleben
Stefan Schuster, Jena
Joachim Selbig, Potsdam
Jens Stoye, Bielefeld
Arndt Von Haeseler, Wien
Ralf Zimmer, München
Sponsors and Supporters

Supporting Scientific Institutions

DECHEMA Gesellschaft für Chemische Technik und Biotechnologie e.V.
http://www.dechema.de/

Fachgruppe “Informatik in den Biowissenschaften” der GI
http://www.cebitec.uni-bielefeld.de/groups/fg402/

GBM Gesellschaft für Biochemie und Molekularbiologie e.V.
http://www.gbm-online.de/

Internationales DFG Graduiertenkolleg 1906/1
http://www.didy.uni-bielefeld.de/

Universität Bielefeld
http://www.uni-bielefeld.de/

Sponsors and Donors

BLUEBEE
http://www.bluebee-tech.com/

GATC Biotech AG
http://www.gatc-biotech.com/

LEOXX GmbH
http://www.leoxx.net/

TimeLogic
http://www.timelogic.com/

MCS Moorbek Computer Systeme GmbH
https://www.mcs.de/

Omix Visualization GmbH und Co. KG
http://www.omix-visualization.com/
Sponsors and Donors

Pacific Biosciences
http://www.pacificbiosciences.com/

sysGen GmbH
http://www.sysgen.de/
# Table of Contents

**Preface** .......................................................... 5  
**Organizers** .......................................................... 7  
**Sponsors and Supporters** ........................................ 8  

**Pavankumar Videm, Dominic Rose, Fabrizio Costa, Rolf Backofen**  
*BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-Seq profiles* .......................................................... 12  

**Nicolas Terrapon, Andrew Moore and Erich Bornberg-Bauer**  
*Protein family analysis at the domain-level* ......................... 23  

**Yvonne Poeschl, Ivo Grosse and Andreas Gogol-Döring**  
*Explaining gene responses by linear modeling* ....................... 27  

**Franziska Zickmann, Martin S. Lindner and Bernhard Y. Renard**  
*RNA-Seq Driven Gene Identification* ................................ 36  

**Jan Grau, Stefan Posch, Ivo Grosse and Jens Keilwagen**  
*A general approach for discriminative de novo motif discovery from high-throughput data* .......................................................... 41  

**Magnus Rathke, Jan Kölling and Tim W. Nattkemper**  
*Interactive and dynamic web-based visual exploration of high dimensional bioimages with real time clustering* ......................... 44  

**Peng Sun, Jiong Guo and Jan Baumbach**  
*Efficient Large-scale Bicluster Editing* ............................... 54  

**Jana Tillack, Melanie Bende, Michael Rother, Maurice Scheer, Susanne Ulas and Dietmar Schomburg**  
*Flexible database-assisted graphical representation of metabolic networks for model comparison and the display of experimental data* . 61  

**Thomas Lingner and Peter Meinicke**  
*Characterizing metagenomic novelty with unexplained protein domain hits* .......................................................... 69
# Table of Contents

Eudes Barbosa, Richard Röttger, Anne-Christin Hauschild, Vasco Azevedo and Jan Baumbach  
*On the limits of computational functional genomics for bacterial lifestyle prediction*  ........................................... 79

Huda Al-Nayef, Christophe Guyeux and Jacques M. Bahi  
*A Pipeline for Insertion Sequence Detection and Study for Bacterial Genome*  .................................................. 85

Marko Djordjevic  
*Towards Accurate Transcription Start Site Prediction: a modelling approach*  ..................................................... 99

Posters  .......................................................... 106
BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-Seq profiles

Pavankumar Videm¹, Dominic Rose¹,², Fabrizio Costa¹, Rolf Backofen¹,³,⁴,⁵

¹Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg, Georges-Köhler-Allee 106, D-79110 Freiburg, Germany. ²Munich Leukemia Laboratory (MLL), Munich, Germany. ³Centre for Biological Signalling Studies (BIOSS), Albert-Ludwigs-University Freiburg, D-79104 Freiburg, Germany. ⁴Centre for Biological Systems Analysis (ZBSA), Albert-Ludwigs-University Freiburg, Habsburgerstr. 49, D-79104 Freiburg, Germany. ⁵Centre for Non-coding RNA in Technology and Health, University of Copenhagen, Gronnegardsvej 3, DK-1870 Frederiksberg C, Denmark.

{videmp, rose, costa, backofen}@informatik.uni-freiburg.de

Abstract: Sequence and secondary structure analysis can be used to assign putative functions to non-coding RNAs. However sequence information is changed by post-transcriptional modifications and secondary structure is only a proxy for the true 3D conformation of the RNA polymer. In order to tackle these issues we can extract a different type of description using the pattern of processing that can be observed through the traces left in small RNA-seq reads data. To obtain an efficient and scalable procedure, we propose to encode expression profiles in discrete structures, and process them using fast graph-kernel techniques.

We present BlockClust for both clustering and classification of small non-coding RNA transcripts with similar processing patterns. We show how the proposed approach is scalable, accurate and robust across different organisms, tissues and cell lines. BlockClust was successfully applied on a comprehensive set of eukaryotic data. It is the first tool for eukaryotic non-coding RNA analysis available on the galaxy framework.

1 Motivation

The study of non-coding RNAs (ncRNAs) is nowadays becoming important to fully understand cellular functions. On the one hand, most of the transcribed DNA is non-protein-coding [Jac09]; on the other hand ncRNAs play a vital role in many cellular processes. Although up to 450 000 ncRNAs were predicted in the human genome [RBT+10], the large majority is still missing functional annotation. Sequence and secondary structure analysis can be used to assign putative functions to ncRNAs, however sequence information is changed by post-transcriptional modifications [FLSH11], and secondary structure is only a proxy for the true 3D conformation of the RNA polymer. A different type of information that does not suffer from these issues and that can be used for the detection of RNA functional classes, is the pattern of processing.
that can be observed through the traces left in small RNA-seq reads data. For example, the primary microRNA transcript cleaved by the Drosha complex and forms hairpin like pre-miRNA with 2-nt 3’ overhang where Dicer binds and processes into double stranded miRNA and (complementary) miRNA* duplex [GST+08]. The miRNA strand then binds to Ago2 proteins to form RNA-induced silencing complex, which subsequently targets mRNA for regulation while the remaining miRNA* strand is degraded. Traces of this process are often observed in RNA-seq data of miRNA precursor as two adjacent piles of reads separated by few bases (length of hairpin). While one of the pile that corresponds miRNA strand is expressed, the other one that corresponds to the miRNA* strand, is not. Computational approaches such as mirDeep [FCA+08] rely on this miRNA biogenesis for annotation. Other examples involve snoRNAs, where snoRNA-derived fragments size and position distributions are conserved across species [TGL+09]. The tRNA molecules also undergo post-transcriptional cleavage to form smaller tRNA fragments which carry distinct expression levels and possibly different regulatory functions [GP13].

In this article, we propose BlockClust [VRCB14] as a novel technique to capture these processing patterns and detect transcripts that can have evolutionary relationship.

2 Methods

The core idea of the BlockClust is to characterize transcripts from small RNA-seq data by extracting characteristic attributes from their expression profiles. Those attributes are encoded into compact discrete structures, which can be processed using fast graph-kernel techniques to find similar expression profiles.

Given the mapped reads we consider only unique reads in the sample (tags). For each tag, the expression is normalized by dividing the number of reads associated with that tag by the number of times the tag is mapped to the reference genome. The notion of tags allows the elimination of duplicated data, hence speeding up subsequent processing. We use the blockbuster tool [LBSH+09] to identify consecutive tags with high expression and group them into blocks. Adjacent blocks, that are either overlapping or that are within a small distance, are then grouped into larger blockgroups. Here we assume that a ncRNA gene can span at most a single blockgroup. Each blockgroup is then encoded as a discrete graph. We consider different types of information, ranging from the information available for each individual block, to the relation between two consecutive blocks and finally also the information available globally on the whole blockgroup. For the whole blockgroup we measure quantities such as: the entropy of read starts, the entropy of read ends, the entropy of read lengths, the median of normalized read expressions and the normalized read expression levels in the first quartile. For each block we measure: the number of multi-mapped reads, the entropy of read lengths, the entropy of read expressions, the minimum read length and the block length. All measures are then discretized into a small number of discretization levels using an equal-frequency algorithm. The discretized attributes are then used to label the nodes of the resulting graph representation.
Figure 1 shows the graph encoding of a blockgroup with actual attributes used in the method. Each graph is made up of two disconnected components: the first one is used to encode the blockgroup attributes (shown as BLOCKGROUP ATTRIBUTES in Figure 1), while the second one represents the sequence of individual blocks and their attributes (shown as BLOCK ATTRIBUTES in Figure 1). Finally, the resulting graphs are processed using a fast graph kernel called Neighbourhood Subgraph Pairwise Distance Kernel (NSPDK) [CG10]. This type of kernel evaluates the similarity between two graphs as the fraction of neighbourhood subgraph pairs that are in common. This similarity notion is parametrized by the maximal size of the neighbourhood subgraphs and by the maximal distance allowed between the subgraphs in each pair. Intuitively this approach can be considered as an extension of the gapped k-mer similarity for strings to the graph domain. Formally: a neighbourhood graph is a subgraph specified by a root vertex \( v \) and a radius \( R \), consisting of all vertices that are at a distance (the distance between two vertices \( v \) and \( u \) on a graph is defined as the number of edges in the shortest path between \( v \) and \( u \)) not greater than \( R \) from \( v \). All pairs of such neighbourhood subgraphs whose root vertices are at a maximum distance \( D \) are extracted by the kernel (see Figure 2 for an illustration of the subgraph pair extraction by NSPDK). Since neighbourhood subgraphs can be efficiently enumerated in near linear time, the resulting approach has in practice linear complexity and can be used in large scale settings.
The resulting pairwise similarity matrix can be used with several existing clustering algorithms. In this work we used the Markov Cluster Algorithm (MCL) [EVDO02] on the nearest neighbour adjacency matrix.

When functional annotation is available, we can design a supervised task and build a classifier for each specific ncRNA family using kernelized Support Vector Machine models. Currently, we offer models for three families, namely: miRNA, tRNA and CD-box snoRNA.

3 Results and Discussion

We applied BlockClust on several datasets to evaluate the predictive performance and its robustness. Finally we have compared BlockClust to other state-of-art tools.

3.1 Datasets and processing

To train our predictive models we have used NGS data generated by Illumina sequencing of human embryoid body and embryonic stem cells, H1 cell line and IMR90 cell line (Development Data). In order to compare to other tools and evaluate the robustness of BlockClust we have used a comprehensive collection of test datasets (Benchmark Data), that includes 32 samples from human, mouse, fly, chimp, worm and plant in a variety of tissues and cell lines.

BlockClust is a pipeline that combining several tools namely: blockbuster, NSPDK and MCL. In order to achieve optimal predictive performance, we have optimized the hyper-parameters of each tool. For the blockbuster tool we need to specify the minimum distance between two blockgroups (cluster distance) and the standard deviation of a single read (scale); for NSPDK the radius and the distance are the parameters of choice in order to extract the neighbourhood subgraph pairs; in MCL the cluster granularities were controlled via the inflation and pre-inflation parameters.
In addition to the parameter optimization of the tools used, we also have to choose the number of discretization levels for attributes and select the most discriminative attributes. Table 1 shows the overview of the value ranges, the search step size and the selected optimal values for the aforementioned parameters.

<table>
<thead>
<tr>
<th>Component</th>
<th>Parameter</th>
<th>Interval</th>
<th>Step</th>
<th>Optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>blockbuster</td>
<td>Cluster distance</td>
<td>20–100</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>blockbuster</td>
<td>Scale of standard deviation</td>
<td>0.2–0.8</td>
<td>0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>Encoding</td>
<td>Discretization bins</td>
<td>3, 5, 7</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>NSPDK</td>
<td>Radius R</td>
<td>1, 3, 5, 7</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>MCL</td>
<td>Inflation</td>
<td>1–30</td>
<td>0.3</td>
<td>20</td>
</tr>
<tr>
<td>MCL</td>
<td>Pre-inflation</td>
<td>1–30</td>
<td>0.3</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 1: Parameter optimization. Overview of tools and probed parameter values and selected optimal values. Note that distance is defined as a function of radius: \( D = 2 \times R + 1 \).

All the parameters were optimized by splitting the Development Data into train/validation/test sets with sizes 35/35/30% respectively. Hyper-parameter were set using the train and validation sets, whereas the predictive performance is reported on the test set alone.

### 3.2 Performance of BlockClust

To assess the quality of the similarity notion generated by our approach, we measured the tendency for transcripts of functionally identical RNAs to be neighbours. We computed the Area Under the Curve for the Receiver Operating Characteristic (AUC ROC) using the distance as a predictor function to evaluate the quality of the induced metric; in addition we computed the purity of the partition generated by the MCL approach to evaluate the clustering quality (see Table 2).

<table>
<thead>
<tr>
<th>ncRNA class</th>
<th>#transcripts</th>
<th>AUC</th>
<th>#clusters</th>
<th>cluster purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>miRNA</td>
<td>168</td>
<td>0.896</td>
<td>10</td>
<td>0.855</td>
</tr>
<tr>
<td>tRNA</td>
<td>173</td>
<td>0.741</td>
<td>17</td>
<td>0.837</td>
</tr>
<tr>
<td>C/D-box snoRNA</td>
<td>78</td>
<td>0.731</td>
<td>7</td>
<td>0.683</td>
</tr>
<tr>
<td>H/ACA-box snoRNA</td>
<td>4</td>
<td>0.838</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rRNA</td>
<td>20</td>
<td>0.872</td>
<td>2</td>
<td>0.956</td>
</tr>
<tr>
<td>snRNA</td>
<td>7</td>
<td>0.637</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y_RNA</td>
<td>8</td>
<td>0.685</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Weighted average</td>
<td>458</td>
<td>0.805</td>
<td>36</td>
<td>0.813</td>
</tr>
</tbody>
</table>
Table 2: Clustering performance of BlockClust averaged over 10 random test splits of Development Data.

Out of 458 known transcripts in the test set miRNA, tRNA and C/D-box snoRNAs contribute to the majority. There are quite less number of known profiles from the remaining four classes. After clustering with MCL, we could capture only 2 clusters of rRNAs out of these four classes, while for the majority classes we got a decent number of clusters. On average we observed a good AUC of 0.8 for the the similarity notion. The best performance was found for miRNA in terms of similarity notion and cluster precisions, followed by rRNAs, tRNAs and C/D-box snoRNAs. Though H/ACA-box snoRNAs have a good AUC, due to their low population MCL could not cluster them together. Poor performance can be seen for Y_RNA and snRNA classes.

In Table we report instead the classification performance on the test set split of Development Data when we train family specific models in a one-vs-all setting. We chose Positive Predictive Value (PPV) and Recall as performance measures. The PPV for all three classes are very good (≈ 0.9). The miRNA model could successfully retrieve 89% of the miRNAs, while 80% and only 48% recalls were observed for tRNAs and C/D-box snoRNAs respectively.

<table>
<thead>
<tr>
<th>ncRNA class</th>
<th>#transcripts</th>
<th>PPV</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>miRNA</td>
<td>168</td>
<td>0.901</td>
<td>0.886</td>
</tr>
<tr>
<td>tRNA</td>
<td>173</td>
<td>0.899</td>
<td>0.796</td>
</tr>
<tr>
<td>C/D-box snoRNA</td>
<td>78</td>
<td>0.870</td>
<td>0.474</td>
</tr>
</tbody>
</table>

Table 3: Classification performance of BlockClust averaged over 10 random test splits of Development Data.

3.3 Comparison with other tools

We compared BlockClust to other tools that can process read profiles of small ncRNAs from RNA-seq data and perform predictions or clustering. The deepBlockAlign [LPE+12]) is a tool which uses a variant of Sankoff algorithm to align all input blockgroups and cluster them. DARIO [FLB+11] is a web server which is used for annotating miRNA, tRNA and snoRNAs from deep sequencing data using a random forest classifier. The comparison with deepBlockAlign was done on the whole Benchmark Data.

Note that since DARIO is not available as a standalone tool, we considered only one of the Benchmark Data (Gene Expression Omnibus (GEO) sample id: GSM769510) for comparison.

To compare the similarity notion of BlockClust and deepBlockAlign we computed AUC ROC on similarity matrices of both tools. Compared to deepBlockAlign, BlockClust performs better on average (AUC 0.84 vs. 0.7) and also in each individual class. See Table 4 for AUCs of both tools for each individual class and weighted average over all classes. In terms of computational complexity BlockClust is very competitive, achieving a 60-fold speed-up (50 seconds as compared to 58 minutes of deepBlockAlign on a dataset of ≈600 profiles). This is due to BlockClust quasi-linear complexity compared to the $O(m^2)O(n^6)$ complexity of the Sankoff algorithm used in deepBlockAlign (where $n$ is the number of blocks per instance and $m$ is the number of sequences).

<table>
<thead>
<tr>
<th>ncRNA class</th>
<th>#transcripts</th>
<th>BlockClust AUC ROC</th>
<th>deepBlockAlign AUC ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>miRNA</td>
<td>3869</td>
<td>0.925</td>
<td>0.714</td>
</tr>
<tr>
<td>tRNA</td>
<td>4988</td>
<td>0.795</td>
<td>0.701</td>
</tr>
<tr>
<td>C/D-box snoRNA</td>
<td>731</td>
<td>0.762</td>
<td>0.615</td>
</tr>
<tr>
<td>H/ACA-box snoRNA</td>
<td>142</td>
<td>0.859</td>
<td>0.720</td>
</tr>
<tr>
<td>rRNA</td>
<td>770</td>
<td>0.873</td>
<td>0.759</td>
</tr>
<tr>
<td>snRNA</td>
<td>240</td>
<td>0.698</td>
<td>0.610</td>
</tr>
<tr>
<td>Y_RNA</td>
<td>244</td>
<td>0.694</td>
<td>0.656</td>
</tr>
<tr>
<td>Weighted average</td>
<td>11061</td>
<td>0.839</td>
<td>0.700</td>
</tr>
</tbody>
</table>

Table 4: Comparison of BlockClust vs. deepBlockAlign on whole Benchmark Data.
In order to compare the precision of the clusters that can be obtained from BlockClust and deepBlockAlign, we applied MCL on similarity matrices from both tools. We used one sample from the Benchmark Data (GEO sample id: GSM450239) for comparison. The inflation and the pre-inflation parameters of MCL affect the cluster granularity, so by varying these parameters we obtained varying number of clusters for both tools. Figure 3 depicts the median cluster purities for number of clusters obtained at different inflations. In theory, with increasing number of clusters, the cluster sizes decrease. In turn, the smaller clusters tend to be more pure than larger ones. At all inflation settings, BlockClust produced less number of clusters with higher median precisions compared deepBlockAlign. Hence BlockClust potentially produces larger clusters with a higher precision.

Please note that the deepBlockAlign is an algorithm designed and optimized to identify similar processing patterns regardless of the ncRNA class. Therefore it might not give optimal results when used to cluster the ncRNAs into the families of their primary function.

<table>
<thead>
<tr>
<th>ncRNA class</th>
<th>miRNA PPV</th>
<th>miRNA Recall</th>
<th>tRNA PPV</th>
<th>tRNA Recall</th>
<th>snoRNA C/D-box PPV</th>
<th>snoRNA C/D-box Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlockClust</td>
<td>0.88</td>
<td>0.89</td>
<td>0.95</td>
<td>0.80</td>
<td>0.74</td>
<td>0.39</td>
</tr>
<tr>
<td>DARIO</td>
<td>0.85</td>
<td>0.81</td>
<td>0.92</td>
<td>0.88</td>
<td>0.46</td>
<td>0.52</td>
</tr>
</tbody>
</table>

Table 5: Comparison of classification performance of BlockClust against DARIO.

Compared to DARIO, BlockClust exhibits a better precision for all three ncRNA classes and also slightly better recall for miRNAs. Whereas, DARIO achieves a better
recall for the remaining two classes. Please refer to Table 5 for comparison of BlockClust and DARIO. Note that since DARIO is available only as a web server we could not reliably assess its run times.

### 3.4 Clustering analysis

To examine whether the BlockClust encoding of the attributes is discriminative enough to cluster ncRNA classes, we analysed the clusters generated by the BlockClust as follows. First we clustered all blockgroups in one sample from Benchmark Data (GEO sample id: GSM768988) using BlockClust. Then for each ncRNA family, we considered the clusters with highest precision. The hierarchical clustering of these cluster instances along with the representative expression profiles are shown in Figure 4.

![Figure 4: Hierarchical clustering of the BlockClust clusters of each family with highest precision. One representative read profile for miRNAs and snoRNAs, and two for tRNAs are shown. The annotation of the ncRNA can be seen under each profile as an horizontal bar.](image)

From clustering (Figure 4), we observed that the tRNA (blue) branch is constitute two different representative profiles for 5' and 3'- derived fragments. For miRNAs (purple), the classic 2-block profile can be found, where expressed block represents the miRNA and non-expressed block represents the degraded miRNA*. According to literature [TGL+09], the CD-box snoRNAs are mostly 5'-derived fragments. Surprisingly, in our example dataset, we observe CD-box snoRNAs with consistent 3'-derived fragments.

Finally, we investigate the tRNA (marked with *) that was clustered together with the miRNAs. Similar to miRNAs the read profile of this tRNA has a precisely cut 5'-derived
fragment (see top right corner box in Figure 4). It has been already demonstrated that such 5’-derived tRNA fragments could possibly processed by dicer as miRNAs [GP13] and have functional characteristics of miRNAs [MSS+13].

4 Conclusion

We presented **BlockClust**, an approach that can exploit processing traces of small ncRNAs to reliably and efficiently identify functional non-coding genes. We encode read expression profiles in compact discrete structures in order to use fast graph kernel approaches, obtaining competitive predictive performance and a significant speed-up compared to existing approaches. The complete work-flow of **BlockClust** and its tool dependencies are easily installable and usable from the galaxy [GNT+10] main toolshed: [http://toolshed.g2.bx.psu.edu/view/rnateam/blockclust_workflow](http://toolshed.g2.bx.psu.edu/view/rnateam/blockclust_workflow)

**Funding**

German Research Foundation (DFG-grant SFB 992/1 and BA 2168/3-1 to R.B.).

**References**


Protein family analysis at the domain-level

Nicolas Terrapon, Andrew D. Moore and Erich Bornberg-Bauer

Institute for Evolution and Biodiversity, University of Münster

ebb@uni-muenster.de

Abstract: The analysis of protein domains has gained considerable attention over the last years. Many new insights on protein modular evolution, combined with improved domain detection, have paved the way for an integrated analysis of protein families from a domain-centric perspective. We recently released DoMosaics, a JAVA application that facilitates the interactive analysis of protein domain arrangements. DoMosaics combines guided domain annotation, a highly-customisable visualization of arrangements, and a number of analysis tools. It also integrates domain-centric algorithms such as CODD, which is used for the detection of divergent domain occurrences that have escaped Pfam thresholds, as well as RADS/RAMPAGE which provides means to search for proteins with a domain arrangement similar to a given query. RADS provides an alignment of domain strings as opposed to amino-acid sequences, while RAMPAGE produces an amino-acid alignment guided by RADS results. Hence, RADS/RAMPAGE produces fast and yet accurate alignments, and associated ranking, of proteins with similar domain arrangements. Together, these tools greatly simplify the domain-centric analysis of protein function, structure and evolution.

1 Introduction

The evolution of gene-encoding proteins is not only driven by mutation, insertion or deletion of single nucleotides, but also involves the rearrangement of larger genomic regions which frequently correspond to protein domains. Domains are the smallest structural, functional, and evolutionary units of proteins. They usually vary in length between 100-250 amino-acids, except for short repeat motifs [CGVT03]. Domains became a cornerstone of protein annotation thanks to Hidden Markov Models (HMMs), a powerful approach that captures family diversity, and databases of domain families such as Pfam that cover a large part of the protein universe [FBC+14]. Most proteins contain only one domain, while multidomain proteins represent less than 33% of proteins encoded by genomes [WKCA11]. However, the sequential order of the domains in a protein, or a protein’s “domain arrangement”, can be subject to recombination and has been shown to be a major factor of evolution and novelty in complex multicellular organisms. In the past 15 years, many studies have provided insights into the underlying mechanisms of modular protein evolution. An important foundation was the observation that many domains are ancient and shared between all organisms, as well as some domain combination with a strong conservation of N- to C-terminus order [AGT01]. In a recent study, we observed that the majority of novel arrangements can be explained by simple rearranging processes such as fusion, fission and
terminal domain loss \cite{MGS13}. These properties notably gave rise to domain-based algorithms such as the Co-occurrent Domain Detection (CODD) which allows detection of divergent domains \cite{TGMB09} based on patterns of domain co-occurrence, or the recent RADS/RAMPAGE approach which can identify and align similar proteins based on their domain content \cite{TWG14}.

2 RADS/RAMPAGE \cite{TWG14}

A key task in the analysis of protein families is the identification of a protein set which share similar domain arrangements. RADS (Rapid Alignment of Domain Strings) determines the similarity between two proteins by aligning their domain arrangements, using a classical dynamic programming algorithm, and hence does not require any amino-acid sequence information. A key advantage to this approach is the reduction in time complexity: while proteins in UniProt contain on average 324 amino acids, they harbour an average of only 1.5 domains (2.6 for multi-domain arrangements). The second method, RAMPAGE (Rapid Alignment Method of Proteins based on domain ArranGEments) complements RADS and addresses the need for increased sensitivity. RAMPAGE creates global alignments of amino-acid sequences using the domain-wise alignments provided by RADS as a guideline (see Figure 1). RAMPAGE alleviates the problem of aligning single-domain arrangements with RADS and performs with a sensitivity similar to, but significantly faster than, BLAST. We demonstrated that these methods yield biologically meaningful results, which work at a speed that is significantly faster than classical local alignment tools. We provided a fast C-based command-line application for running custom domain-string comparisons, a web interface for querying UniProt with Pfam and a command-line JAVA application for querying the web interface in batch mode, which can also be used as a JAVA library for programmatic access. To satisfy the need for a tool which simplifies the analysis of protein families by uniting these new approaches and by offering powerful visualization abilities, we developed DoMosaics \cite{MHT14}.

3 DoMosaics \cite{MHT14}

DoMosaics is a Java application that unifies protein domain annotation, domain arrangement analysis and visualization in a single tool. It simplifies the analysis of protein families by unifying disjunct procedures based on partly inconvenient command-line based applications and complex analysis tools. DoMosaics provides easy, GUI-based access to domain-annotation services such as InterPro, and can work without internet connection after downloading HMMER binaries, at http://hmmer.janelia.org, and HMM libraries from public domain databases as Pfam \cite{FBC14}. It can be used to analyze the change of domain arrangements along a phylogenetic tree, construct domain-guided dotplots and domain co-occurrence graphs, and perform divergent domain detection with CODD \cite{TGMB09}, and retrieve proteins with similar domain arrangements using queries to Uniprot with
Figure 1: Principle of the domain-driven amino-acid alignment by RAMPAGE (left panel). The right panel illustrates a query A2VP54 with RADS that recognizes the similar arrangement of C7TXY1, while BLAST fails to find it with an e-value threshold of 0.1 but finds the distant arrangement of C9Z9C4 at $10^{-6}$.

Figure 2: DoMosaics view of one protein family and domain-wise dotplot.

RADS/RAMPAGE [TWG+14]. Finally, DoMosaics allows for highly-customisable visualization, and can export high-quality, publication-ready images of protein domain arrangements.

References


Explaining gene responses by linear modeling

Yvonne Poeschl, Ivo Grosse, and Andreas Gogol-Döring

German Center of Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, Germany
Institute of Computer Science, Martin Luther University Halle–Wittenberg, Germany

yvonne.poeschl@informatik.uni-halle.de

Abstract: Increasing our knowledge about molecular processes in response to a certain treatment or infection in plants, insects, or other organisms requires the identification of the genes involved in this response. In this paper, we propose the Profile Interaction Finder (PIF) to identify such genes from gene expression data which is based on a convex linear model, and we investigate its efficacy for two applications related to stimulus response. First, we seek to identify sets of putative regulatory genes that explain the expression levels of a gene under different stimuli best. Second, we aim at identifying genes that show a specific response to a stimulus or a combination of stimuli. For both applications, we study the expression response of two Arabidopsis species to treatment with the plant hormone auxin and of Apis mellifera to pathogen infection. The proposed approach may be of general utility for analyzing expression data with a focus on genes and gene sets that explain specific stimulus response.

1 Introduction

Genes in a living cell form a complex network in which the expression level of each gene, i.e., the concentration of messenger RNA molecules, depends on the expression level of other genes. For instance, the expression of a gene encoding a transcription factor (TF) could rise because of an external stimulus, which consequently influences – directly or indirectly – the transcription of tens or hundreds of other genes.

The investigation of the causal effects between one TF and its target genes is a difficult task requiring complex laboratory experiments. Fortunately, it is possible to get indications of potential regulatory relationships between genes by comparing their expression levels under different conditions, e.g., before and after stimulation. A variety of methods have been developed for this purpose, for a recent review see [WH14]. The higher the number of the involved data sets, and the more different conditions (treatments, time points, cell types, pathogens, etc.) are covered, the more detailed and accurate a prediction of the regulatory network could be. The underlying assumption is that genes that are closely connected in the regulatory network will also tend to have similar expression patterns under varying conditions. Mathematically speaking, gene expression levels obtained from $M$ experiments can be represented by an $M$-dimensional vector, and if two genes are neighbors within this $M$-dimensional space, they presumably have a tight relation to each
other in terms of their regulation.

A conventional clustering method like HCLUST [MC11] relying only on the relation between pairs of genes sometimes fails to model cases in which one gene is jointly regulated by several other genes while it is only loosely correlated with each individual regulator. The Local Context Finder (LCF) introduced by Katagiri and Glazebrook [KG03] addresses this problem by reconstructing the \( M \)-dimensional expression profile of a gene as a linear combination of the expression profiles of other (neighboring) genes. One limitation of this approach is that it does not regard anti-correlated expression profiles. Although it is well known that, e.g., TFs could either increase or suppress the transcription of target genes, the latter case is not considered by the LCF.

In this paper we propose a new approach, the Profile Interaction Finder (PIF), that uses a distance metric that takes into account both positive and negative correlations. The approach selects for each gene a set of neighboring reference profiles that together explain the expression values of the gene best. Reference profiles are either expression profiles of other genes, which possibly have a regulatory influence on the current gene, or prototype profiles that reflect in which data set a certain experimental condition was present or absent. The proposed approach extends the LCF in two aspects, namely by considering both positive and negative interactions, and by using the flexible and generalizing notion of reference profiles. These extensions are instrumental in answering two central questions when analyzing expression data: (i) Which genes might have a positive or negative influence on the expression pattern of other genes?, And (ii) which genes respond positively or negatively to certain experimental conditions?

## 2 Methods

Supposed that we measure the expression of genes under varying conditions in \( M \) different experiments. To each gene we assign an expression profile \( \mathbf{x} = (x_1, \ldots, x_M) \) containing the expression values of this gene. All expression profiles are normalized using a linear transformation such that the length \( \| \mathbf{x} \| = 1 \) and the mean \( \bar{\mathbf{x}} = 0 \). This normalization does not affect the Pearson correlation coefficient between two profiles \( \mathbf{x} \) and \( \mathbf{y} \), but it simplifies its calculation as the dot product \( \mathbf{x} \cdot \mathbf{y}^T \) which can be interpreted as the cosine of the angle between the two vectors.

The goal of the proposed algorithm is to approximate a given expression profile by a linear model of reference profiles that could be either expression profiles of other genes or artificially created prototype profiles describing experimental conditions. Supposed for example that we set \( n_m = 1 \) if the \( m \)-th experiment is measured under a certain condition \( c \), and \( n_m = 0 \) otherwise, then \( \mathbf{n} = (n_1, \ldots, n_M) \) is after normalization a prototype profile for the condition 'measured on condition \( c \)’. More detailed examples for prototype profiles will be given in Section 3.2.

PIF returns for each gene \( \mathbf{x} \) a set of neighboring profiles which are most informative for predicting \( \mathbf{x} \). The proposed approach consists of three steps: (i) PIF first selects candidate reference profiles \( \mathbf{n}_1, \ldots, \mathbf{n}_K \) related to \( \mathbf{x} \) (Section 2.1), which (ii) are used to reconstruct \( \mathbf{x} \)
by a linear model (Section 2.2), and finally (iii) the results are filtered using bootstrapping
(Section 2.3).

If gene expression profiles are used as references, the output could be interpreted as a
gene regulatory network in which every gene is linked to all genes in its neighborhood.
In case of prototype profiles, the genes could be sorted into clusters according to their
neighborhoods. Examples for both applications are discussed in Section 3.

2.1 Selection of Reference Profiles

Fitting a linear model to a given input profile \( \mathbf{x} \) could be computationally demanding,
especially if the number of reference profiles is large. We therefore restrict the calculation
to the subset of reference profiles that are most appropriate for reconstructing the input
profile. This filtering process reduces computational costs and also improves the quality
of the reconstruction by reducing noise.

For scoring the predictive power of a reference profile \( \mathbf{n} \) relative to \( \mathbf{x} \), we first compute the
Pearson correlation coefficient between the two profiles. If this value is either close to 1
(positive correlation) or close to \(-1\) (anti-correlation), then the two profiles are strongly
connected, and in both cases the reference profile would be appropriate for reconstructing
the input profile. A correlation coefficient of 0 on the other hand means that both vectors
are orthogonal and no information about the input profile could be derived from the ref-
ERENCE profile. The absolute value of the correlation coefficient \( s = |\mathbf{x} \cdot \mathbf{n}^T| \) is a good
indicator for the applicability of \( \mathbf{n} \) for reconstructing \( \mathbf{x} \). In contrast to the LCF given in
[KG03], which only chooses reference profiles with maximum positive dot product, PIF
also takes highly informative reference profiles with negative dot product into account.

We select at most \( K \) profiles with maximal score \( s \geq t \), where \( t \) is a user-defined threshold.
A high value of \( t \) ensures that only reference profiles in close proximity to the input profile
are used, whereas with \( t = 0 \) the filtering step would be omitted completely. In this paper
we use \( K = 10 \) and \( t = 0.25 \).

2.2 Linear Model Reconstruction

In the main step of our approach, we reconstruct the input profile \( \mathbf{x} \) as a linear combina-
tion of the reference profiles \( \mathbf{n}_1, \ldots, \mathbf{n}_K \) selected in step (i) (Section 2.1). We calculate
non-negative weights \( w_1, \ldots, w_K \) by a constrained linear fit such that the squared error
function \( f(\mathbf{w}) \) is minimized,

\[
f(\mathbf{w}) = \left\| \mathbf{x} - \sum_{k=1}^{K} w_k \mu_k \mathbf{n}_k \right\|^2 \text{ and } 1 = \sum_{k=1}^{K} w_k,
\]  

(1)
where \( \mu = (\mu_1, \ldots, \mu_K) \in \{-1, 1\}^K \) denotes the signs of the dot products \( x \cdot n_k^T \), i.e., \( \mu_k = 1 \) if \( x \cdot n_k^T \geq 0 \), and \( \mu_k = -1 \) if \( x \cdot n_k^T < 0 \). For reference profiles \( n_k \) that are anti-correlated to \( x \) the factor \( \mu_k = -1 \) reverts the direction of the reference profile such that the resulting profile \( v_k = \mu_k n_k \) and \( x \) are positively correlated. This reduces the reconstruction to a convex linear combination, where all weights \( w_k \) are non-negative and sum to one.

We reformulate the optimization problem by including the constraint on the weights by introducing the Lagrangian multiplier \( \lambda \):

\[
L(w, \lambda) = \left\| x - \sum_{k=1}^{K} w_k v_k \right\|^2 + \lambda \left( 1 - \left( \sum_{k=1}^{K} w_k \right) \right)
\]

We minimize \( L(w, \lambda) \) in eq. 2 by computing the derivatives for all \( w_k \) and then use the constraint in eq. 2 to compute \( \lambda \), yielding

\[
w_k = \sum_{j=1}^{K} s_{k,j}^{-1} \left( \frac{\lambda}{2} + v_j x^T \right) \\
\lambda = 2 \cdot \frac{1 - \left( \sum_{j=1}^{K} v_j x^T \left( \sum_{k=1}^{K} s_{k,j}^{-1} \right) \right)}{\sum_{j=1}^{K} \sum_{k=1}^{K} s_{j,k}^{-1}},
\]

where \( v_j = \mu_j n_j \), and \( s^{-1} \) is the inverse of \( s = v \cdot v^T \) with \( v = (v_1, \ldots, v_K)^T \). If \( s \) becomes singular due to the linear dependence of some reference profiles, we compute the pseudo-inverse as suggested by [RS00].

The intended reconstruction of \( x \) is then given by the linear combination \( r = \sum_{k=1}^{K} w_k \mu_k n_k \).

### 2.3 Determining Robust Neighborhoods

The weights \( w_1, \ldots, w_K \) calculated in the previous section can be interpreted as degrees of relative importance of the reference profiles \( n_1, \ldots, n_K \) for the explanation of an expression profile \( x \). Reference profiles \( n_k \) with a low weight \( w_k \) are likely expendable. Given a user-defined threshold \( r \), we call the set \( \{ n_k | w_k \geq r \} \) of all reference profiles with weights of at least \( r \) the neighborhood of \( x \). In this paper, we set \( r = 0.1 \).

The approach comprised of step (i) and (ii) (Section 2.1 and 2.2) described so far could be affected by noise in the gene expression data. Hence, we use bootstrapping in order to increase the reliability of the results. Given a data set with \( M \) samples, bootstrapping samples \( M \) out of these \( M \) samples with replacement, and we apply PIF to this sampled data set. We perform this bootstrapping step \( L = 1000 \) times and keep only reference profiles in the neighborhood of a gene which occurred in this neighborhood for at least \( p \) percent of the \( L \) repeats. In this paper, we use a thresholds of \( p = 50\% \) for gene expression reference profiles (Section 3.1) and \( p = 75\% \) for prototype profiles (Section 3.2).
3 Results

We will now investigate if PIF is capable of producing biologically relevant results when applied to reconstructing gene regulatory networks (Section 3.1) and to clustering genes according to experimental conditions (Section 3.2).

3.1 Reconstruction of Regulatory Networks

Auxin is one of the key phytohormones that controls plant development and growth. So far, only parts of auxin signaling are understood [DRQ08]. For the identification of novel candidate genes that might be involved in auxin signaling network, we applied PIF on a time-series of gene expression data of the two closely related plant species *Arabidopsis thaliana* and *Arabidopsis lyrata*, measured using expression microarrays at 0, 1, and 3 hours after auxin treatment. Each measurement was repeated three times, yielding $M = 2 \times 3 \times 3 = 18$ data sets.

We processed and normalized the raw data as described in [P+13]. 9091 genes with a coefficient of variation above 0.05 were selected for further analysis. Each of these genes could be regulated either enhanced or repressed by any of the other genes, so we used the expression profiles of all 9091 genes as possible reference profiles.

Figure 1 shows a part of the reconstructed gene network connected to the well known auxin responsive gene AT5G54510 that is up-regulated upon auxin stimulation. According to the PIF analysis, AT5G54510 is part of the neighborhoods of four other genes. The correlation coefficients between AT5G54510 and the two genes AT3G58190 and AT4G37295 are positive, so AT5G54510 might have an enhancing effect on their expression. In contrast to that, the correlation coefficients to the other two target genes AT4G10270 and AT3G10040 are negative, suggesting that AT5G54510 possibly suppresses their transcription.

None of the four genes related to AT5G54510 had been identified to be involved in the auxin signaling pathway. Nevertheless, especially AT3G58190 seems to be very likely involved in hormone signaling, since this gene is also connected to two more factors AT4G14560 and AT4G27260 both related to the hormone metabolism.

3.2 Prototype Analysis

In addition to the reconstruction of gene regulatory networks we can use the *Arabidopsis* data from the previous section to address various further questions. Examples are: ‘Which genes respond quickly, or with a delay to auxin stimulation?’ or ‘Which genes are regulated differently in the two species?’ PIF is capable of answering these questions by using prototype profiles that reflect the different time points and species of the data sets (Figure 2A). Figure 2B-D shows an example of the results of this analysis, a cluster of 16 genes initially highly expressed in both species and later down-regulated, but more
strongly in *Arabidopsis thaliana* than in *Arabidopsis lyrata*.

This expression pattern is described by a combination of three prototype profiles (Figure 2B). Each single prototype profile differs strongly from the expression profiles of the genes in this cluster (Figure 2C and D), so the cluster could only be found because PIF reconstructs expression profiles by combining several reference profiles (Section 2.2).

Statistical analysis reveals that for the GO-term [T+04] ‘RNA’ the number of annotated genes in this cluster is significantly higher than expected (p-value > 0.05, Fisher’s exact test). This indicates that PIF possibly sorted the genes into biological meaningful clusters.

To investigate if PIF could also handle more diverse input data, we applied it to multiple data sets collected for a metastudy [Tra14] concerning the impact of different pathogens on gene expression in honeybee (*Apis mellifera*), see Table 1. The expression data were collected from different sources, measured for different tissues and on different platforms, and preprocessed with different methods, so they have very different dynamic ranges. Hence, we decided to use relative ranks [BAAH04] instead of raw gene expressions as input for PIF.

We group 6242 genes present in all 9 data sets according to their response pattern to different experimental conditions, namely pathogens and tissues, see Figure 2A. Figure 2B-D show the example of a gene cluster containing 15 genes that respond positively to nosema infection in the fat body but negatively in the gut. Gut and fat body are distinct parts of the honeybee abdomen; genes in this group may be related to the immune response activated due to the infection. Although the individual genes within the clusters are more diverse than in the data set for *Arabidopsis*, their expression profiles broadly follow the pattern defined by the prototypes.
Figure 2: PIF analysis using prototype profiles as reference. The left panel shows results of the *Arabidopsis* data analysis, and the right panel shows results of the *Apis mellifera* metastudy. A: The complete set of prototype profiles (before normalization) used in the analysis. B: Neighboring prototype profiles for one selected gene cluster. Prototype profiles which correlate positively to the genes in the cluster are shown in red; anti-correlations are shown in green. C: Averaged expression profiles of the genes in the cluster. The orange boxes show the area between the first and the third quartile. D: Heat maps showing the expression profiles of genes in the cluster. Each line represents one gene. Red boxes show highly expressed/up-regulated genes, and green boxes show low expressed/down-regulated genes.
4 Conclusions

The identification of genes acting as regulators of other genes or responding specifically to certain experimental conditions is an important aspect of gaining knowledge about gene regulatory processes in response to a treatment or infection. In this paper, we propose PIF, the Profile Interaction Finder, a novel approach that can be applied to expression data sets in order to tackle these questions.

Studying data sets of *A. thaliana* and *A. lyrata* after auxin treatment, and of *A. mellifera* after infection with different pathogens, PIF successfully identified genes related to the cell responses for the respective stimulus. In addition to that, PIF determined novel putative regulators that might affect several other genes in the downstream response. The detected targets of the *Arabidopsis* gene AT5G54510 for example had not yet been identified to be involved in the auxin signaling pathway. This shows that PIF is capable to discover previously unknown relationships between genes. The obtained results are highly relevant, as shown by linking them to already existing biological knowledge, represented for example in the gene ontology. Being capable to identify not only enhancing but also suppressing regulators is another advantageous feature of PIF. For example, with our method we were able to find two genes which are possibly down-regulated by AT5G54510.

Hence we conclude that PIF is a valuable tool for getting deeper insights into biological processes by analyzing gene expression data under varying experimental conditions.

5 Acknowledgements

We thank Carolin Delker, Jan Grau, Marcel Quint, Jana Trenner, and all participants of the Trans-Bee workshop for valuable discussions.

The honeybee transcriptome data used in Section 3.2 were collected and analyzed for the project Trans-Bee [Tra14], which was kindly supported by sDiv, the Synthesis Centre for Biodiversity Sciences – a unit of the German Centre for Integrative Biodiversity Research.
(iDiv) Halle-Jena-Leipzig, funded by the German Research Foundation (FZT 118).

References


RNA-Seq Driven Gene Identification
Franziska Zickmann, Martin S. Lindner and Bernhard Y. Renard

Research Group Bioinformatics (NG4)
Robert Koch-Institute
Nordufer 20
13353 Berlin, Germany
zickmannF@rki.de
renardB@rki.de

Abstract: The reliable identification of genes is a challenging and crucial part of genome research. Various methods aiming at accurate predictions have evolved that predict genes \textit{ab initio} on reference sequences or \textit{evidence based} with help of additional information. With high-throughput RNA-Seq data reflecting currently expressed genes, a particularly meaningful source of information has become commonly available. However, a particular challenge in including RNA-Seq data is the difficult handling of ambiguously mapped reads. Therefore we developed GIIRA, a novel gene finder that is exclusively based on RNA-Seq data and inherently includes ambiguously mapped reads. Evaluation on simulated and real data and comparison with existing methods incorporating RNA-Seq information highlight the accuracy of GIIRA in identifying the expressed genes. Further, we developed a framework to integrate GIIRA and other gene finders to obtain a verified and accurate set of gene predictions.

1 Introduction

Accurate gene identification is an important and also challenging part of genome analysis pipelines. Hence, various gene finders have evolved, which are categorized as \textit{ab initio} and \textit{evidence based} (including \textit{comparative}) gene finders. Ab initio approaches predict genes exclusively on the target sequence and perform identifications based on training data and strategies such as Hidden Markov models. In contrast, evidence based methods report genes depending on observed evidence, such as EST libraries or protein sequences. Further, there exist hybrid approaches that combine ab initio gene prediction with additional evidence to verify the predicted genes [GKE12].
Since none of these strategies is bias-free, also methods combining predictions have evolved. These approaches introduce weighting schemes to score different predictions and merge the output of different input gene finders.

Despite all efforts, gene identification still faces significant challenges handling complex gene structures, rare splice sites or mutations in genes [GKE12]. These problems can be overcome by using the knowledge available from high-throughput RNA-Seq experiments, which reflect genes expressed under experimental conditions and hence provide valuable information to identify novel or confirm predicted genes. One challenge in including RNA-Seq information is the presence of ambiguous reads, which map to several locations in the genome and therefore complicate the gene prediction. However, discarding ambiguously mapped reads may result in a significant loss of prediction accuracy, since for instance repetitive or highly similar regions or paralogous genes lead to a substantial part of non-unique mappings.

To utilize the complete information contained in RNA-Seq experiments for gene identification we developed GIIRA (Gene Identification Incorporating RNA-Seq and Ambiguous reads), a de novo gene predictor that works on a reference genome and reads derived in a RNA-Seq experiment [ZLR14]. Since GIIRA is also excellently suited to be combined with predictions from other gene finders we further developed an easy-to-use framework to merge results of different prediction strategies. This allows to combine advantages of diverse methods to obtain a verified and accurate set of gene predictions [ZR14].

2 Methods and Results

![Workflow of GIIRA](image)

Figure 1: Workflow of GIIRA: Given a genomic sequence and a set of RNA-Seq reads, reads are mapped to the reference (A) and the resulting alignment is then analyzed by GIIRA. Candidate genes are extracted (B) and ambiguous reads are reassigned using a maximum-flow optimization (C). Finally, candidate genes are evaluated based on the reallocated reads (D).

As depicted in Figure 1, GIIRA is based on an alignment of reads from a RNA-Seq experiment to the DNA sequence of interest. Based on the observed mapping coverage GIIRA identifies candidate genes and searches the most likely start and stop position for each candidate (Fig. 1.B). Several parameters that can be user-defined or estimated from
the alignment control the verification of splicing events or coverage variations. All overlapping reads, regardless if unique or ambiguous mapping, are assigned to their corresponding candidate gene.

Prokaryotic candidates undergo an additional extraction step because prokaryotic operons contain a continuously expressed region including one or more genes that have to be identified respecting the present open reading frames (ORFs). An algorithm formulated as a linear program optimizes the set of chosen ORFs with focus on selecting a set that covers a large number of bases in this operon while restricting the overall number of ORFs.

In the previous steps all read mappings contributed equally to the extraction of candidate regions, even if a read had multiple mappings with similar quality. However, as each sequenced read can only arise from one genomic locus, we have to reassign ambiguously mapped reads to their most likely origin. To do so, GIIRA uses a maximum-flow approach formulated as an integer linear program that is based on the coverage of the gene candidates, their support from unique mappings and the ambiguity of the reads themselves (Fig. 1.C).

The rationale behind this approach is that if several genes compete for the same read, their overall read coverage and the presence of support from unique reads indicates the most likely origin of this read. Both factors do not only enhance the probability for a candidate to be chosen, but also decrease the chances of the competitors such that the number and quality of the competitors directly affects the choice for the best origin.

The problem of assigning each read to exactly one gene candidate can be formulated as a network problem as illustrated in Figure 2. We define a network $G=(N,E)$ with edge set $E$ and node set $N=R \cup C \cup s \cup t$ with nodes $r \in R$ representing reads and nodes $c \in C$ representing gene candidates, respectively. Source node $s$ and target node $t$ are defined for technical reasons. Further, all edges are directed and an edge $e_{ij} \in E$ between two nodes represents that read $r_i \in R$ is assigned to gene $c_j \in C$. Note that each edge has a capacity, which can be understood as the maximal input that can pass through this edge. In contrast, nodes have an unlimited throughput.

The aim of the maximum-flow is to set all capacities $\varphi_{ij}$ (belonging to edges $e_{ij}$ connecting a read $r_i$ to a candidate $c_j$) in a way that the flow passing from source to target node is maximized.

After a unique position for each read has been assigned, the candidate genes are refined accordingly and genes lacking read support are erased (Fig. 1.D). Further, all remaining genes are scored based on their read coverage and the quality and former ambiguity of assigned reads. This allows an easy post-processing to verify genes for follow-up analyses, for instance to filter out genes with overall low support.
We evaluated GIIRA in three simulations and two real datasets, where it performed favorably in comparison to three prokaryotic and eukaryotic gene finders as well as the RNA-Seq based method Cufflinks [TWP+10]. In particular for prokaryotes, GIIRA showed a substantial increase in both sensitivity and specificity compared to existing gene finders. While comparing the methods, we also tested combinations of GIIRA with predictions from other gene finders. Combining the results of different gene finding strategies complemented the single method predictions and resulted in significantly improved prediction accuracy, also in comparison to existing approaches for prediction combination [ZR14].

References


A general approach for discriminative de novo motif discovery from high-throughput data

Jan Grau¹, Stefan Posch¹, Ivo Grosse¹, and Jens Keilwagen²,³

¹Institute of Computer Science, Martin Luther University Halle–Wittenberg, Halle (Saale), Germany
²Julius Kühn-Institut (JKI) - Federal Research Centre for Cultivated Plants, Quedlinburg, Germany
³Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Seeland OT Gatersleben, Germany

{grau|posch|grosse}@informatik.uni-halle.de
jens.keilwagen@jki.bund.de

Abstract: High-throughput techniques like ChIP-seq, ChIP-exo, and protein binding microarrays (PBMs) demand for novel de novo motif discovery approaches that focus on accuracy and runtime on large data sets. While specialized algorithms have been designed for discovering motifs in in-vivo ChIP-seq/ChIP-exo or in in-vitro PBM data, none of these works equally well for all these high-throughput techniques. Here, we present Dimont, a general approach for fast and accurate de-novo motif discovery from high-throughput data, which achieves a competitive performance on both ChIP-seq and PBM data compared to recent approaches specifically designed for either technique. Hence, Dimont allows for investigating differences between in-vitro and in-vivo binding in an unbiased manner using a unified approach. For most transcription factors, Dimont discovers similar motifs from in-vivo and in-vitro data, but we also find notable exceptions. Scrutinizing the benefit of modeling dependencies between binding site positions, we find that more complex motif models often increase prediction performance and, hence, are a worthwhile field of research.

Original paper: doi: 10.1093/nar/gkt831
http://nar.oxfordjournals.org/content/41/21/e197

1 Introduction

Transcription factors are a major component of gene regulation as they bind to specific binding sites in promoters of genes and subsequently activate or repress gene expression. The de novo discovery of transcription factor binding motifs and binding sites from data obtained by wet-lab experiments is still a challenging problem in bioinformatics, and has not been fully solved yet.

Today, two prevalent sources of experimental data are chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-seq and ChIP-exo) and protein binding microarrays (PBMs). Chromatin IP experiments yield approximate genomic regions bound
by transcription factors \textit{in-vivo}, where each of the bound regions is associated with a measure of transcription factors abundance (often termed \textit{peak statistics}) at that specific region. PBM experiments yield information about binding affinity (measured as \textit{probe intensities}) of transcription factors to an unbiased collection of probe sequences \textit{in-vitro}.

2 \hspace{1cm} \textbf{Approach}

We present Dimont [GPGK13], a \textit{de novo} motif discovery approach especially tailored to data from these high-throughput techniques. In contrast to previous approaches, Dimont uses a weighted discriminative learning principle for learning model parameters, which exploits the measures of confidence obtained experimentally, namely peak statistics or probe intensities. This learning principle relies on a common rank-based weighting schema that allows for integrating these different measures of confidence. Strategies for runtime optimization implemented in Dimont result in runtimes of approximately 10 minutes on an average-sized ChIP-seq data set.

3 \hspace{1cm} \textbf{Key findings}

Dimont successfully discovers all motifs of the ChIP-seq data sets of Ma \textit{et al.} [M12]. On the data sets of Weirauch \textit{et al.} [W13], it predicts PBM intensities from probe sequence with higher accuracy than any of the approaches specifically designed for that purpose. Dimont also reports the expected motifs for several ChIP-exo data sets. Hence, Dimont is the first approach that yields a competitive performance on both \textit{in-vitro} and \textit{in-vivo} data using a unified approach.

This allows us to investigate differences between \textit{in-vitro} and \textit{in-vivo} binding in an unbiased manner. We find that for most transcription factors, the motifs discovered by Dimont are in good accordance between techniques. However, we also find notable exceptions, where the motifs obtained from ChIP and PBM experiments for the same transcription factor differ substantially, although both yield accurate predictions on data sets obtained by the corresponding technique.

We use the common framework of Dimont to additionally study the impact of motif models incorporating dependencies between adjacent positions (inhomogeneous Markov models of higher order) compared to standard position weight matrices on prediction accuracy. We find that modeling adjacent dependencies indeed improves prediction accuracy for several transcription factors for both, \textit{in-vitro} and \textit{in-vivo} data. Notably, this improvement often persists for predictions across techniques, i.e., for learning a model from \textit{in-vivo} data and testing it on \textit{in-vitro} data, and vice versa. The latter finding supports that the more complex motif models indeed capture relevant biological information instead of amplifying experimental biases due to the different techniques.
4 Availability

We provide a Dimont web-server at galaxy.informatik.uni-halle.de and a command line application at www.jstacs.de/index.php/Dimont. For installing Dimont into a local Galaxy, the web-application is also available from the Galaxy tool shed [B+14] at toolshed.g2.bx.psu.edu/ view/grau/dimont_motif_discovery. Dimont has been integrated into the open source platform Chipster [K+11] since Chipster v2.11.

5 Talk outline

After a brief introduction into the problem of de novo motif discovery and experimental techniques, the first part of the talk focuses on the specifics of the approach implemented in Dimont. While some of the methods presented in this part are specific to motif discovery, others are also applicable to other fields of bioinformatics as, for instance, the common weighting schema developed for ChIP and PBM data. In the second part, we will present results of the comparison of in-vivo and in-vitro data and of modeling intra-motif dependencies. Finally, we give a brief overview of the Dimont web and command line applications.

References


Interactive and dynamic web-based visual exploration of high dimensional bioimages with real time clustering

Magnus Rathke¹, Jan Kölling¹, Karin Gorzolka², Karsten Niehaus², Tim W. Nattkemper¹

¹Biodata Mining Group, Faculty of Technology
²Proteome and Metabolome Research, Faculty of Biology
Bielefeld University
Universitätsstraße 25
D-33615 Bielefeld
mrathke@cebitec.uni-bielefeld.de
koelling@cebitec.uni-bielefeld.de

Abstract: Web browsers and web applications have become common tools in bioinformatics over the past decades. Many existing web applications revolve around server-client interaction, where heavy computational tasks are often outsourced to the server and the presentation is handled on the client-side. However, more recent additions to the web browser technology embrace the capability of handling more complex operations on the client-side itself, cutting out most of the server-client interaction except for data loading.

This paper contributes to the exploration of the potential of approaches to implement and speed up computational expensive tasks, like image cluster analysis, within a client-side web browser environment. The experimental results, incorporating the well-known \( k \)-means algorithm which serves as a platform for various parallelization approaches, indicate the possibility to achieve real time image clustering. Especially for the available MALDI-MSI data set the results look promising. Despite good results of multithreading approaches, algorithmic approaches appear to be relevant too. Therefore advancements in accelerating the \( k \)-means algorithm itself are considered.

1 Introduction

Imaging systems have become a relevant factor in biological disciplines. Especially sophisticated imaging methods such as Matrix-Assisted Laser Desorption/Ionization Mass Spectrometry Imaging (MALDI-MSI) or Raman imaging, but also the enhancement of traditional optical light microscopic methods through fluorescence microscopy, provide the possibility of visual analysis of biological specimen on a cellular and subcellular level. MALDI-MSI and Raman imaging in particular not only provide high fidelity molecular compound analysis but also acquire regiospecific molecular measurements, which allows a visual representation of tissue biology on the basis of specific molecules, e.g., proteins, metabolites or peptides [NC13].

Although modern imaging techniques are able to give deeper insight into the biological system of an analyzed specimen and greatly contribute to the understanding of biologi-
cal molecules, the evaluation and analysis of such generated data sets can be complex. Multivariate images generated by biological imaging methods usually contain hundreds (multi-spectral) to several thousands (hyper-spectral) images, depending on the complexity of a probed sample and the chosen imaging method [F⁺13].

To provide the required computational resources for the analysis of complex data sets and to make the results easily available to researchers in different settings, modern analysis and visualization methods often involve server-client applications. The server side of an application handles expensive computational tasks for multivariate data analysis, employing either computing clusters or cloud computing solutions to ensure reasonably fast computation. The client side of such applications handles the visualization of the computational results and offers an interface to the analysis methods. The drawback of server-client applications is asynchronous processing, since it is not ensured that a task submitted to a server is computed immediately. This prevents server-client applications from true interactivity, which is desired especially in the context of data mining and exploratory data analysis [L⁺11] [HLN11].

Modern desktop computers are able to use extensive computational resources. Due to technological advancements, multi-core processors as well as powerful graphic cards are common assets. Furthermore, advancements in web browser technology enable them to use the provided resources even from within a standard web browser. Modern web browser implementations provide multithreading capabilities and also allow accelerated 3D graphics support by accessing graphic cards. As a result, client side applications with access to more and more computational capabilities of current desktop computers, which are also easy to access and update, are possible.

In the following, the focus is laid on achieving real-time cluster analysis of spectral image data within a client-side web browser environment. This aims at interactive exploration of high dimensional data sets, with respect to different clustering parameters, e.g., number of prototypes, or varying input, e.g., subsets of the data, for visual display of different clustering results (clustringmaps) [K⁺12]. Therefore, it is investigated how much speedup, if any, can be achieved for k-means using the current multithreading capabilities of web browsers. To better evaluate the effects of multithreading and further accelerate the analysis, several established algorithmic enhancements of k-means were considered in addition to a standard implementation. For convenience, a short introduction into the clustering problem with k-means and a definition of the data structure is given in Section 2. Section 3 introduces two approaches to accelerate the k-means algorithm, whereas the first approach is based on using so called web workers¹ for parallel execution and the second approach refers to the strategy to reduce the amount of distance calculations needed, by applying the triangle inequality [Elk03]. The experimental results of these approaches are presented in Section 4 and discussed in Section 5. Section 6 summarizes the findings and gives an outlook on improvements for parallel computation using future web browser features.

¹http://www.whatwg.org/specs/web-apps/current-work/multipage/workers.html
2 Data- and Problem Definition

This section presents the clustering problem with the $k$-means algorithm on multivariate images. Therefore definitions are given for high dimensional image datasets and the $k$-means algorithm.

2.1 Data definition

![Multivariate image](image)

Figure 1: Multivariate image $X$ with dimensionality $d$. In the context of imaging $d$ is often referred to as the number of *channels*. Each channel represents one image $X_i \in X_1, \ldots, X_d$, with dimensions $x \times y$.

Biological multivariate images generated with, e.g., MALDI-MSI or Raman Imaging combine information on molecular composition and position and thereby enable localization of molecular compounds. The underlying concept is to fixate a biological specimen or a section of one and probe it in a regular order (rasterisation), acquiring a spectrum of information per sampled position. This not only allows the correlation of identified features to a specific location within the sample, but also visualization.

A multivariate image $X$ contains an ordered set of greyscale images $\{X_1, \ldots, X_d\}$ (see Figure 1) where each image $X_i = (x_d)$ is a $x \times y$ matrix of intensity values. Each image $X_i$ visually represents the spatial distribution of a selected feature $i$ in the original sample. Concatenating all intensity values for a fixed $x, y$ coordinate over all images $X_i$ recovers the original spectrum.

2.2 $k$-means

The $k$-means algorithm is one of the most famous unsupervised learning algorithms for cluster analysis [W+08] [Jai10]. Since its publication in the middle of the 20th century, several variations were developed. Lloyd’s algorithm is often referred to as the standard $k$-means algorithm [Llo82].
The goal of $k$-means is to partition a set $X = x_1, x_2, ..., x_n$ into $k$ subsets ($k < N$). Each subset is represented through a prototype $\mu_k$. These subsets are referred to as clusters $C = c_1, c_2, ..., c_k$ with their respective prototypes $\mu_k$. To partition a set $X$ into $k$ subsets the algorithm locates similarities by minimizing $J(C)$ (see equation 1), the sum-of-squared errors, between each datapoint $x_i$ and its corresponding prototype $\mu_k$ of a cluster $c_k$.

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2$$ (1)

To achieve a local minimum $J(C)$, the algorithm cycles through two steps which can be described as:

1. **Find the closest prototype**
2. **Update prototypes**

The runtime of the $k$-means algorithm is thereby primarily influenced by the distance calculations, which are performed to find the closest prototype $\mu_k$ to a datapoint $x_i$. Since this action needs to be performed for each datapoint $x_i$, the runtime is denoted as $\Theta(n \cdot k \cdot d)$, where $d$ depicts the dimensionality of the datapoint and the prototype. The recalculation of the centroids is the second factor in the $k$-means algorithm. The runtime of this step is denoted as $\Theta(n \cdot k \cdot d)$. However these two parts do not increase cardinality. Thus, the overall runtime of the $k$-means algorithm is linear in all its factors and is described as $\Theta(n \cdot k \cdot d \cdot i)$, where $i$ denotes the number of iterations.

### 3 Methods

This section introduces the approaches which were considered to achieve real time clustering in a client-side web browser environment.

#### 3.1 Multithreading in web browsers

Modern web browsers allow JavaScript (JS) web applications to spawn operating system level threads via the **Web Worker**-interface, which enables actual concurrent computing. Applications can thus execute time consuming operations in the background without interfering with user interfaces. Threads invoked by the application execute a JS script, which is called a worker. Each worker operates in its own global scope, thus does not share any resources with other threads. Communication, e.g., messaging or data transfer, between threads is done in a **Message Passing Interface (MPI)** like manner and allows bidirectional messaging between parent and child threads. Transferring data can be done

in either two ways. The first way is structural cloning, where objects can be cloned and copied to other threads. The second way is transferable objects. Incorporating another more recent feature of modern web browsers in the form of typed arrays\(^3\), this technique passes a reference to the buffer of an object to other threads. Generally workers can be separated into two different categories, depending on their way to communicate with other threads [Fla11].

**Dedicated worker** is the standard worker which is implemented by all modern browsers. The dedicated worker can send to and receive from one parent thread, but is not able to send messages sideways. Thus the dedicated worker can’t interact with other worker threads.

**Shared worker** is an implementation which is not communicating via messages alone, but each worker also has specific *ports* it listens to. The shared worker can receive messages from more than one application and allows worker-threads to communicate with each other.

Thus, the actual parallelization scheme becomes similar to approaches using MPI. The idea for dedicated workers is to distribute data evenly between the worker-threads. Speedup is gained, since each thread has to do less calculations:

```plaintext
#Parent thread:
   Distribute data evenly between the child threads.
Repeat until convergence:
  #Child threads:
    Calculate the best matching unit.
    Return indices.
  #Parent thread:
    Calculate new prototypes.
```

### 3.2 Accelerated \(k\)-means through geometric reasoning

A focus of research concerning \(k\)-means is acceleration by using additional information available at runtime through geometric reasoning [PM99]. The central concept is to avoid unnecessary distance calculations by using the triangle inequality. For any three points \(x\), \(y\) and \(z\) the following equation applies:

\[
d(x, z) \leq d(x, y) + d(y, z)
\]  

(2)

Especially in later iterations Lloyd’s algorithm tends to undergo unnecessary distance calculations when cluster centers are almost settled. This is particularly time consuming for higher dimensional datasets. Furthermore, it is not necessary to know the exact distance from a datapoint \(x_i\) to its corresponding center \(c_j\), as long as the triangle inequality holds true.

\(^3\)https://www.khronos.org/registry/typedarray/specs/latest/
Additional information is passed from one iteration to the next iteration in the form of upper bounds and lower bounds. The upper bound $u(x)$ denotes the distance of a datapoint $x_i$ to its closest corresponding center $c_j$. The lower bound $l(x, c')$ denotes the distance of a datapoint $x_i$ to each centroid $c'$, where $c_j \neq c'$ [Elk03].

Obtaining useful lower bounds that help to skip unnecessary distance calculations will not be discussed here, detailed information can be found in C. Elkan’s paper “Using the Triangle Inequality to Accelerate $k$-means” [Elk03]. Elkan proposes to utilize the triangle inequality by computing an upper bound $u(x)$ and keeps track of $k - 1$ lower bounds $l(x, c')$ for each datapoint $x_i$. Although the algorithm achieves massive speedup compared to the standard $k$-means algorithm, it needs additional memory for lower bounds, which is used to keep track of the distances of a datapoint $x_i$ to each centroid $c'$ (where $c' \neq c$ and $c$ denotes the closest center).

Variations of the accelerated $k$-means algorithm proposed by G. Hamerly and Drake & Hamerly’s adaptive $k$-means algorithm show that it is possible to skip most of the distance computations by keeping track of fewer lower bounds. Hamerly’s algorithm keeps track of only one upper bound and one lower bound [Ham10]. The adaptive $k$-means algorithm keeps track of a variable number $b$ of useful lower bounds, which are considered for distance computations. Generally good runtime of the algorithm is achieved for an interval of $\frac{k}{8} \leq b \leq \frac{k}{4}$ [DH12]. Thus, both variations have a much lower memory profile but exceed Elkan’s algorithm for lower (Hamerly’s algorithm) to mid range (adaptive $k$-means) dimensional datasets [DH12].

Adaptive $k$-means is used here as an algorithmic approach to faster execution and was implemented in JavaScript. Additionally, a threaded version using web workers was developed.

4 Experimental Results

This section presents the observations made during the experiments with the different $k$-means implementations. All results were produced with an Intel Core i7-3632 QM @ 2.20 ×4 CPU and 8GB Memory. As web browsers Mozilla Firefox version 27 and Google Chrome version 32 were used. The experimental setup primarily considers multithreading approaches with dedicated workers and algorithmic enhancements using adaptive $k$-means.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard-kmeans</td>
<td>Standard $k$-means implementation</td>
</tr>
<tr>
<td>worker-kmeans</td>
<td>$standard-kmeans$ executed from a web worker script</td>
</tr>
<tr>
<td>threaded-kmeans</td>
<td>Threaded $k$-means implementation employing web workers</td>
</tr>
<tr>
<td>trans-thread-kmeans</td>
<td>$threaded-kmeans$ modified to use transferable objects</td>
</tr>
<tr>
<td>adapt-kmeans</td>
<td>$k$-means implementation based on the adaptive $k$-means algorithm</td>
</tr>
<tr>
<td>thread-adapt-kmeans</td>
<td>Threaded version of $adapt-kmeans$</td>
</tr>
</tbody>
</table>
Figure 2: Sample images for three different datasets: 1. A 500 × 500 multivariate image of random intensity images ranging from 0 to 255 and dimensions of \( d = 30 \) and \( d = 60 \). 2. A 300 × 300 tiled constructed multivariate image containing 9 clusters plus background and dimensions of \( d = 30 \) and \( d = 60 \). The tiles vary their intensities between \(-5\) and \(+5\) of their average. 3. A 120 × 50 MALDI-MSI dataset from a study on barley seed germination [Gor13] with dimensions of \( d = 54 \) and \( d = 94 \).

In the following, the different versions will be referenced to as shown in Table 1. The algorithms are considered to reach convergence if 90% of the centroids do not differ in the distance of an \( \epsilon \)-environment of \( 1 \cdot 10^{-7} \) to the centroids of the previous iteration. If no convergence occurs, the process is terminated after 1000 iterations. For the experiments, three different datasets were employed to verify performance in terms of runtime (Figure 2.1), accuracy (Figure 2.2) and behavior on real MALDI-MSI data (Figure 2.3). The runtime of the different implementations listed in Table 1 depends on the web browser used. Overall, Chrome achieved better execution times than Firefox for the single threaded versions, i.e., standard-kmeans and adapt-kmeans. In particular, the standard-kmeans method showed the best results for runtime on the datasets using Chrome, but good runtime results can also be observed using Firefox (see Figure 3). The second single threaded \( k \)-means version, i.e., adapt-kmeans, showed high variance in its runtime performance. The obtained runtime results for the Firefox browser show that the algorithmic enhancement achieved slightly better performance for the constructed and the MALDI-MSI dataset (see Figure 3(c) and (e)), however, particularly for the Chrome browser the adapt-kmeans algorithm performs worse than standard-kmeans (see right column of Figure 3).

Firefox performs consistently better for threaded \( k \)-means versions such as threaded-kmeans, trans-thread-kmeans and adapt-thread-kmeans (see Figure 3). In the case of the threaded version of the adaptive \( k \)-means, the runtime experiments had to be terminated. Due to extensive memory usage, the test runs could not be finished (see Figure 3(a) and (b)). Also, the threaded \( k \)-means method using transferable objects, i.e., trans-thread-kmeans, was outperformed by the version using structural cloning, i.e., threaded-kmeans, in most cases (see Figure 3). The threaded-kmeans method achieves runtime performance comparable to standard-kmeans (see Figure 3(c) and (e)) and also exceeds the runtime performance of the single threaded standard-kmeans version (see Figure 3(a)). In addition, it could be observed that the single threaded \( k \)-means version pushed into a background thread, i.e., worker-kmeans, overall performed worse than the standard-kmeans version (see Figure 3).
(a) Random image dataset with Firefox

(b) Random image dataset with Chrome

(c) Constructed image dataset with Firefox

(d) Constructed image dataset with Chrome

(e) MALDI-MSI dataset with Firefox

(f) MALDI-MSI dataset with Chrome

Figure 3: Runtime results of the employed datasets for the Firefox browser (left column) and the Chrome browser (right column). Figures (a) and (b) Runtime results of the employed $k$-means versions for $k = 15, 50$ and $100$ on the random image dataset. Figures (c) and (d) Runtime results of the employed $k$-means versions for $d = 30$ and $60$ on the constructed image dataset. Figures (e) and (f) Runtime results of the employed $k$-means versions for $d = 54$ and $94$ on the MALDI-MSI image dataset.
5 Discussion

On the one hand, the results show that both Firefox and Chrome are able to perform complex tasks such as clustering multi-spectral image data in a client-side environment. The standard single threaded $k$-means version overall shows good performance on any of the datasets and delivers reasonable fast results even for huge datasets (see Figure 1.1 and Figure 3). On the other hand, the runtime experiments show unexpected variability in the performance of the algorithmic enhancement approaches and the threaded versions of the $k$-means algorithm. One possible reason for the adaptive $k$-means methods to not deliver the expected performance boost might be that the termination criterion does not fit and other criteria might show better results. The high memory usage observed for the threaded version of the adaptive $k$-means can partly be explained by the characteristics of web workers, since a web worker has high memory usage.

For the threaded approaches in general, the overhead for worker start-up and communication between the threads likely predominates the effects from parallel execution and no speedup can be measured. Another unexpected observation was made regarding the use of transferable objects and structural cloning. Transferable objects are a reference that is passed to a worker-thread. As a consequence the communication delay becomes smaller and a one to two orders of magnitude faster data transfer rate, than with structural cloning, can be achieved. Despite faster transfer rates the trans-thread-kmeans method overall took longer to finish, which is likely caused by the usage of native arrays and thus achieving higher precision in the distance computation and recalculation of the $k$-means prototypes.

6 Conclusion and Outlook

With respect to the initially formulated question, if it is possible to achieve online real time clustering of high dimensional datasets within a client-side web browser environment, the experimental results show that this goal could be successfully achieved for multi-spectral image datasets. The second question, whether or not it is feasible to accelerate heavy computational tasks, like clustering, with either computational approaches or algorithmic enhancements needs to be investigated further. At least for the current browser versions, trying to accelerate $k$-means with mutithreading or by using the adaptive version actually performs slower than the standard implementation.

Regardless of problems concerning the multithreaded versions of the $k$-means algorithm, the runtime experiment results show, that online real time clustering of multi-spectral datasets is possible. This enables new visualizations, e.g., a user can cluster a hyperspectral image, de-select single channels of it and get a new clustering result directly. Thereby, interactive exploration of datasets with the help of cluster analysis becomes possible.

Both web browsers achieve good performance and respond with a new clustering result in less than three seconds for either of the datasets and with all channels selected. It remains to be seen if the threading approaches get more potential with future features for web browsers. With broader support across all browsers the shared worker API might be a
possible candidate for concurrency computing models in a web browser environment. Up until now, single threaded approaches seem to be superior to concurrency models. With later iterations of the Web Worker interface and optimized approaches for concurrency computing in a web browser environment, client-side web browser solutions can become a viable addition to current approaches for the analysis of high dimensional image datasets.

Acknowledgments:
This work was supported by the DFG Großgeräteinitiative “Bildgebende Massenspektrometrie” and the CLIB Graduate Cluster Industrial Biotechnology.

References


Efficient Large-scale Bicluster Editing

Peng Sun\textsuperscript{1,2}, Jan Baumbach\textsuperscript{1,3} and Jiong Guo\textsuperscript{2}

\textsuperscript{1}Max Planck Institute for Informatics. Campus E1 4, 66123 Saarbrücken, Germany
\textsuperscript{2}Cluster of Excellence for Multimodel Computing and Interaction. Campus E1 7, Saarland University, 66123 Saarbrücken, Germany
\textsuperscript{3}Institute for Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark

psun@mpi-inf.mpg.de

Abstract: The explosion of the biological data has dramatically reformed today’s biological research. The need to integrate and analyze high-dimensional biological data on a large scale is driving the development of novel bioinformatics approaches. Biclustering, also known as simultaneous clustering or co-clustering, has been successfully utilized to discover local patterns in gene expression data and similar biomedical data types. Here, we contribute a new approach: Bi-Force. It is based on the weighted bicluster editing model, to perform biclustering on arbitrary sets of biological entities, given any kind of similarity function. We first evaluated the power of Bi-Force to solve dedicated bicluster editing problems by comparing Bi-Force with two existing algorithms in the BiCluE software package. We then followed a biclustering evaluation protocol from a recent review paper from Eren et al. and compared Bi-Force against eight existing tools: FABIA, QUBIC, Cheng and Church, Plaid, Bimax, Spectral, xMOTIFS and ISA. To this end, a suite of synthetic data sets as well as nine large gene expression data sets from Gene Expression Omnibus were analyzed. All resulting biclusters were subsequently investigated by Gene Ontology enrichment analysis to evaluate their biological relevance. The distinct theoretical foundation of Bi-Force (bicluster editing) is more powerful than strict biclustering. We thus outperformed existing tools with Bi-Force at least when following the evaluation protocols from Eren et al. Bi-Force is implemented in Java and integrated into the open source software package of BiCluE. The software as well as all used data sets are publicly available at http://biclue.mpi-inf.mpg.de.

1 Introduction

Clustering is commonly accepted as a powerful approach to explore gene expression data sets [MAN+11]. Given a pairwise similarity function transformed into a similarity matrix, clustering algorithms seek to partition the data items into a list of disjoint groups, such that the similarities within each group are maximized and those between different groups are minimized. Traditional clustering approaches cluster only rows or columns in one run, which is not always beneficial [TSS02]. In contrast, biclustering allows to simultaneously partition both, rows and columns. If we are given, for instance, gene expression data sets for different cellular conditions, biclustering is more powerful in capturing biologically meaningful subsets of condition-specific genes. The major reason is that the expression
of gene subsets may correlated only under some conditions while being independent under other conditions. Biclustering approaches are generally capable of discovering such local patterns. They have proven particularly useful in various types of gene expression data analyses [GMO09] but should also work on other omics data sets, proteomics or metabolomics.

The first such “biclustering” tool was developed by Cheng and Church and applied to gene expression data [CC00]. Since then, many other biclustering tools have been reported (e.g. [MO04, BPP08, FBP10, BBP+06, CLSL07]) and been suggested for applications to various biomedical problems [SGB12, HPC+10]. Biclustering tools became increasingly popular due to their ability to simultaneously cluster biological data from different sources in order to discover local bi-correlations patterns. Several systematic comparisons have been published, using various measurements to evaluate a number of prevailing biclustering tools on both synthetic and real-world data sets [TBK05, PBZ+06, EDKÇ13].

Here, we present a software implementing a novel heuristic algorithm that efficiently solves the biclustering problem: Bi-Force [SSR+14]. It comes as an extension of the BiCluE software package. The Bi-Force extension is dedicated to solve the large-scale problem instances that we face in nowadays bioinformatics more and more frequently. Bi-Force was compared to eight existing biclustering software implementations on artificial and real biological data sets.

2 METHODS

The main methodological contribution of this paper is an algorithm that heuristically solves the weighted bicluster editing problem. Bi-Force is motivated by the well-known physics-inspired graph layout algorithm of Fruchterman and Reingold [FR91]. It mainly seeks to arrange all nodes of a graph in a two-dimensional plane such that “similar” nodes are located more close to each other than to others. Bi-Force, afterwards, assigned the nodes from each “dense” part of the graph layout to one bicluster by single linkage clustering or k-means clustering based on the Euclidean distances. The algorithm is carried out in a three-step procedure: (a) layout generation, (b) bicluster partitioning, and (c) post-processing.

2.1 Layout generation

In this stage, the coordinates of all nodes are generated and re-arranged in a way that the nodes with higher similarities are located next to each other, and far away from those that are dissimilar. Bi-Force computes pairwisely the “physical forces” between two nodes, i.e., the magnitudes that similar nodes attract each other, dragging them closer while dissimilar nodes repulse each other, pushing them farther away. The whole algorithm starts with an initial layout where nodes are “almost” evenly located on a two-dimensional circle with randomly permuted order. The radius $R$ of the circle is a parameter of Bi-Force. The
strength of attracting/repulsing force depends on the current positions of the two nodes, attraction/repulsion coefficient and the corresponding cost to delete the edge or to insert the missing edge between the two nodes. The re-arrangement is performed in an iterative manner. In each round, the movement of each node is the cumulative effect of the attractions and repulsions from all other nodes. Afterwards, all nodes are re-positioned to the new locations simultaneously according to the magnitudes of the movements. The whole procedure is repeated for $I$ times. The attracting/repulsing effect from node $v$ to $u$ is computed by the following formula:

$$f_{u \leftarrow v} = \begin{cases} \frac{\text{cost}(uv) \cdot f_{\text{att}} \cdot \log(d(u,v) + 1)}{|V|} & \text{for attraction} \\ \frac{\text{cost}(uv) \cdot f_{\text{rep}}}{|V| \cdot \log(d(u,v) + 1)} & \text{for repulsion} \end{cases}$$

In the formula above, $f_{u \leftarrow v}$ represents the attracting/repulsing effect from node $v$ to $u$, i.e., the magnitude of the movement of $u$ caused by $v$. When there is an edge between $u$ and $v$, they attract each other and if otherwise, they repulse. $f_{\text{att}}$ and $f_{\text{rep}}$ are the attractive and repulsive factors, respectively. $d(u,v)$ represents the Euclidean distance between node $u$ and $v$. Obviously, the threshold $t_0$ affects the density/granularity of the bicluster editing model: the smaller $t_0$ is, the fewer biclusters there are and the larger of their sizes, and vice versa.

To accelerate the convergence of the nodes to stable positions, a cooling parameter is used to limit the maximal magnitudes of attractions and repulsions. This means in a certain iteration $i$, the movement magnitude cannot exceed the current cooling parameter $M_i$. Cooling parameter starts with an initial value $M_0$ as a parameter in Bi-Force and decreases with every iteration.

At the end of this stage, the positions of all nodes are fixed and similar nodes should be close to each other. In the next step, we make use of this assumption and partition the layouted graph in a way that optimizes the editing costs.

### 2.2 Bicluster partitioning

Based on the coordinates of the nodes obtained in the previous stage, we partition the graph into disjoint biclusters using two different geometric clustering methods: single-linkage clustering (SLC) and $k$-means. Both, SLC and $k$-means are standard methods in computational cluster analysis [WRR+11]. The density parameters of the two algorithms (distance threshold $\delta$ for SLC and the number of clusters $k$ for $k$-means) are varied systematically (SLC: $\delta = 0 \ldots M_0 + R$ in steps of $\sigma$, $k$-means: $k = 2 \ldots |V|/3$). For each clustering result we compute the necessary editing costs to create this solution. Finally, we keep the solution that has minimal editing costs before we proceed to post-processing.
2.3 Post-processing

Here, we try to further reduce the clustering costs, which includes two steps: (a) Biclusters merging and (b) nodes moving.

To reduce the number of redundant biclusters, particularly the singletons, we try to merge biclusters. First, all the biclusters are ordered by size in an ascending order. Let $B = (b_1, b_2, ..., b_l)$ be the $l$ ordered biclusters, where $|b_i| \leq |b_j|$, for all $i \leq j$. For all pairs of biclusters $b_i$ and $b_j$ with $1 \leq i < j \leq l$, we calculate the cost that would emerge from merging the two, i.e., $\text{cost}(b_1, b_2, ..., b_i \cup b_j, ..., b_l)$. Once a $B'$ with a lower overall cost than before is found, we re-define the biclusters according to $B'$ by merging $b_i$ and $b_j$. This step is repeated until no beneficial merging can be done anymore.

After merging the clusters, another post-processing step similar to Restricted Neighborhood Search Clustering [KPJ04] is carried out. Let $B = (b_1, b_2, ..., b_l')$ be the biclusters after the merging step, for each $b_i$ and $b_j$, such that $1 \leq i < j \leq l'$, we compute the costs that would emerge from moving $v \in b_i$ to $b_j$. If the overall cost can thereby be reduced in this step, $v$ is moved to $b_j$. Similarly, this step is repeated until no vertex move is beneficial.

The details of parameter training of Bi-Force could be found in [SSR+14].

2.4 Analysis

The worst-case running time of Bi-Force is dependent on the three steps mentioned above. Let $n = |U| + |V|$ for an input graph $G = (U, V, E)$. In the “layout generation” step, where Bi-Force arranges the positions of all nodes, it consumes $\mathcal{O}(n^2)$ time to compute the mutual attracting/repulsing forces in each iteration. Thus the layout generation step finishes in $\mathcal{O}(I \cdot n^2)$, where $I$ is the number of iterations. The single-linkage clustering runs in $\mathcal{O}(D_1 \cdot n^2)$, where $D_1$ is the number of different thresholds used. The k-means problem is by its nature NP-hard [ADHP09, MNV09]. However, we limited the maximal number of iterations in k-means to be 200 and thus it finishes in $\mathcal{O}(200 \cdot n)$ time. Finally for post-processing, each iteration takes $\mathcal{O}(n^2)$ time and the total running time is bounded by $\mathcal{O}(D_2 \cdot n^2)$ for $D_2$ iterations. Since $D_2$ might increase with $n$, we added an empirical limit of 500 iterations to $D_2$. In most cases Bi-Force did not reach this limit and we observed only small numbers of iterations before it terminated.

In summary, the overall running time for Bi-Force grows quadratic in the number of nodes.

3 RESULTS

We applied all nine algorithms including Bi-Force and eight other prevalent biclustering tools mentioned above to real-world biological data: gene expression microarray data from the GEO database. Their performance was evaluated by means of GO Term Enrichment.
Figure 1: Proportions of GO-enriched biclusters for different biclustering tools on five significance levels.

Figure 1 gives the proportions for different significance levels of the biclusters found by all algorithms.

4 SUMMARY

We have presented Bi-Force and demonstrated its flexibility by applying it to biclustering, a restricted version of bicluster editing with many applications in gene expression data analysis. We compared it to eight existing tools by following an established evaluation protocol from Eren et. al.’s review paper. We show that Bi-Force outperformed the existing tools on synthetic data sets and on real-world gene expression data. Last but not the least,
we wish to emphasize that Bi-Force has the ability to perform simultaneous clustering of arbitrary multiple data sets. It is not restricted to gene expression scenarios. Instead any types of biological data that can be modeled as bipartite graph can be partitioned by using Bi-Force. It is now part of the BiCluE software package and publicly available at http://biclue.mpi-inf.mpg.de.

References


Flexible database-assisted graphical representation of metabolic networks for model comparison and the display of experimental data

Jana Tillack, Melanie Bende, Michael Rother, Maurice Scheer, Susanne Ulas, Dietmar Schomburg

Department of Bioinformatics and Biochemistry
Institute for Biochemistry, Biotechnology and Bioinformatics
Technische Universität Carolo-Wilhelmina zu Braunschweig
Langer Kamp 19B
38106 Braunschweig, Germany
j.tillack@tu-braunschweig.de
d.schomburg@tu-braunschweig.de

Abstract: Intracellular processes in living organisms are described by metabolic models. A visualization of metabolic models assists interpretation of data or analyzing results. We introduce the visualization tool DaViMM creating personalized graphical representations of metabolic networks for model comparison or the display of measurements or analyzing results. The tool is coupled to a relational database containing graphical network properties like coordinates, which ensure an intuitive network layout. A combination of DaViMM, the graphical database, and available biochemical databases enables an automated creation of metabolic network maps. The flexibility of this combination is demonstrated with some application examples.

1 Introduction

A genome-scale metabolic model comprises all information known about the metabolic interactions in an organism of interest [FHT’09]. Metabolic network maps are the graphical representation of metabolic models and support interpretation of data or analyzing results. In network maps, each substance and each reaction is visualized by a node and these nodes are connected via edges related to their metabolic role. In other words, all substances involved in a reaction as substrates or products are connected with this reaction by edges (Figure 1).

The benefits of the graphical representation of metabolic networks are obvious, but the process of its creation is often complex. Manual drawing of metabolic networks is time-consuming and user-dependent, but the networks are usually easy to handle and intuitive. Automatic drawing is fast and reproducible, but the resulting layout is often unstructured regarding biology and thus non-intuitive.
Different approaches try to combine the benefits of manual and automatic creation of metabolic networks [CBC⁺12]. Here, we present an automatic creation of metabolic network maps supported by a database containing positioning information in terms of coordinates for each node. The graphical information combined with biological information from databases like BRENDA (Braunschweig Enzyme Database, [SCP⁺12]) and BKM-react [LSS11] draw a complete and easy to handle metabolic network map for diverse visualization purposes.

The tool we are presenting produces personalized metabolic network maps based on a predefined database. The generated maps have a defined and intuitive layout and can be built in less than one minute. The tool is flexible and allows creating an effective visualization of genome-scale metabolic models, experimental data, or analyzing results in the context of metabolic networks.

2 Information Visualization in the Context of Metabolic Networks

2.1 Database - Basis for Visualization

To speed up the generation of metabolic network maps, all information needed for reproducible visualization is stored in a predefined database. This core database assisting the drawing of metabolic network maps has been developed in three main steps:

1. Manual creation of a genome-scale metabolic network map to get positioning, identity, and role of each node. This basic network has been created in Cytoscape [SMO⁺03] and the following information is defined manually within this network:
   - **Positioning information:** coordinates
   - **Identity:** ligand ID, EC number
   - **Role:** enzyme/reaction, substrate/product, transporter

2. Storage of the graphical information in a MySQL database.

3. Addition of biochemical information to the database. This information is taken from BRENDA and BKM-react and based on the defined identifier. For the different node types the following information is added:
**Enzyme:** EC number, recommended name, enzyme to reaction allocation, enzyme to node allocation, enzyme to organism allocation

**Reaction:** reaction to pathway allocation

**Substance:** BRENDA ligand ID, recommended name, short name, synonyms, substance to reaction allocation with stoichiometric information

The resulting database contains all graphical and biochemical information for 1882 reactions of the metabolism and allows fast and reproducible map generation. The database content will continuously be extended.

This publication will not deal with the database generation in detail, but using the information stored in the database to build metabolic network maps for various applications.

### 2.2 Visualizing Metabolic Networks

Based on the introduced database personalized metabolic network can be visualized. For this purpose, the tool DaViMM has been developed in the programming language Python. The general workflow of network generation is presented in Figure 2.

![Figure 2: Workflow of metabolic network generation.](image)

First, the information included in the database about nodes and edges is imported. From that point on, the nodes and edges can be written into a file in vector image file format (svg) composing a metabolic network.

![Figure 3: Network layout modification: A - monochrome network, B - adapted edge width, C - node colors depending on the node kind, D - node colors depending on data.](image)
Additionally, the visual properties of nodes and edges can be changed before export via import of a data file or definition of the layout in prepared configuration files (Figure 3).

The opportunity of network layout modification allows to personalize the networks for presentations or publications and to visualize various kinds of data in the context of a network. The following sections will give an insight in how flexible the metabolic network maps can be applied.

2.3 Metabolic Network Maps integrated into BRENDA

One application for the network visualization is the database BRENDA. BRENDA contains information about enzymes and ligands involved in enzymatic conversions.

An integration of metabolic network maps into the BRENDA website enhanced the classification of e.g. an enzyme. The network maps characterize the surrounding of an enzyme or a substance of interest, they allocate enzymes to pathways via their location, they allocate enzymes to organisms via highlighting of nodes, and they allow visualizing taxonomic information for an organism using multiple colors.

The basic metabolic network maps generated for BRENDA are an overview map containing one pseudo node for each pathway and pathway maps containing reactions and substances. The pathways in the overview map are classified in four main groups - carbohydrate and lipid metabolism, nucleotide and amino acid metabolism, energy metabolism, and secondary metabolism (Figure 4). The pathway names are always shown on the map or will appear on mouseover depending on the size of the pathways.

Figure 4: Overview map with classified pathways. The size of each pseudo node depends on the number of nodes located in the pathway.

Starting with this overview map, all single pathways can be selected. In each pathway the substances and enzymes are linked to the BRENDA website via the node identifier. Due to the complex interactions in a metabolic network, edges may combine nodes...
located in different pathways. For pathway visualization external substance nodes linked to internal enzyme nodes will be shifted to the pathway boundary (Figure 5).

![Pathway map with highlighted enzyme.](image)

**2.4 Data Visualization in the Context of Metabolic Networks**

Data visualization for publication or presentation is often hand-made. To support and speed up the visualization process the network maps, always known from the sections before, can be modified to visualize data in the context of a metabolic network.

To change network properties depending on information stored in data files, different methods have been developed and are presented in detail.

**Highlight of nodes of a list:** The first method simply highlights all nodes included in the input file by filling the node with one defined color. Therefore, the name used in the input file will be compared with the synonyms stored for each element of the metabolic pathway.

**Highlight of models:** A second method highlights all reaction nodes included in a model in text format. Additionally, models can directly be compared within one map with another model or with database knowledge. Database knowledge means all enzymes known for an organism or the taxonomic information of an organism from BRENDA. In contrast to the method before, matching takes place based on all substrates and products involved in a reaction.
Visualization of data: Most of the other methods assign measurement values to the nodes. To use these methods, the data files should contain pairs of node name and measurement value. The nodes will be filled with a color depending on the measurement value and the defined colors. Two different types of data visualization will be distinguished.

1. If data only include positive values, the nodes will be colored using one gradient. An example for this kind of visualization is the display of metabolite concentrations e.g. from a metabolome analysis (Figure 6A).
2. For coloring of nodes with data, which can be negative as well as positive, two gradients will be combined. This method will be applied to visualize fold changes, e.g. for gene expression or enzyme activity (Figure 6B).

Visualization of flux rates: The last method assigns analyzing results to the edges by adapting the width of each edge based on e.g. results of flux balance or $^{13}$C flux analysis (Figure 6C).

2.5 Network Maps Supporting Modeling Projects

All different visualization applications introduced so far can be combined to support different steps of metabolic modeling. A detailed description of the steps occurring during genome-scale stoichiometric modeling can be found in [TP10]. In the following selected
steps will be discussed regarding their potential to be supported by metabolic network maps.

**Model generation:** An overview of all known enzymes for the organism of interest is a good starting point of the modeling procedure. This information is available in databases and can be visualized using the introduced methods. In addition to the organism of interest itself, visualization of members of its taxonomic tree will give information about enzymes active in similar organisms which might also be relevant for the organism of interest (Figure 7).

![Figure 7: Taxonomic information from BRENDA for the organism of interest. Different taxonomic levels are visualized by the colors of pseudo nodes and nodes.](image)

Besides database knowledge, measurements are needed for model generation. All kinds of omics data can be visualized in the context of a metabolic network map compared to Figure 6.

**Model verification:** Nearly all of the introduced visualization methods can support model verification. The first approach is to highlight all enzymatic reactions included in the model in a metabolic network map, e.g. to find gaps. Furthermore, the model reactions can be compared with database information (Figure 8), all reactions of another model, or measurements (Figure 6).

Finally, the model is used to generate hypotheses about the organisms’ behaviour. Therefore, mathematical algorithms, e.g. Flux-Balance Analysis, are applied to the network. The analyzing results in terms of flux ratios can also be visualized in the network context (Figure 6C).

### 3 Conclusion

The visualization of metabolic networks helps to understand the complex interactions in an organism. Taking the network maps as basis for the visualization of measurements or analyzing results supports data interpretation.
Figure 8: Comparison of the model and taxonomic information from BRENDA for the modeled organism. Different taxonomic levels (blue) and the model reactions (grey) are visualized by the colors of pseudo nodes and nodes.

The challenge in metabolic network visualization is to find the balance between time consuming manual creation and an intuitive network layout. The introduced tools and databases allow building network maps automatically. The layout is defined, but the appearance can be modified individually. For this reason, personalized and colorful network maps for presentations and publications as well as for integration into websites can be built with less effort and time.

By being programmed in Python, the visualization tools run on all common operating systems. The source code is available on request.

References


Characterizing metagenomic novelty with unexplained protein domain hits

Thomas Lingner, Peter Meinicke

Department of Bioinformatics
University of Göttingen
Goldschmidtstr. 1
37077 Göttingen
thomas@gobics.de
peter@gobics.de

Abstract: In metagenomics, the discovery of functional novelty has always been pursued in a gene-centered manner. In that way, sequence-based analysis has been restricted to particular features and to a sufficient length of the sequences. We propose a statistical approach that is independent from the identification of single sequences but rather yields an overall characterization of a metagenome. Our method is based on the analysis of significant differences between the functional profile of a metagenome and its reconstruction from a combination of genomic profiles using the Taxy-Pro mixture model. Here, protein families with a large proportion of domain hits that cannot be explained by the model are interesting candidates for the exploration of metagenomic novelty. The results of three case studies indicate that our method is able to characterize metagenomic novelty in terms of the protein families that significantly contribute to unexplained domain counts. We found a good correspondence between our predictions and the discoveries in the original studies as well as specific indicators of functional novelty that have not yet been described.

1 Background

Metagenomics rigorously extends the exploration of microbial life beyond the borders of culturable organisms. Therefore the vast amount of metagenomic sequence data potentially provides a gold mine for the discovery of novel genes. Several approaches have been proposed for gene mining on metagenomic data. The corresponding methods aim at the identification of candidate sequences using either gene neighbourhood context [HSD+07], protein domain architecture [MCR10], clustering of open reading frames [YSR+07] or phylogenetic trees of known protein families [KTZ+07]. The different methods have specific strengths and limitations, and they can be combined to identify interesting candidate sequences for further analysis [SDL+09]. However, none of these methods is capable to characterize the novelty of metagenomic data beyond the identification of a number of sequences that can be detected according to the above-mentioned criteria.

We here present an approach for the characterization of metagenomic novelty based on
the Taxy-Pro mixture model [KALM13]. Taxy-Pro performs a reconstruction of the functional profile of a metagenome using a linear combination of genomic reference profiles of known organisms. The reconstruction error in terms of the fraction of protein domain hits unexplained (FDU) can be analyzed with respect to the most contributing domain families. We propose that all domain families that are overrepresented in the metagenome when compared to its genomic reconstruction are candidates for the characterization of functional novelty. In contrast to all previous methods, our approach does not detect single candidate sequences but rather predicts the functional categories that contribute to metagenomic novelty. Therefore our method can be applied prior to the identification of candidate sequences to narrow down the search space for gene mining. Furthermore, for the first time we are able to identify and select metagenomes that are most promising in terms of the predicted novelty.

We conducted a number of case studies on real-world metagenomic datasets to evaluate the potential of our approach. The results indicate that our method identifies protein domain-specific functional novelty that is known or suspected with respect to particular environments and also show how hypotheses on novel genes can be obtained from large metagenomic dataset collections.

2 Materials and methods

Our approach to characterization of novelty in metagenomes is based on the statistical analysis of overrepresented Pfam domain families in a metagenome’s functional profile as compared to a reconstruction from genome profiles by a mixture model. We analyze our method in terms of case studies for functional description of novelty in real-world metagenomic datasets from different environments. In the following, we will describe the datasets and methods that we used for our case studies.

2.1 Metagenomic datasets

From the 'Cow rumen' metagenome collection [HSE+11] we selected the largest sample (EBI accession SRR094415, downloaded from www.ebi.ac.uk/metagenomics/sample/ as of April 2014), which contains 11,334,156 Illumina GAIIx paired-end reads of 2 x 125 bp average length (2.8 Gbp total).

The 'Sediment' dataset ([BWTH11], EBI acc. SRS004796) has been collected from marine sediments at the Brazos-Trinity Basin in the western Gulf of Mexico and contained after quality control 402,793 reads with \( \approx 264 \) bp length (106.4 Mbp total). The sediment metagenome was sequenced using the Roche 454 GS FLX technology.

The Human Microbiome Project (HMP, [PGG+09]) provides an extensive collection of samples from human body sites of healthy individuals for large-scale comparative studies. For our evaluation, we used 750 data samples of the HMP as described in detail in [KALM13]. Briefly, the samples have been taken from five major human body sites (‘Uro-
genital tract’, ‘Oral’, ‘Gastrointestinal tract’, ‘Skin’, ‘Airways’) that encompass up to nine subsites. Sequencing was performed using the Illumina HiSeq 2000 platform resulting in paired-end reads of about 2 x 100 bp average length.

2.2 Taxonomic profiling

The Taxy-Pro approach to taxonomic profiling of metagenomes is based on mixture models of functional profiles in terms of protein domain frequencies and is described in detail in [KALM13]. Briefly, a metagenome’s profile \( y \) is estimated from Pfam domain counts and then reconstructed by a linear combination of genomic profiles \( x_1, \ldots, x_N \) from a reference database containing several thousand domain signatures of microbial genomes according to \( \hat{y} = \sum_i^N w_i x_i \). The coefficients \( w_i \) of the linear combination then represent estimated contributions of particular organisms to the explanation of the metagenomes’ signature. The estimation of the coefficients can in principle be implemented as a linear least-squares regression problem. However, due to the unit-sum and positivity contraints on the weights one has to resort to quadratic programming for a solution. Furthermore, because many of the protein domain frequencies are estimated from small counts the errors can be far from normal deviates making a squared-error criterion less adequate. For these reasons we use the classical EM-optimization in case of protein domain features [KALM13]. As a unique feature of the Taxy-Pro method, the approximation error \( \frac{1}{2} \sum_j^D |y_j - \hat{y}_j| \) of the mixture model provides a quality measure in terms of the fraction of domains unexplained (FDU), i.e. the entirety of the \( D \) domain-specific deviations of the metagenome’s actual protein domain profile from its reconstruction. In this way, the FDU can reflect metagenomic novelty in terms of a lack of explanation according to the discrepancy between the observed domain profile and the mixture model.

To obtain the domain signatures we used the ultrafast protein classification (UProC) tool (uproc.gobics.de) in combination with version 27 of the Pfam protein family database [FBC14]. The domain detection significance threshold was left at the default value (0.1% FPR). Depending on the average read length of the metagenomic dataset we applied the default ORF mode (length >200 bp) or the short read mode of UProC.

2.3 Unexplained domain counts from overrepresentation analysis

Our model is based on the assumption that the protein domain frequencies in a metagenome that cannot be well explained by genomic reference profiles are likely to be indicators of metagenomic novelty. This novelty, in general, results from unknown organisms in the microbial community that are missing in the database. In particular, genomes with unusual protein domain signatures that have no evolutionary close neighbors among the references will contribute to the deviations between the observed domain frequencies and their model-based approximation.

The domain-specific deviations of the mixture model \( d_j = y_j - \hat{y}_j \) can be used to identify
over- \((d_j > 0)\) or underrepresented \((d_j < 0)\) domain families in the actual metagenome’s functional profile as compared to its reconstruction. We here focus on the overrepresented families that correspond to domain counts that are significantly higher than the reconstructed counts of the model. To identify significantly overrepresented domain families, we analyzed the difference of domain frequencies for each family using a binomial test. Here, the relative domain frequencies \(\hat{y}_j\) of the reconstructed profile are used as estimators of the event probabilities and the total number of hits \(n\) to Pfam domains in the metagenome are used as the number of draws in terms of Bernoulli trials. In order to avoid the singular case of zero event probabilities that would always result in significant overrepresentation, we used a unit pseudocount value for the estimation of probabilities. The value \(P(c_j|n, \hat{y}_j)\) of the cumulative binomial distribution function calculated for the actual domain count value \(c_j\) then corresponds to the probability of observing up to \(c_j\) counts in \(n\) hits. The value \(1 - P\) corresponds to the probability (p-value) of \(y_j\) being significantly larger as compared to \(\hat{y}_j\). To account for multiple testing with \(D > 14,000\) Pfam families, we calculated p-values using the family-wise error rate (FWER) correction method [Hol79]. For selection of significantly overrepresented domains we used a p-value threshold of 0.05.

For our evaluation we use two domain-specific novelty indices: firstly, the number of expected unexplained domain counts (EUC), which we define as the rounded product of the domain-specific deviation \(d_j\) and the total number of domain counts \(n\) in the dataset. Furthermore, we use the original count (OC) value \(c_j\) associated with a domain to calculate the “novelty ratio” according to the fraction EUC/OC. We integrated the statistical overrepresentation analysis in the Taxy-Pro toolbox available at gobics.de/TaxyPro/.

### 2.4 GO enrichment analysis

The Pfam family descriptions usually characterize enzymatic, regulatory or structural properties of the associated domains on a very specific and non-standardized level. Conveniently, the Pfam database provides a mapping of a large number of domain families to the standardized vocabulary of the Gene Ontology (GO, [BLH+10]) database. In order to simplify the inspection of long lists of significantly overrepresented domain families w.r.t. to common properties, we perform an aggregation (“enrichment analysis”) of Pfam domain hits into GO term categories. For this purpose, we calculate for each GO term \(g_1, \ldots, g_M\) the sum of family-specific deviations for the domains for which a mapping to this GO term exist: \(g_j = \sum_i d_i \times I_{i,j}\). Here, the \(I_{i,j}\) are the elements of an indicator matrix which contains non-zero elements for those entries that correspond to mappings from a particular Pfam domain family to GO terms. Note that some Pfam families lack a mapping to GO terms and thus systematically do not contribute to the aggregation, which may introduce a bias towards more positive or negative deviations. Therefore, we excluded these families from the analysis and renormalized the vector of domain-specific deviations to zero-mean before computing the GO enrichment.
3 Results and discussion

Our approach to characterization of functional novelty in metagenomes is based on the analysis of unexplained protein domain hits as obtained from a mixture model for taxonomic profiling. In order to evaluate the utility of our method, we conducted several case studies using different real-world metagenomic datasets. While the first two studies focus on the characterization of novelty in single datasets, the third case represents a comparative analysis of the proposed novelty indicators for large metagenome data collections.

3.1 Characterization of metabolic novelty in complex communities

Metagenomes of complex communities such as found in biomass-degrading environments have been intensively investigated with respect to novel genes because of their relevance regarding biotechnological applications [HSE+11, SBD+08]. Therefore, as a first case study we analyzed a cow rumen metagenomic dataset (see also section 2.1). The Pfam domain detection resulted in a high fraction of sequences without valid domain assignments (FSU) of \( \approx 73\% \) but still provided a sufficient amount of domain hits (\( \approx 6.2 \times 10^6 \)) for our overrepresentation analysis (see section 2.3). A brief inspection of the taxonomic profile as estimated by the Taxy-Pro method revealed negligible proportions (<1%) of eukaryotes or viruses in the dataset, with Proteobacteria (\( \approx 42\% \)) and Firmicutes (\( \approx 13\% \)) accounting for the largest fractions of bacteria. Furthermore, the approximation error of the mixture model in terms of the fraction of domains unexplained (FDU: \( \approx 19\% \)) indicates a good model fit and a modest level of the overall novelty.

The analysis of unexplained domain counts in the cow rumen dataset resulted in a high number (1039) of potentially interesting families (p-value \( \leq 0.05 \)), which can partly be explained by the large number of reads associated with this dataset. However, by concentrating on the novelty indicators of our approach in terms of the number of expected unexplained domain counts (EUC, see section 2.3) and their fraction as compared to the original counts (OC), we restrict the discussion of results to the most promising candidates. Figure 1 shows a scatter plot of the significantly overrepresented Pfam families in terms of their EUC and EUC/OC ratio, whereby the nine data points associated with the highest EUC peak out of the distribution. Here, the “Leucine rich repeats” (LRR) family consisting of 6 copies (PF13306) shows the highest EUC (27,392) and a relatively high “novelty ratio” (EUC/OC) of \( \approx 62\% \). According to the family-specific Pfam summary, these leucine-rich repeat motifs are found in many (functionally unrelated) protein families and represent a structural rather than a specific functional property. Further inspection of figure 1 reveals functionally more specific domains such as two glycosyl hydrolases, a carboxylesterase and an aldo/keto reductase family. Glycosyl hydrolases are known to play an important role in fermentation and have been associated with novel genes during the analyses of the Cow rumen metagenome [HSE+11, FGC+05, FGB+12]. Interestingly, multi-domain proteins consisting of glycosyl hydrolase domains and leucine-rich repeats have been identified from newly sequenced species (e.g. UniProt accession I0TBE2, www.uniprot.org/uniprot/I0TBE2). However, the combined appearance of such domains
can usually not be identified in short-read data prior to an assembly of the reads. Our findings suggest that a large amount of novelty in the Cow rumen dataset may be represented by yet unknown combinations of LRR with enzymatic domains.

In principle, metagenomes with a high degree of overall novelty (and thus a high FDU) are particularly interesting for identification of novel genes, however, an FDU of $>80\%$ indicates an insufficient reconstruction of the metagenomic profile by reference organisms [KALM13]. As a second case study and as an example of an environment with extreme conditions, we analyzed a deep-sea sediment metagenome from the western Gulf of Mexico (see also section 2.1). The ‘Sediment’ dataset showed a high FSU ($\approx 91\%$) with only 37,318 domain hits and a fairly high FDU ($\approx 58\%$) that already indicates the limits of the mixture model. Analysis of unexplained domain counts resulted in 229 significantly over-represented families many of which can be related to iron supply to the cell under anaerobic conditions. The highest degree of novelty was observed for a “Ferrous iron transport protein B” (PF02421; EUC: 1079, EUC/OC: 99\%) which has already been identified during the initial analysis of this sample [BWTH11]. The large amount of novelty related to iron processing can easily be identified in the list of Gene Ontology terms as obtained by our enrichment analysis (see section 2.4). Table 1 shows the top ten GO terms associated
Table 1: Top ten overrepresented GO categories for the Sediment dataset according to the sum of deviations (SOD, column three) of associated domain families.

<table>
<thead>
<tr>
<th>GO ID</th>
<th>GO name</th>
<th>SOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0006812</td>
<td>cation transport</td>
<td>0.050</td>
</tr>
<tr>
<td>GO:0006811</td>
<td>ion transport</td>
<td>0.045</td>
</tr>
<tr>
<td>GO:0003674</td>
<td>molecular function</td>
<td>0.045</td>
</tr>
<tr>
<td>GO:0006810</td>
<td>transport</td>
<td>0.042</td>
</tr>
<tr>
<td>GO:0005488</td>
<td>binding</td>
<td>0.040</td>
</tr>
<tr>
<td>GO:0044765</td>
<td>single-organism transport</td>
<td>0.040</td>
</tr>
<tr>
<td>GO:0000041</td>
<td>transition metal ion transport</td>
<td>0.037</td>
</tr>
<tr>
<td>GO:0015684</td>
<td>ferrous iron transport</td>
<td>0.036</td>
</tr>
<tr>
<td>GO:0015093</td>
<td>ferrous iron transmembrane transporter activity</td>
<td>0.036</td>
</tr>
<tr>
<td>GO:0072511</td>
<td>divalent inorganic cation transport</td>
<td>0.036</td>
</tr>
</tbody>
</table>

with the overrepresented families. Here, the GO categories are sorted according to the sum of deviations of the associated domain families. The list allows an intuitive interpretation of the distribution of novelty over different domain families and a quick characterization of the functional novelty in terms of a standardized vocabulary of metabolic processes.

3.2 Large-scale comparative analysis on human microbiome data

Our third case study describes the analysis of novelty in large collections of short-read data and the comparison of novelty profiles. Here, we used 750 metagenome datasets from the human microbiome project (HMP) associated with five major body sites (see section 2.1). The average number of significantly overrepresented domain families over a body site varied between \(\sim 200\) (Airways) and \(\sim 330\) (Oral cavities). The average FDU ranged from \(\approx 8\%\) (Oral cavities and Gastrointestinal tract) to \(17\%\) (Urogenital tract), indicating a good fit of the mixture model in general.

For a more detailed analysis of the HMP dataset we focused on Oral samples, which showed the highest degree of novelty in terms of the number of significant domain families in our overrepresentation analysis. Here, we compared samples from three out of nine subsites with more than 100 associated datasets: tongue dorsum (135 samples), supragingival plaque (128) and buccal mucosa (122). In order to aggregate the results, we calculated the average EUC for each domain family and subsite over all samples. Table 2 shows the top ten domain families associated with a high overrepresentation in terms of maximum average EUC regarding the oral subsites. Here, the “IgA1-specific Metalloendopeptidase” (PF07580) on rank one can be identified as representing a high amount of novelty in the buccal mucosa samples. IgA1 peptidases cleave specific peptide bonds in mammalian immunoglobulin A1 (IgA1) and can be found in particular in pathogenic bacteria. At mucosal sites of infection they can destroy the structure and function of human IgA1 and eliminate an important aspect of host defense [MS06]. On rank five in the list, a “G5 domain” (PF07501) also shows a high degree of novelty associated with buccal
Table 2: Average expected unexplained counts (EUC, rounded) for Pfam domains associated with significantly overrepresented domain families for three oral subsites from HMP datasets. Domain families are sorted from high to low according to the maximum average EUC regarding the subsites (in boldface type).

<table>
<thead>
<tr>
<th>Pfam ID</th>
<th>name</th>
<th>tongue</th>
<th>plaque</th>
<th>mucosa</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF07580</td>
<td>M26 IgA1-specific Metallo-endopeptidase C-terminal region</td>
<td>11</td>
<td>100</td>
<td>1568</td>
</tr>
<tr>
<td>PF00496</td>
<td>Bacterial extracellular solute-binding proteins, family 5 Middle</td>
<td>432</td>
<td>285</td>
<td>1452</td>
</tr>
<tr>
<td>PF07690</td>
<td>Major Facilitator Superfamily</td>
<td>59</td>
<td>779</td>
<td>242</td>
</tr>
<tr>
<td>PF12698</td>
<td>ABC-2 family transporter protein</td>
<td>12</td>
<td>61</td>
<td>771</td>
</tr>
<tr>
<td>PF07501</td>
<td>G5 domain</td>
<td>18</td>
<td>94</td>
<td>735</td>
</tr>
<tr>
<td>PF01610</td>
<td>Transposase</td>
<td>165</td>
<td>12</td>
<td>718</td>
</tr>
<tr>
<td>PF08428</td>
<td>Rib/alpha-like repeat</td>
<td>9</td>
<td>25</td>
<td>650</td>
</tr>
<tr>
<td>PF00005</td>
<td>ABC transporter</td>
<td>193</td>
<td>92</td>
<td>626</td>
</tr>
<tr>
<td>PF07564</td>
<td>Domain of Unknown Function (DUF1542)</td>
<td>5</td>
<td>58</td>
<td>585</td>
</tr>
<tr>
<td>PF00593</td>
<td>TonB dependent receptor</td>
<td>324</td>
<td>499</td>
<td>65</td>
</tr>
</tbody>
</table>

mucosa (see also summary of Pfam database entry). The G5 domain is found in the N-terminus of peptidases belonging to the M26 family and is suspected to have an adhesive function. Furthermore, on rank nine we can see a mucosa-specific domain of unknown function (“DUF1542”, PF07564). This domain is found in several cell surface proteins some of which are involved in antibiotic resistance and/or cellular adhesion (see also Pfam summary). Our findings suggest that a high degree of novelty that we identified is related to yet unknown mucosa-specific bacteria providing many proteins related to pathogenicity. This first glimpse of metagenomic novelty in the human microbiome already indicates the potential utility of our approach for medical research.

4 Conclusion

We presented an approach for the characterization of metagenomic novelty based on the analysis of the functional profile of a metagenome. Using the unexplained protein domain hits as obtained from the UProC tool for domain detection and the Taxy-Pro mixture model estimation, the identification of significant domain families usually takes a few minutes for a real-world metagenomic dataset. The results in terms of lists of domain families highlight the functional and structural properties that are not well-explained by the mixture model and therefore provide interesting candidates for further analysis. We are aware that using binomial distributions under the assumption of statistical independence is a coarse approximation which is likely to overestimate the number of significant differences. This shortcoming can be overcome if biological (or technical) replicates are available which would allow the application of more sophisticated models, such as the negative binomial distribution that is often used in RNA-seq analysis.
Classical gene mining approaches are based on a bottom up strategy that requires the identification of complete genes, which typically extend to lengths around 1000 bp for microbial organisms. Therefore, the methods require datasets with long sequencing reads or have to rely on an assembly of reads prior to gene identification. In contrast, our mixture model is based on functional profiles that are estimated by detecting and counting Pfam protein domains. Therefore the method can even be applied to short-read data as shown in our studies on rumen and HMP samples. In that way, our approach can be used for novelty mining on large metagenomic data collections to identify interesting habitats and samples that may be further explored by functional screening methods. On the other side, datasets with a high degree of novelty usually yield a large proportion of sequence reads without domain assignments, which, as a consequence, do not contribute to the characterization of novelty. However, this problem affects all approaches for gene mining and can only be solved by a higher coverage of functionally annotated sequences in reference databases.

Using sequences without similarities to known families, the interpretation of results from a classical bottom up analysis can be rather difficult. In the extreme, a cluster analysis may end up in a large number of putatively new protein families with no or only little evidence for functional properties [YSR+07]. In contrast, our top down approach always yields a prediction of novelty in terms of biologically defined categories. However, the descriptions of Pfam domain families are often not easy to interpret in terms of metabolical relevance. In this study we showed how a mapping of domains to Gene Ontology (GO) categories can be used to facilitate the interpretation of functional annotations. On the downside, the lack of mappings for many Pfam domains and the hierarchical structure of GO also result in a systematic overrepresentation of functionally unspecific terms. Future work will focus on the extension of our approach to protein families that can directly be associated with metabolic pathways [KGS+14]. Furthermore, we plan to integrate the characterization of functional novelty into the CoMet web server for comparative functional profiling of metagenomes [LASM11].

5 Acknowledgments

We thank Heiner Klingenberg for technical support. This work was supported by the Deutsche Forschungsgemeinschaft (grant Me3138 to P.M.).

References


Abstract: We review the level of genomic specificity regarding actinobacterial pathogenicity. As they occupy various niches in diverse habitats, one may assume the existence of lifestyle-specific genomic features. We include 240 actinobacteria classified into four pathogenicity classes: human pathogens (HP), broad-spectrum pathogens (BP), opportunistic pathogens (OP), and non-pathogenic (NP). We hypothesize: (H1) Pathogens (HPs and BPs) possess specific pathogenicity signature genes. (H2) The same holds for opportunistic pathogens. (H3) Broad-spectrum and exclusively human pathogens cannot be distinguished from each other due to an observation bias, i.e. many HPs might be yet unclassified BPs. (H4) There is no intrinsic genomic characteristic of opportunistic pathogens compared to pathogens, as small mutations are likely to play a more dominant role in order to survive the immune system. To study these hypotheses, we implemented a bioinformatics pipeline that combines evolutionary sequence analysis with statistical learning methods (Random Forest with feature selection, model tuning and robustness analysis). Essentially, we present orthologous gene sets that computationally distinguish pathogens from non-pathogens (H1). We further show a clear limit in differentiating opportunistic pathogens from both, non-pathogens (H2) and pathogens (H4). Human pathogens may also not be distinguished from bacteria annotated as broad-spectrum pathogens based on a small set of orthologous genes only (H3), as many human pathogens might as well target a broad range of mammals but have not been annotated accordingly. In conclusion, we illustrate that even in the post-genome era and despite next-generation sequencing technology our ability to efficiently deduce real-world conclusions, such as pathogenicity classification, remains quite limited.
1 Background and Results

With the emergence of the so-called next-generation sequencing technology, the available data sets exploded such that we have over 27,000 registered sequencing projects at NCBI (NCBI web site, August 1, 2014). We now wonder to what extent we can deduce real-world qualitative information from this treasure of data. The aim of this paper was to investigate the power of computational functional genomics to predict bacterial lifestyles utilizing DNA sequence information only. Specifically, we asked the question if we may utilize sequence similarity to identify dedicated lifestyle-specific protein-coding genes. We restrict our report, first, to a set of 240 well-studied actinobacterial genomes and, second, to four pathogenicity lifestyles, namely: human pathogens (HP), broad-spectrum pathogens (BP), opportunistic pathogens (OP), and non-pathogenic (NP).

In [EB14], we illustrate and quantify the boundaries we face when trying to deduce a specific microbial pathogenicity class if given the genomic repertoire only, at least in the case of actinobacteria (see figure 1 below). In summary, we show that we indeed find signature genes that differentiate pathogens from non-pathogens. Only a small set of three genes for each bias, i.e. classification direction, is sufficient to reach an approximately 90% accuracy (figures 2-4). When trying to classify the different pathogenicity lifestyles though, it appears that too many confounding factors unbalance our data sets such that we cannot differentiate, for instance, a strain-specific from a lifestyle-specific gene.

We conclude that even in the post-genome era, and even for supposedly simple questions, our ability to efficiently deduce real-world implications from large-scale de novo next-generation sequencing data remains quite limited.
Figure 1: Illustration of our bias introduction strategy. Distribution of homologous gene clusters over two lifestyles (pathogens vs. non-pathogens) and illustration of our strategy to introduce a feature selection bias into our statistical learning pipelines. Both axes in all three plots describe the percentage of species in the respective class(es), here human pathogens (HP) and broad pathogens (BP) vs. non-pathogens (NP). The color-coding of the heat map depicts the number of clusters of homologous genes that certain percentages of pathogens/non-pathogens share. Thus, in the upper right of such a joint distribution plot, we find the core genome (homologous genes present in all species of both classes); and in the lower left, we see unique, species-specific genes. Genes close to the axis are more class specific. Genes close to the axis tails are highly class specific and, thus, the distinctive homologous gene candidates we were hoping to find. As there is no single such gene, we scanned for sets of lifestyle-distinctive genes. To find such feature genes for pathogenic lifestyles, for instance, we remove all genes that are found more often in non-pathogens (NP) than in pathogens (HP+BP), i.e. the gene clusters below the dotted line in the upper plot, such that our follow-up machine learning routines are biased towards utilizing pathogenicity-specific features (genes in the bottom left plot) for classification.
Figure 2: **Classification performance non-pathogens vs. pathogens.** ROC (receiver operating characteristics) plots were generated to inspect the performance of the classification models. The data was evaluated five times using different 5-fold cross validation sets to receive robust quality estimations of our classifiers. The real label classifier curves are presented in dark green dashed lines, while the random label classifier curves are given in light green dotted lines (the ones close to the base line). The variation of the AUCs (area under curve) in the cross validation was included in the figure as a box plot (bottom right). The numbers below each box plot are the lower and upper quartiles. 

**a)** Pathogen classifier results (NP vs. HP+BP). We biased the predictors towards using pathogen-specific genes (see Figure 1). 

**b)** Non-pathogen classifier results (NP vs. HP+BP) where the predictor now was biased to prefer the non-pathogen-specific genes. See text for a full description of our machine learning strategy and refer to Figure 1 regarding the “bias”.

---

82
Figure 3: Decision tree created using the genes most discriminative for pathogen (HP+BP). Our classification pipeline (see full text) selected the above three genes as most representative for pathogens. We learned and visualize them as a simple decision tree by using the RapidMiner software. Nodes represent gene clusters with the following Transitivity Clustering IDs: 219529, 205393 and 221713, which are associated to the GenBank annotations “Phosphate permease” (e.g. UniprotKB AC: I6YD06 or P65712), “ABC transporter ATP-binding protein” (e.g. UniprotKB AC: D9Q9K6) and “transmembrane protein” (e.g. UniprotKB AC: G2MY46), respectively. The small circles close to the Transitivity Clustering IDs indicate the cluster size. Using only these three features the decision tree already obtains an accuracy of 93.9%.
Figure 4: Decision tree created using the genes most discriminative for non-pathogen (NP).
Our classification pipeline (see full text) selected the above three genes as most representative for pathogens. We learned and visualize them as a simple decision tree by using the RapidMiner software. Nodes represent gene clusters with the following Transitivity Clustering IDs: 210148, 209987 and 211191, which are associated to the GenBank annotations “Threonine dehydratase” (e.g. UniprotKB AC: E3ERF0), “Beta-galactosidase” (e.g. UniprotKB AC: D6Y6J1) and “ATP-dependent DNA helicase” (e.g. UniprotKB AC: G0FLF9), respectively. The small circles close to the Transitivity Clustering IDs indicate the cluster size. Using only these three features the decision tree already obtains an accuracy of 89.9%.

References

A Pipeline for Insertion Sequence Detection and Study for Bacterial Genome

Huda Al-Nayyef¹,², Christophe Guyeux¹, and Jacques M. Bahi¹

¹ FEMTO-ST Institute, UMR 6174 CNRS, DISC Computer Science Department
Université de Franche-Comté, 16, Rue de Gray, 25000 Besançon, France
² Computer Science Department, University of Mustansiriyah, Iraq
{huda.al-nayyef, christophe.guyeux, jacques.bahi}@univ-fcomte.fr

Abstract: Insertion Sequences (ISs) are small DNA segments that have the ability of moving themselves into genomes. These types of mobile genetic elements (MGEs) seem to play an essential role in genomes rearrangements and evolution of prokaryotic genomes, but the tools that deal with discovering ISs in an efficient and accurate way are still too few and not totally precise. Two main factors have big effects on IS discovery, namely: genes annotation and functionality prediction. Indeed, some specific genes called “transposases” are enzymes that are responsible of the production and catalysis for such transposition, but there is currently no fully accurate method that could decide whether a given predicted gene is either a real transposase or not. This is why authors of this article aim at designing a novel pipeline for ISs detection and classification, which embeds the most recently available tools developed in this field of research, namely OASIS (Optimized Annotation System for Insertion Sequence) and ISFinder database (an up-to-date and accurate repository of known insertion sequences). As this latter depend on predicted coding sequences, the proposed pipeline will encompass too various kinds of bacterial genes annotation tools (that is, Prokka, BASys, and Prodigal). A complete IS detection and classification pipeline is then proposed and tested on a set of 23 complete genomes of Pseudomonas aeruginosa. This pipeline can also be used as an investigator of annotation tools performance, which has led us to conclude that Prodigal is the best software for IS prediction. A deepen study regarding IS elements in P. aeruginosa has then been conducted, leading to the conclusion that close genomes inside this species have also a close numbers of IS families and groups.

1 Introduction

The number of completely sequenced bacterial and archaeal genomes are rising steadily, such an increasing makes it possible to develop novel kind of large scale approaches to understand genomes structure and evolution over time. Gene content prediction and genome comparison have both provided new important information and deciphering keys to understand evolution of prokaryotes [VSG⁺¹¹]. Important sequences in understanding rearrangement of genomes during evolution are so-called transposable elements (TEs), which are DNA fragments or segments that have the ability to insert themselves into new chromosomal locations, and often make duplicate copies of themselves during transposition process [HKN⁺'06]. Remark that, in bacterial reign, only cut-and-paste mechanism of
transposition can be found, the transposable elements involved in such a move being the insertion sequences (ISs).

Insertion sequences range in size from 600 to more than 3000 bp. They are divided into 26 main different families in prokaryotes, as described in ISFinder\(^1\) \([\text{SPL}^{+}06]\), an international reference database for bacterial and archaeal ISs that includes background information on transposons. The main function of ISFinder is to assign IS names and to produce a focal point for a coherent nomenclature for all discovered insertion sequences. This database includes over than 3500 bacterial ISs \([\text{HCD}10, \text{ZOX}08]\). Data come from a detection of repeated patterns, which can be easily found by using homology-based techniques \([\text{FP}07]\). Classification process of families, for its part, depends on transposases homology and overall genetic organization. Indeed, most ISs consist of short inverted repeat sequences that flank one or more open reading frames (ORFs, see Figure 1), whose products encode the transposase proteins necessary for transposition process. The main problem with such approaches for ISs detection and classification is that they are obviously highly dependent on the annotations, and existing tools evoked above only use the NCBI ones, whose quality is limited and very variable.

In this research work, the authors’ intention is to find an accurate method for discovering insertion sequences in prokaryotic genomes. To achieve this goal, we propose to use one of the most recent computational tool for automated annotation of insertion sequences, namely OASIS, together with the international database for all known IS sequences (ISFinder). More precisely, OASIS works with genbank files that have fully described genes functionality: this tool identifies ISs in each genome by finding conserved regions surrounding already-annotated transposases. Such technique makes it possible to discover new insertion sequences, even if they are not in ISFinder database. A novel pipeline that solves the dependence on NCBI annotations, and that works with any annotation tool (with or without description of gene functionality) is then proposed. The output of our pipeline contains all detected IS sequences supported with other important information like inverted repeats (IRs) sequences, lengths, positions, names of family and group, and other details that help in studying IS structures.

The contributions of this article can be summarized as follows. (1) A pipeline for insertion sequences discovery and classification is proposed, which does not depend on NCBI annotations. It uses unannotated genomes and embeds various annotation tools specific to Bacteria (such as Prokka, BASys, and Prodigal) in its process. (2) Overlapping and consensus problems that naturally appear after merging annotation methods recalled above are solved, in order to obtain large and accurate number of ISs with their names of families and groups. And finally (3) the pipeline is tested on a set of 23 complete genomes of \textit{Pseudomonas aeruginosa}, and biological consequences are outlined.

The remainder of this article is organized as follows. In Section 2, various tools for discovering IS elements in different species of Bacteria and Archaea are presented. The suggested methodology for increasing both the number and accuracy of detecting IS elements is explained in Section 3. The pipeline is detailed in Section 4, while an application example using 23 completed genomes of \textit{P. aeruginosa} is provided in Section 5. This arti-

\(^1\)www-is.biotoul.fr
State of the art in ISs detection or annotation

The study on the plant-pathogenic prokaryote *Xanthomonas oryzae pv. oryzae* (*Xoo*), which causes bacterial blight (one of the most important diseases of rice) was published in 2005 by Ochiai *et al.* [OIT+05]. They used GeneHacker [YH96], GenomeGambler version 1.51, and Glimmer program [DHK+99] for coding sequence prediction. Insertion sequences were finally classified by a BLAST analysis using ISFinder database evoked previously.

**IScan**, developed by Wagner *et al.* [WLB07], has then been proposed in 2007. Inverted repeats are found using smith waterman local alignments on transposase references found with BLAST and used as a local database. This tool has been applied on 438 completely sequenced bacterial genomes by using BLAST with referenced transposases, to determine which transposases are related to insertion sequences. Touchon *et al.,* for their parts, have analyzed 262 different bacterial and archaeal genomes downloaded from GenBank NCBI in 2007 [TR07]. A coding sequence has then been considered as an IS element if its BLASTP best hit in ISFinder database has an e-value lower than $10^{-10}$.

**ISA** has been created by Zhou *et al.* in 2008 [ZOX08]. This annotation program depends on both NCBI annotations and ISFinder. More precisely, authors manually collected 1,356 IS elements with both sequences and terminal signals from the ISFinder database, which have been used as templates for identification of all IS elements and map construction in the targeted genomes. ISA, which is not publicly available, has finally been used for an analysis of 19 cyanobacterial and 31 archaeal annotated genomes downloaded from NCBI.

In 2010, Plague *et al.* analyzed the neighboring gene orientations (NGOs) of all ISs in 326 fully sequenced bacterial chromosomes. They obtained primary annotations from the Comprehensive Microbial Resource database (release 1.0-20.0) at the Institute for Genomic Research\(^2\). Their approach for extracting IS elements from these genomes was to consider that a coding sequence with a best BLASTX hit e-value lower than $10^{-10}$ is an insertion sequence [Pla10]. **ISSage**, for its part, has been developed in 2011 by Varani

\(^2\)http://cmr.tigr.org/tigr-scripts/CMR/CmrHomePage.cgi

---

**Figure 1:** IS element types [ZOX08]
et al. [VSG et al.]. They used eight different bacterial genomes downloaded from NCBI, and produced a web application pipeline that allows semi-automated annotation based on BLAST against the ISFinder database. However, ISsage cannot automatically identify new insertion sequences which are not already present in ISFinder database.

A new computational tool for automated annotation of ISs has then been released in 2012 by Robinson et al. [RLM12]. This tool has been called OASIS, which stands for “Optimized Annotation System for Insertion Sequences”. They worked with 1,737 bacterial and archaeal genomes downloaded from NCBI. OASIS identifies ISs in each genome by finding conserved regions surrounding already-annotated transposase genes. OASIS uses a maximum likelihood algorithm to determine the edges of multicopy ISs based on conservation between their surrounding regions. For defining inverted repeats, the same strategy as IScan was used (Smith-Waterman alignment). Authors also used hierarchical agglomerative clustering to identify groups of IS lengths. The IS set is then classified according to the family and group after a BLASTP best hit in ISFinder database with an e-value lower than $10^{-12}$. When a cluster cannot match with any entry of the database, the IS set is considered as new. Thus OASIS has the ability to discover new insertion sequences, that is, which cannot be found in ISFinder.

Table 1: Input set of 23 complete genomes of *P. aeruginosa*

<table>
<thead>
<tr>
<th>Index</th>
<th>Genome Name</th>
<th>INSDC(Genbank)</th>
<th>Refseqs</th>
<th>Input Genomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PACS2</td>
<td>106896550</td>
<td>AAQW01000001.1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PAO1</td>
<td>110227054</td>
<td>NC002516.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>UCBPP-PA14</td>
<td>115583796</td>
<td>NC008463.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PA7</td>
<td>150958624</td>
<td>NC009656.1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>19BR</td>
<td>343788106</td>
<td>485462089</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>213BR</td>
<td>343788107</td>
<td>485462091</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>M18</td>
<td>347302377</td>
<td>386056071</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DK2</td>
<td>392316915</td>
<td>392981410</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>B136-33</td>
<td>477548288</td>
<td>478476202</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>RP73</td>
<td>514245605</td>
<td>514407635</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>c3447m</td>
<td>543873856</td>
<td>543873856</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>PA0581</td>
<td>543879514</td>
<td>543879514</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>PA01-VE2</td>
<td>553886202</td>
<td>553886202</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>PA01-VE13</td>
<td>553895034</td>
<td>553895034</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>PA1</td>
<td>557703951</td>
<td>558672313</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>PA1R</td>
<td>557709751</td>
<td>558665962</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>MTB-1</td>
<td>564088818</td>
<td>564949884</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>LES431</td>
<td>566561164</td>
<td>568151185</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SCV20265</td>
<td>567363169</td>
<td>568306739</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>LESB58</td>
<td>218888746</td>
<td>218888746</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>NCJM2.S1</td>
<td>386062973</td>
<td>386062973</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>PA38182</td>
<td>575870901</td>
<td>575870901</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>YL84</td>
<td>576902775</td>
<td>576902775</td>
<td></td>
</tr>
</tbody>
</table>

Finally, in 2014, the analysis of the NGOs for all IS elements within 155 fully sequenced Archaea genomes was presented by Florek et al. [FGP14]. To do so, they have launched a BLASTP in the ISFinder, with an e-value less than or equal to $10^{-10}$, for all protein coding sequences downloaded from NCBI which are related to IS elements.

Two major concerns with the state of the art detailed above can be emphasized. Firstly,
most of them cannot detect new insertion sequences. Secondly, all these tools are based on NCBI annotations of very relative and variable qualities – except ISsaga, which could work with other annotation tools (but it depends only on transposase ORFs that have been already defined in ISFinder). Our objective in the next section is to propose a pipeline that solves these two issues, being able to deal with unannotated genomes and to detect unknown ISs.

3 Prediction and Modules based on OASIS

For illustration purpose, the proposed pipeline system for IS elements prediction will be presented using 23 complete genomes of *P. aeruginosa* available on the NCBI website, RefSeq and INCDS/Genebank databases, see Table 1 (RefSeq genomes were preferred when available). The prediction of IS elements in the proposed pipeline depends on both OASIS [RLM12] and ISFinder [SPL+06].

3.1 Prediction of IS elements from *Pseudomonas aeruginosa*

OASIS is used in this pipeline for predicting insertion sequences in prokaryotic genomes. This latter detects ISs in each genome by finding conserved regions surrounding already-annotated transposase genes, which are identified by the word *transposase* in the “product” field of the GenBank file. Obviously OASIS highly depends on the quality of annotations [RLM12], while to determine whether a given gene is a transposase or not is a very difficult task (indeed transposases are among the most abundant and ubiquitous genes in nature [ABE10], and they are widely separated in Prokaryote genomes). OASIS deals with files having genbank format. It takes them as input and then produces two output files for each provided genome. The first one is a fasta file that contains all IS nucleotide sequences, with start and end positions. It also contains the amino acid sequence for each ORF. The second file is a summary table providing attributes that describe the insertion sequence: set-id, family, group, IS positions, inverted repeat left (IRL) and right (IRR), and orientation. Remark that most of these information are in the ISFinder database too. Indeed OASIS find them alone but it extracts family names and group from ISFinder.

The main problem found in OASIS is solved in the proposed pipeline by using different types of annotations: NCBI will not be used alone, and gene functionality taken from annotation tools will either or not be used depending on the situation. Finally, transposases within IS will be verified using ISFinder database. OASIS can thus be used in two different ways in our pipeline, depending on the provided genbank file. These two modules have been named NOASIS, which uses the original input genbank genome file provided by the NCBI (as it is, without any modification), and DOASIS, which deals with modified genbank files that have been updated to obtain more accurate results than NOASIS. These modules are described thereafter.
3.2 Normal OASIS (NOASIS)

For finding predicted IS in NOASIS module, we simply applied OASIS on the input set of genomes with their NCBI annotations, that is, with the original downloaded genbank file. Using the reference genome named PAO1, the summary outputted by the pipeline is given in Tables 2 and 3. In these NOASIS tables, the summary produced by OASIS is enriched with new features described below:

- **Real IS** IS sequences that have best match (first hit) when using BLASTN with ISFinder database, an e-value equal to 0.0, and with a functionality of each ORF within the IS recognized as a transposase.

- **Partial IS** Sequences that match part of known IS from ISFinder (i.e., have e-value lower than $10^{-10}$) and have also a transposase gene functionality for the ORFs.

- **Putative New IS** Sequences with bad score after making a BLASTN with ISFinder, but with a transposase. They may be real insertion sequences not already added in ISFinder database or false positives, requiring human curation.

Applying this slightly improved version of OASIS in the 23 genomes of *Pseudomonas* leads to a major issue: surprisingly, NOASIS found no real insertion sequences in some genomes like PACS2 or SCV20265. The problem is that OASIS find multiple copies of IS elements in each genome by identifying conserved regions surrounding transposase genes. However some of the considered genomes either have no information about transposase gene into their feature genbank tables or have simply no feature table in their genbank format files. This issue is at the basis of our improved module called DOASIS, which is explained below.

For the sake of comparison, Figure 2 contains similar results for *Mycobacterium tuberculosis* genus.

3.3 Developed OASIS (DOASIS)

The main idea for DOASIS module is that information about transposases within genbank files are potentially incorrect (i.e., may all be false positives). So we simply decide to remove all transposase words in the product fields from all inputted genomes. We thus update these information as follows.

**Step 1: genbank update.** Inputted genbank files are modified following one of the three methods below.

1. **All-Tpase**: we consider that all the genes may potentially be a transposase. So all product fields are set to “transposase”.
2. **Zigzag Odd**: we suggest that genes in odd positions are putative transposases and we update the genbank file adequately. Oddly, this new path will produce new candidates which are not detected during All-Tpase.
Figure 2: IS elements detect in 28 Mycobacterium tuberculosis

Figure 3: Comparison of predicted ISs between randomization method and all/odd/even methods.
3. **Zigzag Even**: similar to Zigzag Odd, but on even positions.

We checked also a randomized method (i.e., by putting “transposase” in randomly picked genes). However we found poorer number of predictive real ISs or new real ISs compared with the three methods previously presented. For these reasons, we will not further investigate the randomized method.

**Step 2.** We apply OASIS three times (i.e. one time per method) on all genomes, and then we take the output fasta file that contains both nucleotides and amino acids sequences for each IS element.

**Step 3.** A BLASTN with ISFinder is applied on each IS sequence. If the e-value of the first hit is 0.0, then the ORF within this IS belongs to known (Real) IS already existing in the ISFinder database. Else, if the e-value is lower than $10^{-10}$, then we found a Partial IS.

**Step 4.** Collect all Real IS from previous three methods (ALL_Tpase, Zigzag odd, and Zigzag even) and then remove overlaps among them. Finally, produce best Real IS with all information. Remark that the problem of finding consensus and overlaps can be treated as a lexical parsing problem.

### 4 The Proposed Pipeline

It is now possible to describe the proposed pipeline that can use the two modules detailed in the previous section. This pipeline, depicted in Figure 4, will increase the number of Real IS detected on the set of *P. aeruginosa* genomes under consideration (indeed, the detection is improved in all categories of insertion sequences, but we only focus on Real IS in the remainder of this article, for the sake of concision). Its steps are detailed in what follows.

**Step 1: ORF identification.** Our pipeline is currently compatible with any type of annotation tools, having either functionality capability or not, but for comparison we only focus in this article on the following tools: BASys, Prokka, and Prodigal. BASys (Bacterial Annotation System) is a web server that performs automated, in-depth annotation of bacterial genomic (chromosomal and plasmid) sequences. It uses more than 30 programs to determine nearly 60 annotation subfields for each gene. Remark that genomes must be sent online manually, and that some curation stage may be required to remove some DNA ambiguity on returned genbank files.

Prokka (rapid prokaryotic genome annotation), for its part, is a classical command line software for fully annotating draft bacterial genomes, producing standards-compliant output files for further analysis [See14]. Finally, Prodigal (Prokaryotic Dynamic Programming Genealgorithm) is an accurate bacterial and archaeal genes finding software provided by the Oak Ridge National Laboratory [HCL+10].

**Step 2: IS Prediction.** The second stage of the pipeline consists in using either NOASIS or DOASIS for predicting IS elements. Notice that NOASIS cannot be used
with Prodigal, as this module requires information about gene functionality (both NOASIS and DOASIS can be use with Prokka and BASys annotations).

**Step 3: IS Validation.** This step is realized by launching BLASTN on each predicted IS sequence with ISFinder. The e-value of the first hit is then checked: if it is 0.0, then the ORF within this sequence is a Real IS known by ISFinder. As described previously, it will be considered as Partial IS if its e-value is lower than $10^{-10}$. Both IS names of family and group are returned too.

**Figure 4: The proposed pipeline**

5 Results and Discussion

We can firstly remark in Figure 5 that, using either Prokka or BASys for genes detection and functionality prediction is better than taking directly the annotated genomes from NCBI: a larger number of Real IS can be found. Additionally, this comparison shows that
Prokka outperforms BASys in 3 families of ISs (namely: IS3, IS30, and ISNCY), while BASys seems better for detecting insertion sequences belonging in the IS5, IS1182, and TN3 families. This variability may be explained by the fact that functionality annotations of these tools depend probably on IS families that where known when these tools have been released.

Table 2: Summary table produced by NOASIS (beginning)

<table>
<thead>
<tr>
<th>Name</th>
<th>Genome</th>
<th>Start</th>
<th>End</th>
<th>Orientation</th>
<th>SetID</th>
<th>ISFinder_name</th>
<th>Family</th>
<th>Group</th>
<th>Group</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAO1</td>
<td>NC_002516.2</td>
<td>499832</td>
<td>501193</td>
<td>-</td>
<td>1</td>
<td>ISPa11</td>
<td>IS10</td>
<td>IS111</td>
<td>IS111</td>
<td>1361</td>
</tr>
<tr>
<td>PAO1</td>
<td>NC_002516.2</td>
<td>2556875</td>
<td>2558236</td>
<td>+</td>
<td>1</td>
<td>ISPa11</td>
<td>IS10</td>
<td>IS111</td>
<td>IS111</td>
<td>1361</td>
</tr>
<tr>
<td>PAO1</td>
<td>NC_002516.2</td>
<td>3043478</td>
<td>3044893</td>
<td>-</td>
<td>1</td>
<td>ISPa11</td>
<td>IS10</td>
<td>IS111</td>
<td>IS111</td>
<td>1361</td>
</tr>
<tr>
<td>PAO1</td>
<td>NC_002516.2</td>
<td>3842002</td>
<td>3843637</td>
<td>-</td>
<td>1</td>
<td>ISPa11</td>
<td>IS10</td>
<td>IS111</td>
<td>IS111</td>
<td>1361</td>
</tr>
<tr>
<td>PAO1</td>
<td>NC_002516.2</td>
<td>4473550</td>
<td>4474911</td>
<td>+</td>
<td>1</td>
<td>ISPa11</td>
<td>IS10</td>
<td>IS111</td>
<td>IS111</td>
<td>1361</td>
</tr>
<tr>
<td>PAO1</td>
<td>NC_002516.2</td>
<td>5382524</td>
<td>5383885</td>
<td>-</td>
<td>1</td>
<td>ISPa11</td>
<td>IS10</td>
<td>IS111</td>
<td>IS111</td>
<td>1361</td>
</tr>
<tr>
<td>PAO1</td>
<td>NC_002516.2</td>
<td>54041</td>
<td>54835</td>
<td>+</td>
<td>2</td>
<td>ISStma5</td>
<td>IS3</td>
<td>IS3</td>
<td>IS3</td>
<td>794</td>
</tr>
</tbody>
</table>

Table 3: Summary table produced by NOASIS (end)

<table>
<thead>
<tr>
<th>IRR=IRL</th>
<th>Locus_tag(gbk)</th>
<th>Product(gbk)</th>
<th>E Value</th>
<th>IS_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATGGACTCCTCCC</td>
<td>[&quot;PA0445&quot;]</td>
<td>[&quot;transposase&quot;]</td>
<td>0.0</td>
<td>Real IS</td>
</tr>
<tr>
<td>ATGGACTCCTCCC</td>
<td>[&quot;PA2319&quot;]</td>
<td>[&quot;transposase&quot;]</td>
<td>0.0</td>
<td>Real IS</td>
</tr>
<tr>
<td>ATGGACTCCTCCC</td>
<td>[&quot;PA2690&quot;]</td>
<td>[&quot;transposase&quot;]</td>
<td>0.0</td>
<td>Real IS</td>
</tr>
<tr>
<td>ATGGACTCCTCCC</td>
<td>[&quot;PA3434&quot;]</td>
<td>[&quot;transposase&quot;]</td>
<td>0.0</td>
<td>Real IS</td>
</tr>
<tr>
<td>ATGGACTCCTCCC</td>
<td>[&quot;PA3993&quot;]</td>
<td>[&quot;transposase&quot;]</td>
<td>0.0</td>
<td>Real IS</td>
</tr>
<tr>
<td>ATGGACTCCTCCC</td>
<td>[&quot;PA4797&quot;]</td>
<td>[&quot;transposase&quot;]</td>
<td>0.0</td>
<td>Real IS</td>
</tr>
<tr>
<td>AAAGGGGACAGATTTATTTTCCCTGCTCTAAT</td>
<td>[&quot;PA0041a&quot;]</td>
<td>[&quot;transposase&quot;]</td>
<td>0.23</td>
<td>Putative New IS</td>
</tr>
</tbody>
</table>

The effects of DOASIS module compared to single OASIS on annotated NCBI genomes are depicted in Figure 6. The improvement in real IS discovery is obvious, illustrating the low quality and inadequacy of NCBI annotations for studying insertion sequences in bacterial genomes, and the improvements when using our pipeline. This chart shows too that a zigzag path in the annotation can oddly improve the detection of insertion sequences.

The prediction of real ISs is based on finding conserved regions (i.e., inverted repeats (IRs)) surrounded by transposase genes. Some ISs have been lost in AllTpase, for the fol-
lowing reason: when we suggested that all genes are transposases, OASIS found predicted ISs that consist of large sets of transposases surrounded by IR in their left and right boundaries. But when these predicted ISs have been verified using ISFinder database, we did not find any good match. Contrarily, in Zigzag methods, good matches have been found (real ISs), because many of these elements consist of one or two transposase genes flanked by IRs. These results are listed with detail in Table 4 using BASys annotation tools.

![Figure 6: NOASIS (NCBI annotation) versus DOASIS](image)

Table 4: BASys annotation using NOASIS and DOASIS

<table>
<thead>
<tr>
<th>BASys</th>
<th>Name</th>
<th>Genome</th>
<th>Normal Real IS</th>
<th>All Transpos</th>
<th>Zigzag Odd</th>
<th>Zigzag Even</th>
<th>(All_T/odd/even)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Real IS</td>
<td>Real IS</td>
<td>Real IS</td>
<td>Real IS</td>
<td>Best Real</td>
</tr>
<tr>
<td>PACS2</td>
<td>106896550</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>PAO1</td>
<td>110643304</td>
<td>9</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>UCRBP-PA14</td>
<td>110643575</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>PA7</td>
<td>15293466</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LEB58</td>
<td>213888746</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>M18</td>
<td>380050701</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NCNML2S1</td>
<td>380062973</td>
<td>15</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>BK3</td>
<td>929981410</td>
<td>8</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>B136-33</td>
<td>478476202</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>19BR</td>
<td>485462089</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>213BR</td>
<td>485462091</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>RP77</td>
<td>514407635</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>c1447m</td>
<td>542873856</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>PAC8X1</td>
<td>543879514</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>PAO1-VE2</td>
<td>553886202</td>
<td>8</td>
<td>6</td>
<td>9</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>PAO1-VE11</td>
<td>553895034</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>PA1R</td>
<td>558665962</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>PA7</td>
<td>558672313</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>MTB-1</td>
<td>564949884</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LEB51</td>
<td>568151185</td>
<td>5</td>
<td>14</td>
<td>13</td>
<td>8</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>SCV20265</td>
<td>568306739</td>
<td>5</td>
<td>14</td>
<td>13</td>
<td>8</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>PA38182</td>
<td>575870901</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>YL34</td>
<td>576902775</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

|            |            |            |            |            |            |            |            |
|            | 131         | 109         | 128         | 91          | 157        |            |            |

We can thus wonder if the source of a wrong prediction of real IS is due to a wrong coding
sequence prediction, or to functionality errors. Switching between NOASIS and DOASIS allows us to answer this question. We can conclude from Table 5 that (1) annotation errors are more frequent on NCBI, while Prokka annotates well the sequences related to ISs (see NOASIS columns), and that (2) both NCBI and Prokka have a better coding sequence prediction than BASys, at least when considering sequences involved in IS elements (see DOASIS columns and the correlation line). More precisely, the correlation is based on the number of predicted real IS elements between NOASIS and DOASIS.

Table 5: Correlation table for different annotation tools

<table>
<thead>
<tr>
<th></th>
<th>NCBI</th>
<th>BASys</th>
<th>Prokka</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOASIS</td>
<td>DOASIS</td>
<td>NOASIS</td>
</tr>
<tr>
<td>Number of Real IS</td>
<td>110</td>
<td>169</td>
<td>131</td>
</tr>
<tr>
<td>Correlations</td>
<td>0.3985580752</td>
<td>0.357346472</td>
<td>0.926355615</td>
</tr>
</tbody>
</table>

Prodigal has been studied separately, as it does not provide genes functionality. The number of Real ISs per genome returned by our pipeline using prodigal is given in Figure 7. As shown in Table 6, the quality of coding sequences predicted with prodigal compared with other annotation tools allows us to discover the best number of real ISs. In particular, we have improved a lot of results produced by OASIS and ISFinder on NCBI annotations, which is usually used in the literature that focuses on bacterial insertion sequences. Furthermore, this table illustrates a certain sensitivity of coding sequence prediction tools with functionality annotation capabilities to detect ISs in some specific genomes like PA7. Indeed we discovered, during other studies we realized on this set of Pseudomonas strains, that PA7 has a lot of specific genes, that is, which are not in the core genome of all Pseudomonases, which may explain such a sensitivity.

Figure 7: Real ISs found by our pipeline using Prodigal
### Table 6: Final comparison using our pipeline

<table>
<thead>
<tr>
<th>Name</th>
<th>NCBI DOASIS</th>
<th>NCBI DOASIS</th>
<th>BASys DOASIS</th>
<th>BASys DOASIS</th>
<th>Prokka DOASIS</th>
<th>Prokka DOASIS</th>
<th>Prodigal DOASIS</th>
<th>Prodigal DOASIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACS2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>c7447m</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>PAO1</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>12</td>
<td>PAO581</td>
<td>10</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>UCBPP-PA14</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>PAO1-VE2</td>
<td>10</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>PA7</td>
<td>15</td>
<td>0</td>
<td>14</td>
<td>18</td>
<td>PAO1-VE13</td>
<td>10</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>LESB58</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>PA1R</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>M18</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>PA1</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>NCGM2.S1</td>
<td>11</td>
<td>12</td>
<td>19</td>
<td>14</td>
<td>MTB-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DK2</td>
<td>12</td>
<td>11</td>
<td>13</td>
<td>17</td>
<td>LES431</td>
<td>6</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>B136-33</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>SCV20265</td>
<td>15</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>19BR</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>11</td>
<td>PA38182</td>
<td>7</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>213BR</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>YL84</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>RP73</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>Total IS</td>
<td>84</td>
<td>71</td>
<td>91</td>
</tr>
</tbody>
</table>

### 6 Conclusion

Insertion sequences of bacterial genomes are usually studied using OASIS and ISFinder on NCBI annotations. We have shown in this article that a pipeline can be designed to improve the accuracy of IS detection and classification by improving the coding sequence prediction stage, and by considering a priori each sequence as a transposase. The source code for this pipeline can be download from the link\(^3\). A comparison has been conducted on a set of *Pseudomonas aeruginosa*, showing an obvious improvement in the detection of insertion sequences for some particular configurations of our pipeline.

In future work, we intend to enlarge the number of coding sequence and functionality prediction tools and to merge all the Real IS results in order to improve again the accuracy of our pipeline. We will then focus on the impact of IS elements in *P.aeruginosa* evolution, comparing the phylogenetic tree of strains of this species with a phylogeny of their insertion sequences. Insertion events will then be investigated, and related to genomes rearrangements found in this collection of strains. We will finally enlarge our pipeline to eukariotic genomes and to other kind of transposable elements.

### References


\(^3\)http://members.femto-st.fr/christophe-guyeux/en/insertion-sequences


Towards Accurate Transcription Start Site Prediction: a modelling approach

Marko Djordjevic
Institute of Physiology and Biochemistry
Faculty of Biology, University of Belgrade
11000 Belgrade, Serbia
dmarko@bio.bg.ac.rs

Abstract: Promoter prediction in bacteria is a classical bioinformatics problem, where available methods for regulatory element detection exhibit a very high number of false positives. We here argue that accurate transcription start site (TSS) prediction is a complex problem, where available methods for sequence motif discovery are not in itself well adopted for solving the problem. We here instead propose that the problem requires integration of quantitative understanding of transcription initiation with careful description of promoter sequence specificity. We review evidence for this viewpoint based on our recent work, and discuss a current progress on accurate TSS detection on the example of sigma70 transcription start sites in *E. coli*.

1 Introduction

Bacterial RNA polymerase is a central enzyme in cell, and initiation of transcription by bacterial RNA polymerase is a major point in gene expression regulation. Core RNA polymerase cannot by itself initiate transcription, so a complex between RNA polymerase core and a σ factor, which is called RNA polymerase holoenzyme (RNAP) is formed. A major σ factor, which is responsible for transcription of housekeeping genes, is called σ70 in *E. coli* and σ in a number of other bacteria. The discussion here will concentrate on this major class of promoter elements [BN01].

Accurate recognition of transcription start sites (TSS) is a necessary first step in understanding transcription regulation. Accurate recognition of bacterial promoters is consequently considered a major problem in bioinformatics, particularly since TSS detection is an important ingredient for number of other bioinformatic applications (e.g. gene and operon prediction). Available methods for TSS search include both standard information-theory based weight matrix searches, and those based on more advanced computational approaches such as neural networks and support vector machines. These methods however show poor accuracy for TSS prediction, i.e. lead to a very high number of false positives [St02]. We here argue that, instead of developing different methods for processing the existing data within the motif search framework, solving the problem requires an integrative approach, which includes: i) quantitatively modelling transcription initiation, which allows calculating kinetic parameters of transcription initiation  ii) accurately describing sequence specificity of the promoter elements, so that
the bioinformatics description is consistent with available biophysical measurements iii) characterizing sequence elements outside of the canonical -10 and -35 box. In the text below we concentrate on promoter detection for sigma 70 class of promoters, which is a major promoter class that is responsible for transcription of housekeeping genes.

Our discussion will emphasize the following: i) accurately aligning promoter elements is highly non-trivial, so that the promoter specificity may not be accurately reflected by the available alignments ii) the promoter specificity is likely determined by additional sequence elements, which are located outside of the canonical -35 and -10 boxes iii) TSS predictions require accurately calculating kinetic parameters of transcription initiation. In addressing these issues we will highlight our recently published work [Dj03,Dj04], and also discuss some of our most recent results on this problem.

2 High degeneracy of promoter elements

Transcription initiation begins with RNAP binding to dsDNA, which is referred to as the closed complex formation [DZR05]. Subsequent to RNAP binding, the two strands of DNA are separated through thermal fluctuations that are facilitated by interactions of RNAP with ssDNA [DR06]. The opening of two DNA strands results in a formation of ~15bps long transcription bubble, which typically extends from -11 to +3 (where +1 corresponds to the transcription start site) [BS07]. After the open complex is formed, RNAP clears the promoter and enters the elongation, which leads to synthesis of RNA from DNA template [BN01].

The main elements that determine functional promoter are -35 element (\(-35^{\text{TTGACA}}\)\(^{\text{-30}}\), where the coordinates in the superscript are relative to the transcription start site), -10 element (\(-12^{\text{TATAAT}}\)\(^{\text{-7}}\)), the spacer between these two elements, and the extended -10 element (\(-15^{\text{TG}}\)\(^{\text{-14}}\)) [HH08]. Interactions of \(\sigma\)70 with dsDNA of -35 element, extended -10 element, and -12 base of -10 element result in the closed complex formation [MD09]. On the other hand, the downstream bases of -10 element (-11 to -7) interact with \(\sigma\)70 in ssDNA form [MD09], and are directly involved in the open complex formation.

Consequently to better relate involvement of different promoter elements with the kinetic steps of transcription initiation (the closed and the open complex formation), it was recently proposed that the region from -15 to -7 is reorganized in the following way [HH08]: Region from -15 to -12 is connected in a new element that is defined as -15 element; this element includes extended -10 element, the most upstream base in -10 element (base -12), and base -13 that is in-between. Consequently, -10 element is shortened for one base-pair (to the region -11 to -7), which we here refer to as the short -10 element. In this way -35 and -15 elements are directly related with \(\sigma\)70-dsDNA interactions, while short -10 element is directly related with \(\sigma\)70-ssDNA interactions.

The basic problem with accurate promoter detection is high degeneracy of the promoter elements (-35, -15 and -10 elements); in addition, variable distance between -35 and -10 element also contributes to the problem. This high degeneracy is illustrated in Table 1, where we show the aligned elements for several randomly selected promoters. For example, if we concentrate on -35 element, we see that the consensus sequence
'TTGACA' does not match any of the promoter instances in the table. Furthermore, only one instance has one mismatch from the consensus, most of the instances have two mismatches, while two of the instances have as much as four mismatches. In order to accommodate such high degeneracy, i.e. to correctly classify majority of the detected promoters, a low value of the detection threshold has to be imposed; this low threshold value than leads to a high number of false positives. One can artificially increase the detection threshold, which would decrease the false positives; however, another problem than emerges, i.e. a number of experimentally detected promoters are than wrongly classified. Consequently, the high degeneracy of the promoter elements, together with the relatively complex core promoter structure (several sequence elements with variable relative distances), is the main reason behind the low prediction accuracy of the available approaches.

<table>
<thead>
<tr>
<th>Promoter</th>
<th>-35</th>
<th>spacer</th>
<th>-15</th>
<th>short</th>
<th>-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>accAp</td>
<td>TTGCTA</td>
<td>17</td>
<td>AGGC</td>
<td>AAATT</td>
<td></td>
</tr>
<tr>
<td>accBp</td>
<td>TTGATT</td>
<td>17</td>
<td>GACC</td>
<td>AGTAT</td>
<td></td>
</tr>
<tr>
<td>accDp</td>
<td>TATCCA</td>
<td>19</td>
<td>TGTT</td>
<td>TTAAT</td>
<td></td>
</tr>
<tr>
<td>aceBp</td>
<td>TTGATT</td>
<td>16</td>
<td>GAGT</td>
<td>AGTCT</td>
<td></td>
</tr>
<tr>
<td>acnAp1</td>
<td>CTAACA</td>
<td>15</td>
<td>GCCT</td>
<td>TTATA</td>
<td></td>
</tr>
<tr>
<td>acnAp2</td>
<td>TCAAT</td>
<td>19</td>
<td>TGTT</td>
<td>ATCTT</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Examples of promoter sequence elements

3 Importance of the accurate promoter alignment

A necessary step in accurate TSS prediction is achieving a quantitative understanding of promoter specificity, i.e. accurately defining sequence elements that constitute bacterial promoter. However, aligning the promoter elements presents in-itself a highly non-trivial bioinformatic task due to both complex structure of bacterial promoter and degeneracy of the promoter elements (see above). A major problem with the existing collections of the promoter elements is due to the following: i) they are based on initial alignments of a small collection of promoter elements which were performed 'by eye' [WB10,HC11,RMC12,MZBM13] ii) accurate aligning of -35 element is complicated by both variable distance from -35 element and by a lower conservation of this element [HC11] iii) it is non-trivial to produce an alignment with sufficient accuracy for analyzing -15 element, given a weaker conservation of this element compared to both -10 and -35 elements [MZBM13].

Having these problems in mind, we recently performed a systematic 'de-novo' alignment of the promoter elements on a large collection of more than 300 experimentally confirmed σ70 TSS in E. coli [Dj04]. This alignment comes directly from experimentally determined TSS assembled in RegulonDB database [GSPSMSJWGL14]. For this we used Gibbs search algorithm for unsupervised alignment of the promoter elements, which we
consequently improved through supervised search by weight matrices defined through the Gibbs algorithm. The approach was to first align -10 element, and to consequently use this element as an anchor to align -35 element. Alignment of other relevant elements (spacer and -15 element) is directly determined once -10 element and -35 element are aligned. One should note that in addition to the canonical -10 and -35 elements our approach also allowed quantitating specificity of -15 element.

The unbiased alignment that we inferred shows notable differences with previously published alignments, as is discussed in more detail in [Dj04]. Furthermore, while our alignment is in accordance with biophysical data on $\sigma^{70}$-DNA interactions, the previously published alignments show notable discrepancies with the interaction data. We therefore next investigated to what extent the improved sequence alignment can in itself improve the prediction accuracy. To that end, we incorporated our improved alignment in the standard (weight matrix based) procedure for TSS detection, and compared the prediction accuracy with those resulting from the previous alignments. The comparison is shown in Figure 2, where we see that our alignment leads to as much as 50% reduction in the number of false positives. However, despite this significant reduction, one can see that the number of false positives is still very high; this then leads us to the next question that we consider, which is to what extent are kinetic effects important in transcription initiation.

![Figure 1: DET (Detection Error Tradeoff) curve, which presents comparison of the method based on our alignment, and the method in [RMC12] that uses the alignment from DPInteract database. The vertical axis presents the number of false positives, which is based on the number of correctly classified sequences in the testing set. The horizontal axis presents the number of false negatives, which is estimated based on the number of hits in the randomized intergenic regions. The blue line and the red line correspond, respectively, to our procedure and the procedure from [RMC12]. The figure adopted from (Djordjevic M, Djordjevic M, to be submitted).](image)

### 4 Importance of the kinetic effects

We next discuss another factor which may have a major impact on the accuracy of TSS predictions, which are kinetic effects in transcription initiation. As the first step of transcription initiation, RNAP reversibly binds to dsDNA of promoter elements, which is called the closed complex formation, and is described by the binding affinity $K_B$. This binding of RNAP leads to opening of the two DNA strands (promoter melting), so that a
transcription bubble is formed. This transcription bubble extends from the upstream edge of -10 element to about two bases downstream of the transcription start site, which roughly corresponds to positions -12 to +2 (+1 is transcription start site) [BS07]. The (inverse) time needed to form the transcription bubble (i.e. to open the two DNA strands) is described by the transition rate from closed to open complex \((k_f)\).

An extreme example of the kinetic effects in transcription initiation are poised promoters: These are locations in genome where RNAP binds with high binding affinity (high \(K_B\)), but has a low rate of transcription initiation due to a slow transition from closed to open complex (low \(k_f\)). It has been proposed that poised promoters may present a major problem for accurate TSS prediction [HC11,SF15]. This is particularly important, given the high number of false positives [St02,HC11,RMC12] that typically originate from computational TSS searches.

We consequently used the kinetic model of transcription initiation that we previously developed [DR06] in order to estimate the importance of the kinetic effects, which is shown in Figure 2 (for more details see [Dj03]).

![Figure 2: Log transcription rate \((\log(\varphi))\) vs. log binding affinity \((\log(K_B))\) for the intergenic segments. -10 element of lacUV5 promoter is substituted by all 6bp long segments from \textit{E. coli} intergenic regions. \(\log(K_B)\) and \(\log(\varphi)\) are calculated for each of these substitutions and shown, respectively, on the horizontal and the vertical axes on each of the panels. The horizontal and the vertical dashed lines correspond, respectively, to the transcription rate threshold and the binding affinity threshold. Green and red dots in the figure correspond to the strongly bound DNA sequences that are, respectively, functional promoters and poised promoters. Figure adapted from [Dj03].](image)

From Figure 2 we see that a significant fraction of the strongly bound sequences corresponds to poised promoters: In Figure 2, the blue dots mark strongly bound DNA segments that correspond to the functional promoters (i.e. to sequences that are above both the binding and the transcription activity threshold), while the red dots mark the sequences that correspond to the poised promoters (i.e. to sequences that are above the binding, but below the transcription activity threshold). One can see that a significant fraction of the strongly bound sequences (~30%) correspond to poised promoters. Such poised promoters can be falsely identified as targets by computational searches of core
promoters. Furthermore, our results from [Dj04] strongly suggest that the relevant kinetic parameter that characterizes functional promoters is the overall transcription activity; this is in contrast to some previous models which suggested that the relevant parameter is binding affinity of RNAP to dsDNA.

5 Conclusion

Accurate promoter prediction in bacteria is crucial not only as the first step in understanding transcription regulation, but also as an important ingredient in other bioinformatics applications such as gene and operon prediction. Despite being a classical bioinformatics problem, current methods for transcription start site prediction lead to a very high number of false positives. We here argue that transcription start site detection is a complex problem whose solution requires integrating several levels of knowledge. In particular, the discussion here strongly indicates that the following elements are necessary: i) accurately aligning promoter elements ii) characterizing sequences outside of canonical -35 and -10 boxes iii) estimating kinetic parameters of transcription initiation for a given sequence of interest, in particularly its transcription activity. Our current work is aimed at addressing these issues.

Acknowledgements

MD acknowledges support by Marie Curie International Reintegration Grant within the 7th European community Framework Programme (PIRG08-GA-2010-276996), by the Ministry of Education and Science of the Republic of Serbia under project number ON173052, and by the Swiss National Science foundation under SCOPES project number IZ73Z0_152297.

References


<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
<th>Source</th>
</tr>
</thead>
</table>
Posters

1. Daniel Arend, Jinbo Chen, Christian Colmsee, Steffen Flemming, Uwe Scholz and Matthias Lange
   The e!DAL Java API: Sharing and Citing Research Data in Life Sciences

2. Markus List, Ines Block, Marlene Lemvig Pedersen, Helle Christiansen, Steffen Schmidt, Mads Thomassen, Qihua Tan, Jan Baumbach, Jan Mollenhauer
   Microarray R-based Analysis of Complex Lysate Experiments with MIRACLE

3. Michael Dondrup, Christian Andreetta, Frank Nilsen and Inge Jonassen
   LiceBase - Building a model organism database and functional genomics tools for the sea lice research community

   BiBiCloud - a Cloud Computing Framework for Big Data Bioinformatics

5. Marc Bonin, Jekaterina Kokatjhuha, Sascha Johannes, Florian Heyl, Irene Ziska, Pascal Schendel, Karsten Mans, Biljana Smiljanovic, Till Sörensen and Thomas Häupl
   A Generic Online Database for Clinical Data Collection

   Bioinformatic Analysis of Regulatory Elements within the Promoter Region of the Cytochrome P450 gene, CYP6M2 in Anopheles gambiae

7. Florian Halbritter and Simon Tomlinson
   Identification of Functionally Related Genomic Elements by Similarity Clustering of ChiP-seq Profiles

8. Minou Nowrousian, Stefanie Traeger, Stefan Gesing and Daniel Schindler
   Comparative genomics and transcriptomics of filamentous fungi

9. Andreas Bremges, Tanja Woyke and Alex Sczyrba
   Metagenomic proxy assemblies of single cell genomes

10. Henrike Indrischek, Sonja Prohaska and Peter Stadler
    Reconstructing the gene phylogeny of arrestins: annotation of multi-exon genes

11. Corinna Ernst and Sven Rahmann
    A Density-based Approach for the Identification of Differentially Methylated Regions
12. Sabrina Ellenberger, Lisa Siegmund and Johannes Wöstemeyer  
A new task for HGT Calculator: Exploring relations between lifestyle and the frequency of horizontal gene transfer in Protozoa

13. Ulrike Löber, Ljerka Lah, Joachim Selbig and Stefanie Hartmann  
Genome assembly and annotation of the phytopathogenic fungus Ophiostoma bi-color

14. Veronika Dubinkina, Alexander Tyakht and Dmitry Alexeev  
Study of applicability limits for k-mer methods in metagenomic data analysis

15. Ben C Stöver and Kai F Müller  
LibrAlign - A Java library with powerful GUI components for multiple sequence alignment and attached data

16. Linda Sundermann, Sebastian Jünemann and Jens Stoye  
ChimP - Chimera Prediction with the Jumping Alignment Algorithm

17. Jinbo Chen, Christian Colmsee, Maria Esch, Matthias Klapperstück, Eva Grafahrend-Belau, Matthias Lange and Uwe Scholz  
LAILAPS: an integrative information retrieval platform for plant genomic resources

18. Mohammad Tagi Hasanzada and Laila Hassanzada  
Epigenetic alignment in HDAC acid amine families

19. Marcel Bargull, Kada Benadjemia, Benjamin Kramer, David Losch, Jens Quedenaud, Sven Schrinner, Jan Stricker, Dominik Köppl, Dominik Kopczynski, Henning Timm, Johannes Fischer and Sven Rahmann  
Variant Tolerant Read Mapping with Locality Sensitive Hashing

20. Dirk Willrodt, Florian Markowsky and Stefan Kurtz  
Unified Methods for Blast Searches on Compressed Sequence Databases

21. Guillaume Holley, Roland Wittler and Jens Stoye  
Bloom Filter Trie - a data structure for pan-genome storage

22. Peter Großmann  
The Bacterial Supercoiling Level - Modeling and Implications

23. Markus Lux, Barbara Hammer and Alexander Sczyrba  
An Efficient Pipeline for Automatic Grouping in Single-Cell Sequencing and Metagenomics

Deletions of chromosomal regulatory boundaries are associated with congenital disease
25. Guokun Zhang, Sebastian Schaaf and Ulrich Mansmann
   *ConDetec - Detecting and removing impurities in NGS data sets*

26. Paulo Pinto and Erich Bornberg-Bauer
   *Moulding genotype-phenotype maps, moulding evolution*

27. Nina Luhmann, Cedric Chauve, Jens Stoye and Roland Wittler
   *Scaffolding of Ancient Contigs and Ancestral Reconstruction in a Phylogenetic Framework*

28. Johannes Köster and Sven Rahmann
   *An algebra of single nucleotide variants*

29. Jan Grau, Jens Boch and Stefan Posch
   *Genome-wide TALEN off-target prediction*

30. Johannes Köster and Sven Rahmann
    *The ParallEl Alignment UTility (PEANUT) for GPU-based read mapping*

31. Hendrik Schäfer, Tim Schäfer, Joerg Ackermann, Claudia Döring, Sylvia Hartmann, Martin-Leo Hansmann and Ina Koch.
    *Hodgkin Lymphoma – From Image Analysis to Cell Graphs*

32. Sascha Daniel Krauß, Dennis Petersen, Daniel Niedieker, Erik Freier, Samir F. El-Mashtoly, Klaus Gerwert and Axel Mosig
    *Label-free Identification of Organelles through Colocalization of Raman and Fluorescence Microscopic Images*

33. Thomas Temme, Martin Schmuck, Axel Mosig and Ellen Fritsche
    *Novel computational approaches for High Content Image Analyses (HCA) of organoid neurosphere cultures in vitro.*

34. Jan Kölling, Karin Gorzolka, Karsten Niehaus and Tim W. Nattkemper
    *Spatio-temporal analysis of metabolite profiles during barley germination*

35. Chen Yang and Axel Mosig
    *An Algorithm for the Registration of Vibrational Microspectroscopic Images in Histopathological Stains*

36. Brijesh Singh Yadav, Pavan Kumar Yadav and Ajay Kumar
    *Structural and functional characterization of TIMP-3 protein in mammary tumor of Canis lupus familiaris*

37. Kutub Ashraf
    *An Immunoinformatics approach for designing epitope based vaccine strategy against S protein of mysterious new Middle East Respiratory Syndrome Coronavirus (MERS-CoV)*

38. Dominik Kopczynski and Sven Rahmann
    *An Online Peak Extraction Algorithm for Ion Mobility Spectrometry Data*
39. Sabrina Ellenberger, Anke Burmester and Johannes Wöstemeyer
The mtDNA of the mycoparasitic fusion parasite Parasitella parasitica: Sequence and comparative analysis

40. Shailendra Gupta, Ulf Schmitz, Xin Lai, Julio Vera and Olaf Wolkenhauer
Cooperative miRNA regulation in anticancer drug resistance - A computational approach

41. Jessica Kaufmann, Zhiqin Huang, Andrius Serva, Barbara Burwinkel, Peter Sinn, Andreas Schneeweiß, Peter Lichter, Marc Zapatka and Niels Grabe
Identification of patient-individual, immunologically addressable combinatorial cancer targets through a database driven RNA-seq bioinformatics pipeline

42. Carlus Deneke and Bernhard Renard
Machine learning for virulence prediction

43. Aarif Mohamed Nazeer Batcha and Ulrich Mansmann
Prevalence of potential mutations in colorectal cancer addressed genes in healthy population

44. Khalid Abnaof, Joao Dinis and Holger Fröhlich.
Using Consensus Clustering to Explore Biological Effect Similarities of Drug Treatments based on Integrated Biological Knowledge from Multiple Sources - An Example Study on HIV and Cancer

45. Rewati Tappu and Daniel Huson
Using MEGAN5 to analyse medical microbiome data

46. Marc Bonin, Irene Ziska, Jekaterina Kokatjuhha, Pascal Schendel, Karsten Mans, Biljana Smiljanovic, Till Sörensen, Andreas Grützkau, Bruno Stuhlmüller and Thomas Häupl.
Comparative Analysis between Rheumatoid Arthritis and Arthritis Model: Study of the Functional Components in Expression Profiles of Synovitis

47. Marc Bonin, Florian Heyl, Jekaterina Kokatjuhha, Sascha Johannes, Irene Ziska, Pascal Schendel, Karsten Mans, Biljana Smiljanovic, Till Sörensen, Bruno Stuhlmüller and Thomas Häupl
Identification of Co-Expression Networks of Inflammatory Response in Immune Cells

48. Marc Bonin, Pascal Schendel, Karsten Mans, Florian Heyl, Jekaterina Kokatjuhha, Sascha Johannes, Irene Ziska, Biljana Smiljanovic, Till Sörensen, Bruno Stuhlmüller
Prediction for Successful Treatment of Methotrexate in Rheumatoid Arthritis with mRNA and miRNA Microarray data

Mycoplasma salivarium-colonised oral squamous cell carcinoma
50. Manabu Sugii, Keisuke Kawano and Hiroshi Matsuno
   *Extension of Genetic Toggle Switch with a Mathematical Analysis for Artificial Genetic Circuits*

51. Jennifer Scheidel, Leonie Amstein, Jörg Ackermann, Liliana Schaefer, Ivan Dikic and Ina Koch
   *Mathematical model of antibacterial autophagy*

52. Leonie Amstein, Jennifer Scheidel, Jörg Ackermann, Simone Fulda, Ivan Dikic and Ina Koch
   *Mathematical model of TNFRI signal transduction*

53. Tatiana Shashkova
   *The Ecological Interactions Model of Human Intestinal Microbiota*

54. Oliver Philipp, Andrea Hamann, Heinz D. Osiewacz and Ina Koch
   *The autophagy interaction network of the fungal aging model Podospora anserina*

55. Stefan Schuster
   *On the history of the Michaelis-Menten equation*

56. Christian Tokarski, Sebastian Vlaic, Jana Schleicher, Reinhard Guthke and Stefan Schuster
   *Hepatic Response to Refeeding - From Ketone Bodies to Lipids*

57. Olga Popik, Björn Sommer, Ralf Hofestäd and Vladimir Ivanisenko
   *Pathway efficiency evaluation based on protein-protein interaction data and protein localizations*

58. Martin Lewinski, Christoph Brinkrolf and Ralf Hofestädt
   *Topological reconstruction and graph analysis of biological interaction networks*

59. Jana Schleicher, Stefan Schuster and Reinhard Guthke
   *Which mechanism determines the zonation of a fatty liver?*

60. Meik Kunz, Muhammad Naseem, Chunguang Liang and Thomas Dandekar
   *Probing the Unknowns in Cytokinin-Mediated Immune Defense in Arabidopsis with Systems Biology Approaches*

61. Thomas Wiebringhaus and Heinrich Brinck
   *Intermodular Connections Based On Highly Connected Regions In The Human Proteome*

62. Jan Grau and Jens Keilwagen
   *Discriminative modeling of dependencies in DNA binding sites*

63. Thomas Sütterlin, Kai Safferling and Niels Grabe
   *EPISIM: A user-friendly platform for multiscale multicellular modeling and simulation of biological systems*
64. **Rim Zaripov**
   *Approaches to harnessing the complexity of metaproteomic data*

65. **Heiko Giese, Joerg Ackermann, Ilka Wittig, Ulrich Brandt and Ina Koch**
   *Exemplary evaluation of complexome profiling data using NOVA*

66. **Ralf Eggeling, Andre Gohr, Jens Keilwagen, Michaela Mohr, Stefan Posch, Andrew D. Smith and Ivo Grosse**
   *On the Value of Intra-Motif Dependencies of Human Insulator Protein CTCF*

67. **Axel Rasche, Matthias Lienhard and Ralf Herwig**
   *ARH/ARH-seq: Discovery Tool for Differential Splicing in High-throughput Data*

68. **Cedric Saule and Robert Giegerich**
   *Observations on the Feasibility of Exact Pareto Optimization with Applications to RNA folding*

69. **Cristina Della Beffa and Frank Klawonn**
   *Certainty intervals for fold-changes of RNA expression*

70. **Jens Einloft, Joerg Ackermann and Ina Koch**
   *Topological properties of metabolic networks*

71. **Norma J. Wendel, Michael U. Höfer, Friedrich Felsenstein, Maria Rosenhauer, Jan Petersen and Antje Krause**
   *A systems biology approach to herbicide resistance in blackgrass*

72. **Franziska Metge and Christoph Dieterich**
   *New Insights to Protein Occupancy Profiling on mRNA*

73. **Heiner Klingenberg and Peter Meinicke**
   *Phylogenetic and functional classification of metatranscriptomic sequencing reads*

74. **Mohammad Tagi Hasanzada and Laila Hassanzada**
   *An Evolutionary relationship between MicroRNAs involved in MiRNA based Cellular Reprogramming*

75. **Pavankumar Videm, Dominic Rose, Fabrizio Costa and Rolf Backofen**
   *BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles.*

76. **Preethy Sasidharan Nair, Anju Philips, Harri Lähdesmäki and Irma Järvelä**
   *RNA-Seq to study music perception*

77. **Andrea Tanzer, Ronny Lorenz and Ivo Hofacker**
   *Detection of RNA G-Quadruplexes within and across Genomes*

78. **Abul Islam, Nuria Lopez-Bigas and Elizaveta Benevolenskaya**
   *Variations in KDM5A/JARID1A/RBP2 Isoform Specific Locations Reveals Contribution of Chromatin-Interacting PHD Domain in Protein Recruitment to Binding Sites*
79. **Vaibhav Sabale and Arun Ingale**
   Homology modeling and docking study of 3 oxoacyl (acyl carrier protein) synthase II protein of Neisseria meningitidis.

80. **Markus Weisser**
   Exchangeable HELM as a new standard for biomolecule representation

81. **Divya Dube, T. P. Singh and Punit Kaur**
   Pharmacophore Mapping, In Silico Screening and Molecular Docking to Identify Selective Trypanosoma brucei Pteridine Reductase Inhibitors

82. **Sabrina Ellenberger, Stefan Schuster and Johannes Wöstemeyer**
   Structure modelling, protein-ligand, and protein-protein interaction of sex pheromone processing dehydrogenases in Mucor-like fungi

83. **Clemens Thoelken**
   Xlinq: Identification and Quantification of Chemical Cross-Linked Peptides in Mass Spectrometry Data

84. **Christian Peikert, Friedel Drepper, Silke Oeljeklaus, Lennart Martens and Bettina Warscheid**
   PROVIS - a tool for the analysis and visualization of SILAC-based quantitative protein interaction data and beyond

85. **Björn Sommer, Benjamin Kormeier, Klaus Hippe, Nils Rothe, Philipp Unruh, Rudolf Warkentin and Pascal Witthus.**
   CELLMicrocosmos X - Cell Modeling at the Integrative Level

86. **Astrid Wachter, Thomas Oellerich, Jasmin Corso, Ekkehard Schütz, Annalen Bleckmann, Henning Urlaub and Tim Beissbarth.**
   High-Throughput Proteomic and Transcriptomic Data Integration based on MS/MS and RNA-Seq Data using Prior Pathway Knowledge

87. **Benedikt Brink, Stefan Albaum and Tim Nattkemper**
   Fusion - a new polyomics platform

88. **Marc Bonin, Sascha Johannes, Stephan Flemming, Andreas Grützkau, Florian Heyl, Irene Ziska, Pascal Schendel, Karsten Mans, Biljana Smiljanovic, Till Sörensen, Stefan Günther and Thomas Häupl.**
   Development of a Database for Combined Analysis of DNA Methylation and Transcriptional Profiles in Different Immune Cells

89. **Daniel Doerr, Jens Stoye, Sebastian Böcker and Katharina Jahn**
   Algorithms for Discovering Weak Common Intervals in Indeterminate Strings

90. **Alexandra Kanygina, Alexander Manolov, Dmitry Alexeev and Tatyana Grigoryeva**
   Proteogenomic analyses of oil sludge microbiota
GI-Edition Lecture Notes in Informatics

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Lan-guage to Information Systems, NLDB'2001.
P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.
P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
P-12 Martin Glinz, Günther Müller-Luchnat (Hrsg.): Modellierung 2002.
P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni (Hrsg.): The Changing Universities – The Role of Technology.
P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.
P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur ”Didaktik der Informatik” – Theorie, Praxis, Evaluation.
P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit.
P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB’2003: Natural Language Processing and Information Systems.
P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications.
P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures.
P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003
P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft
P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationsysteme – MobiS 2003
P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobiS 2004
P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
P-44 Jan von Knop, Wilhelm Havercamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services
P-45 Bernhard Rump, Wolfgang Hesse (Hrsg.): Modellierung 2004
P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven
P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbunden (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbunden (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration
P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government
P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung
P-61 Paul Müller, Reinhard Gotzheim, Jens B. Schmidt (Hrsg.): Kommunikation in verteilten Systemen
P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit
P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications
P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.) Datenbanksysteme in Business, Technologie und Web (BTW)
P-145 Knut Hinkelmann, Holger Wache (Eds.) WM2009: 5th Conference on Professional Knowledge Management
P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.) 9th International Conference on Innovative Internet Community Systems iICCS 2009
P-149 Paul Müller, Bernhard Neumair, Gabi Dreö Rodosek (Hrsg.) 2. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung
P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.) Software Engineering 2009 - Workshopband
P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.) PRIMIUM Process Innovation for Enterprise Software
P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.) Enterprise Modelling and Information Systems Architectures Proceedings of the 3rd Int’l Workshop EMISA 2009
P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.) Lernen im Digitalen Zeitalter DeLFi 2009 – Die 7. E-Learning Fachtagung Informatik
P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.) INFORMATIK 2009 Im Focus das Leben
P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.) BIOSIG 2009: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures
P-156 Bernhard Koerber (Hrsg.) Zukunft braucht Herkunft 25 Jahre »INFOS – Informatik und Schule«
P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.) German Conference on Bioinformatics 2009
P-158 W. Claupein, L. Theuvesen, A. Kämpf, M. Morgenstern (Hrsg.) Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft
P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.) Software Engineering 2010
P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.) Modellierung 2010
P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.) Vernetzte IT für einen effektiven Staat Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010
P-164 Arslan Brömme, Christoph Busch (Eds.) BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures
| P-186 | Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.) | 11th International Conference on Innovative Internet Community Systems (I2CS) |
| P-189 | Thomas, Marco (Hrsg.) | Informatik in Bildung und Beruf INFOS 2011 14. GI-Fachtagung Informatik und Schule |
| P-190 | Markus Nättgens, Oliver Thomas, Barbara Weber (Eds.) | Enterprise Modelling and Information Systems Architectures (EMISA 2011) |
| P-191 | Arslan Brömme, Christoph Busch (Eds.) | BIOSIG 2011 International Conference of the Biometrics Special Interest Group |
| P-192 | Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.) | INFORMATIK 2011 Informatik schafft Communities |
| P-193 | Wolfgang Lehner, Gunther Piller (Hrsg.) | IMDM 2011 |
| P-194 | M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.) | Informationstechnologie für eine nachhaltige Landbewirtschaftung Fokus Forstwirtschaft |
| P-196 | Arslan Brömme, Christoph Busch (Eds.) | BIOSIG 2012 Proceedings of the 11th International Conference of the Biometrics Special Interest Group |
| P-198 | Stefan Jähnichen, Axel Küpper, Sahin Albyar (Hrsg.) | Software Engineering 2012 Fachtagung des GI-Fachbereichs Softwaretechnik |
| P-199 | Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.) | Software Engineering 2012 Workshopband |
| P-200 | Gero Mühl, Jan Richling, Andreas Herkerodorf (Hrsg.) | ARCS 2012 Workshops |
| P-201 | Elmar J. Sinz Andy Schürr (Hrsg.) | Modellierung 2012 |
| P-202 | Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.) | MMS 2012:Mobile und Ubiquitäre Informationssysteme |
| P-203 | Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.) | 5. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung |
| P-204 | Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.) | 12th International Conference on Innovative Internet Community Systems (I2CS 2012) |
| P-205 | Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.) | 5th International Conference on Electronic Voting 2012 (EVOTE2012) Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC |
| P-206 | Stefanie Rinderle-Ma, Mathias Weske (Hrsg.) | EMISA 2012 Der Mensch im Zentrum der Modellierung |