Construction and Visualization of Dynamic Biological Networks: Benchmarking the Neo4J Graph Database

Lena Wiese[1][0000−0003−3515−9209], Chimi Wangmo[2], Lukas Steuernagel[3], Armin O. Schmitt[3,4], and Mehmet Gültas[3,4]

[1] Institute of Computer Science, University of Göttingen
Göttingen, Germany
wiese@cs.uni-goettingen.de
[2] Gyalpozhing College of Information Technology, Royal University of Bhutan
Thimphu, Bhutan
chimiwangmo.gcit@rub.edu.bt
[3] Breeding Informatics Group, Department of Animal Sciences, University of Göttingen
Göttingen, Germany
lukas.steuernagel@stud.uni-goettingen.de
[4] Center for Integrated Breeding Research (CiBreed), University of Göttingen
Göttingen, Germany
armin.schmitt@uni-goettingen.de, gueltas@cs.uni-goettingen.de

Abstract. Genome analysis is a major precondition for future advances in the life sciences. The complex organization of genome data and the interactions between genomic components can often be modeled and visualized in graph structures. In this paper we propose the integration of several data sets into a graph database. We study the aptness of the database system in terms of analysis and visualization of a genome regulatory network (GRN) by running a benchmark on it. Major advantages of using a database system are the modifiability of the data set, the immediate visualization of query results as well as built-in indexing and caching features.

1 Introduction

Genome analysis is a specific use case in the life sciences that has to handle large amounts of data that expose complex relationships. The size and number of genome data sets is increasing at a rapid pace [35]. Visualization of large scale data sets for exploration of various biological processes is essential to understand, e.g., the complex interplay between (bio-)chemical components or the molecular basis of relations among genes and transcription factors in regulatory networks [23]. Therefore, visualizing biological data is increasingly becoming a vital factor in the life sciences. On the one hand, it facilitates the explanation of the potential biological functions of processes in a cell-type, or the discovery of patterns as well as trends in the datasets [25]. On the other hand, visualization approaches

can help researchers to generate new hypotheses to extend their knowledge based on current informative experimental datasets and support the identification of new targets for future work [21].

Over the last decade, large efforts have been put into the visualization of biological data. For this purpose, several groups have published studies on a variety of methods and tools for e.g., statistical analysis, good layout algorithms, searching of clusters as well as data integration with well-known public repositories [1, 3, 8, 15, 18, 27, 28, 32] (for details see review [14]). Recently, by reviewing 146 state-of-the-art visualization techniques Kerren et al. [13] have published a comprehensive interactive online visualization tool, namely BioVis Explorer, which highlights for each technique the data specific type and its characteristic analysis function within systems biology.

A fundamental research aspect of systems biology is the inference of gene regulatory networks (GRN) from experimental data to discover dynamics of disease mechanisms and to understand complex genetic programs [26]. For this aim, various tools (e.g., GENeVis[3], FastMEDUSA [4], SynTReN [5], STARNET2 [10], ARACNe [19], GeneNetWeaver [27], Cytoscape [28], NetBioV [31], LegumeGRN [32]) for the reconstruction and visualization of GRNs have been developed over the past years and those tools are widely used by system and computational biologists. A comprehensive review about (dis-)advantages of these tools can be found in [14]. Kharumnuid et al. [14] have also discussed in their review that the large majority of these tools are implemented in Java and only a few of them have been written using PHP, R, PERL, Matlab or C++, indicating that the analysis of GRNs with those tools, in most cases, needs a *two-stage process*: In the *first* stage, experimental or publicly available data from databases such as FANTOM [17], Expression Atlas [24], RNA Seq Atlas [16], or The Cancer Genome Atlas (https://www.cancer.gov/), have to be prepared; in the *second* stage, network analysis and visualization with GRN tools can be performed. This second stage possibly involves different tools for analysis and for visualization. This requires both time and detailed knowledge of tools and databases.

To overcome this limitation of existing tools as well as to simplify the construction of GRNs, we propose in this study the usage of an integrated tool, namely Neo4J, that offers both analysis as well as visualization functionality. Neo4J which is implemented in Java is a very fast, scalable graph database platform which is particularly devised for the revelation of hidden interactions within highly connected data, like complex interplay within biological systems. Further, Neo4J provides the possibility to construct *dynamic* GRNs that can be constructed and modified at runtime by insertion or deletion of nodes/edges in a stepwise progression. We demonstrate in this study that the usage of a graph database could be favourable for analysis and visualization of biological data. Especially, focusing on the construction of GRNs, it has the following advantages:

- No two-stage process consisting of a data preparation phase and a subsequent analysis and visualization phase

- Built-in disk-memory communication to load only the data relevant for processing into main memory
- Reliability of the database system with respect to long-term storage of the data (as opposed to the management of CSV files in a file system)
- Advanced indexing and caching support by the database system to speed up data processing
- Immediate visualization of analysis results even under modifications of the data set

The article is organized as follows. Section 2 provides the necessary background on genome regulatory networks and the selection of data sets that we integrated in our study. Section 3 introduces the notion of graphs and properties of the applied graph database. Section 4 reports on the experiments with several workload queries that are applied for enhancer-promoter Interaction. Section 5 concludes this article with a discussion.

2 Data Integration

To demonstrate the usability of the Neo4J graph database for analysis and visualization of biological data in the field of life sciences, we construct GRNs based on known enhancer-promoter interactions (EPIs) and their shared regulatory processes by focusing on cooperative transcription factors (TFs). For this purpose, we first obtained biological data from different sources (FANTOM [17], UCSC genome browser [11] and PC-TraFF analysis server [21]) and then performed a mapping-based data integration process based on the following phases:

Phase 1: The information about pre-defined enhancer-promoter interactions (EPI) is obtained from the FANTOM database. FANTOM is the international research consortium for "Functional Annotation of the Mammalian Genome" that stores sets of biological data for mammalian primary cell types according to their active transcripts, transcription factors, promoters and enhancers. Using the *Human Transcribed Enhancer Atlas* in this database, we collected our benchmark data.

Phase 2: Using the UCSC genome browser, which stores a large collection of genome assemblies and annotation data, we obtained for each enhancer and promoter region (defined in Phase 1) the corresponding DNA sequences individually. It is important to note that while the sequences of enhancers are directly extracted based on their pre-defined regions, we used the annotated transcription start sites (TSS) of genes for the determination of promoter regions and extraction of their corresponding sequences (-300 base pairs to $+100$ base pairs relative to the TSS).

Phase 3: Applying the PC-TraFF analysis server to the sequences from Phase 2, we identified for each sequence a list of significant cooperative TF pairs. The PC-TraFF analysis server also provides for each TF cooperations:

- a significance score (*z-score*), which presents the strength of cooperation
- an annotation about the cooperativity of TFs—more precisely whether their physical interaction was experimentally confirmed or not. The information about their experimental validation has been obtained from TransCompel (release 2014.2) [12] and the BioGRID interaction database [6].

The data integration process for the combination of data from different sources is necessary to construct highly informative GRNs, which include complex interactions between the components of biological systems. One of the key players of these systems are the TFs which often have to form cooperative dimers in higher organisms for the effective regulation of gene expression and orchestration of distinct regulatory programs such as cell cycle, development or specificity [21, 29, 33]. The binding of TFs occurs in a specific combination within enhancer- and promoter regions and plays an important role in the mediation of chromatin looping, which enables enhancer-promoter interactions despite the long distances between them [2, 20, 22]. Today, it is well known that enhancers and promoters interact with each other in a highly selective manner through long-distance chromatin interactions to ensure coordinated cellular processes as well as cell type-specific gene expression [2, 20, 22]. However, it is still challenging for life scientists to understand how enhancers precisely select their target promoter(s) and which TFs facilitate such selection processes as well as interactions. To highlight such complex interactions between the elements of GRNs in a stepwise progression, Neo4J provides very effective graph database based solutions for the biological research community.

3 The Graph Database Neo4J

For datasets that lack a clear tabular structure and are of large size, data management in NoSQL databases might be more appropriate than mapping these datasets to a relational tabular format and managing them in a SQL database. Several non-relational data models and NoSQL databases—including graph data management—are surveyed in [34]. Graphs are a very versatile data model when links between entities are important. In this sense, a graph structure is also the most natural representation of a GRN.

Mathematically, a directed graph consists of a set V of nodes (or vertices) and a set E of edges. For any two nodes v_1 and v_2, a directed edge between these nodes is written as (v_1, v_2) where v_1 is the source node and v_2 is the target node. Graph databases often apply the so-called property graph data model. The property graph data model extends the notion of a directed graph by allowing key-value pairs (called "properties") to store information in the nodes and along the edges. Graph databases have been applied to several biomedical use cases in other studies: Previous versions of Neo4J have been used in a benchmark with just three queries by Have and Jensen [9] while Fiannaca et al. [7] present their BioGraphDB integration platform which is based on the OrientDB framework.

Neo4J (`https://neo4j.com/`) is one of the most widely used open source graph databases and has a profound community support. In Neo4J each edge

has a unique type (denoting the semantics of the edge relationship between the two attached nodes); each node can have one or more labels (denoting the type or types of the node in the data model). Neo4J offers a SQL-like query language called *Cypher*. Cypher provides "declarative" syntax that is easy to read. It has an ASCII art syntax visually representing nodes and relationships in the graph structure. Thus, the query pattern for "Find all the genes g to which at least one TFPair t binds" is `MATCH (g:Gene)<-[:binds]-(t:TFPair) RETURN g,t`. Here, `Gene` and `TFPair` are the two types for nodes and the query identifies the relationships labeled `binds` connecting any nodes of type `Gene` and `TFPair`. The resulting nodes and their relationships are immediately visualized in the Neo4J browser. A snippet of the result visualization is shown in Figure 1.

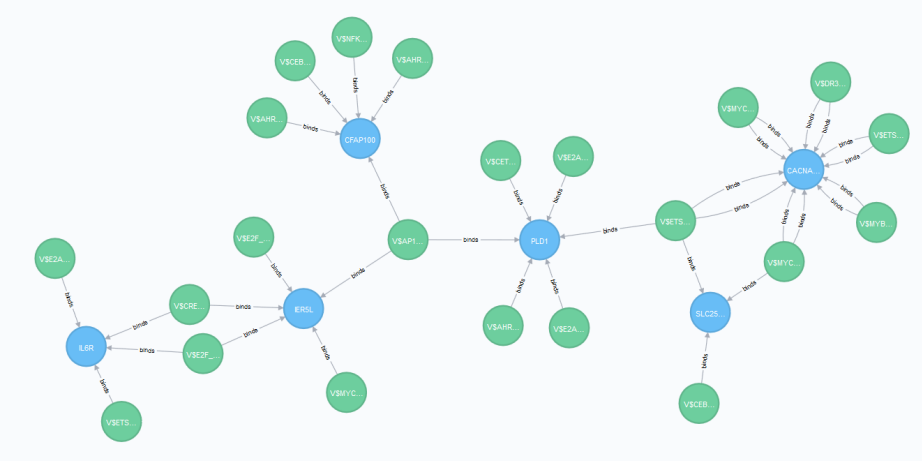

Fig. 1. A snippet of the result visualization of the sample query

Neo4j employs various caching mechanisms; as a result, once the query has been executed the following executions will use the nodes/relationships cache. The Neo4J page cache maintains data blocks in RAM for faster traversal by avoiding disk access. Moreover, the query plan cache helps reducing the computing time for parametrized queries that have already been executed before.

4 Benchmark

The benchmark was executed on a Linux PC running Ubuntu 16.04 LTS with the following specifications: an Intel CPU with 3.40 GHz and eight cores as well as 15.6 GB RAM. For our benchmark, we used Neo4J 3.4.3 Enterprise. We tested the analysis of our GRN data on a small data set and a larger data set.

4.1 Datasets

The input is provided as files in comma separated values (CSV) format. The files representing the genes (corresponding to the promoters), enhancers and pairs of transcription factors were parsed into Neo4j first. We generated three distinct types of nodes (namely, *Gene*, *Enhancer*, and *TFPair*) from them. Each *Gene* node has its *genename* as a property, while each *Enhancer* node contains a property called *enhancerID*; each *TFPair* has several properties: *pwm1*, *pwm2* (which denote the two cooperative transcription factors), the *name* (as a concatenation of the two represented transcription factors), as well as *KnownCompelPair* and *KnownBioGridPair* as properties (as described in Section 2, Phase 3).

Next, we created two types of relationships: *EPI* and *binds*. We extracted the *EPI* relationship between an enhancer and a promoter (located upstream of the specified gene); the EPI relationship represents the known interaction between an enhancer and promoter (as described in Section 2, Phase 1). The *binds* relationship links either a TFPair and an enhancer or a TFPair and a promoter. The relationship *binds* represents the fact that the pair binds to the promoter or enhancer in the order specified in the properties pwm1 and pwm2. Moreover, each *binds* relationship also has a property called *zscore* that denotes the strength of the binding (as described in Section 2, Phase 3).

The size of the small dataset in CSV format was 97.6 kB containing 1422 lines of text. The generated nodes included 11 genes, 619 TFPairs, and 15 enhancers; there were 19 EPI relationships and 757 binds relationships. We also tested a larger dataset of size 873 kB (with 16559 lines of text). There were 314 gene nodes, 3983 TFPair nodes, and 132 enhancer nodes. Furthermore, the numbers of relationships increased to 375 EPI relationships and 11747 binds relationships.

The datasets analyzed in this study and the cypher-commands used to load and analyze them with Neo4J are available under [30].

4.2 Queries

For both benchmark datasets, small and large, the same queries were run. The tests comprised two settings in order to consider the effects of the Neo4J cache:

- one test was conducted on cold boot and executed only once to avoid caching of the dataset;
- the other test was conducted after warming up the cache; in order to test for the real-world scenario, the queries have been run twenty times; then, their average was calculated to find the representative execution time.

The execution time represents not only the query run time on the database but includes the entire round-trip latency for visualizing the results and deserialization (streaming) of the result objects. We used the following test cases:

- Bulk data insertion
  - i1-3: Loading the CSV files (genes, enhancers, TFPairs)
  - c1-3: Assigning a uniqueness constraints to nodes

  - i4-6: Loading relationship data from CSV files (EPI and binds)
- Path queries
  - Q1a: For a given genename, find all enhancers interacting with that gene.
  - Q1b: For a genename set, find all enhancers interacting with the genes.
  - Q2a: For a given genename, find all TFPairs bound to that gene.
  - Q2b: Restrict to the known TFPairs with AND operator.
  - Q2c: Restrict to the known TFPairs with AND and OR operator
  - Q2d: Find the TFPairs of an enhancer that interact with a certain gene.
  - Q2e: Restrict to z-score larger than 4.
  - Q3a: For all genenames find all other genenames that are bound by at least one common TFPair.
  - Q3b: For a specific gene find all other genenames that are bound by at least one common TFPair.
  - Q3c: For a specific enhancerID find all other enhancerIDs that are bound by at least one common TFPair.
  - Q3d: For a specific enhancerID find genenames that are bound by at least one common TFPair.
  - Q4a: For a given enhancer ID (or a prefix of the ID), find all the TFPairs bound to the enhancer.
  - Q5a: For a given enhancerID, find all genes interacting with the enhancer.
  - Q6a: For a given genename, find all TFPairs bound to the gene.
  - Q6b: For a given genename, find all TFPairs bound to the gene restricting to those bindings with a high zscore.
  - Q7a: For a given TF find all TFPairs that contain the TF.
  - Q7b: For a given TF find the names of the two transcription factors in the TFPairs that contain the transcription factor.
- Statistical queries
  - G1a: Count the total number of TFPairs that one enhancer has in common with any other.
  - G1b: Count the TFPairs that two specific enhancers have in common.

4.3 Runtime Results

We analyzed the runtime results to assess the impact of dataset size and cache warming on our sample queries. Bulk loading data from CSV files into Neo4J is taking more time than performing any other queries as shown in Figure 2. The increased amount of nodes in the larger benchmark (insertion steps i1, i2 and i3) did not impact the runtime substantially. In contrast, the increased amount of relationships (insertion steps i4, i5 and i6) led to a significant runtime overhead.

The next executions that cover the cold-boot tests (without cache warming) are depicted in Figure 3. In this case, the runtime for Q2b, Q3c, Q3d, and Q5a was the same for both the small and large benchmark. Interestingly, the path queries Q1a, Q2a, Q2e Q6a, and Q6b, took on average 35% more execution time for the small benchmark than for the large benchmark which demonstrates a good off-the-shelf scalability of the graph database. Lastly, all the other queries were taking more time to execute for the large benchmarks as opposed to the

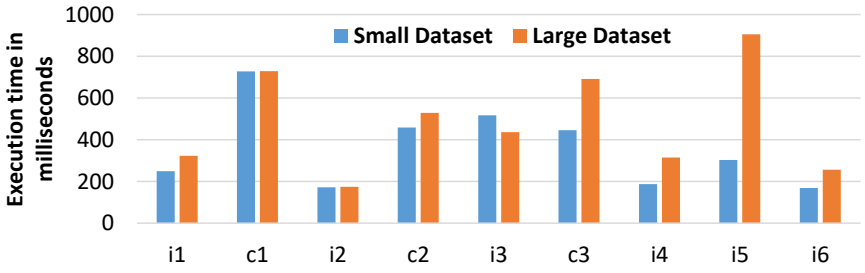

Fig. 2. Execution time for bulk data insertion steps

small one. This overhead can be explained by the fact that the returned amount of result nodes and result relationships is significantly larger for the large benchmark. In particular, the unrestricted query Q3a (which does not provide selection conditions for the queried Genes and TFPairs) could not be executed for the large dataset because the Neo4J browser crashed after 5 minutes.

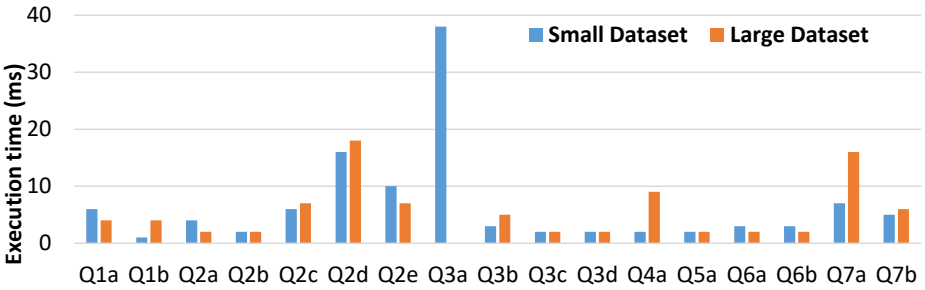

Fig. 3. Execution time of path queries on cold boot

After warming up the cache, the performance improved drastically: the execution time for processing queries decreased by about 64% on average for both the small and the large benchmark after warming up the system as compared to execution time for the cold boot case. Notably, for both datasets the execution times are nearly similar for most of the queries, which demonstrates the positive effect of cache warming. The unrestricted query Q3a remains the exceptional case where the database is not able to finish the execution on the large data set. For some queries, in particular Q3b, Q4a, Q7a and Q7b (taking more time to execute in the large benchmark than in small benchmark) the impact of the larger result sets in the large dataset remains noticeable even after cache warming.

Lastly, we tested the two COUNT queries G1a and G1b as sample queries for statistical analysis of the data sets. Here we observed a significant overhead for the larger benchmark: the first query—counting TFPairs only for each enhancer—took roughly 12 times longer for the larger benchmark (22.4 ms) than

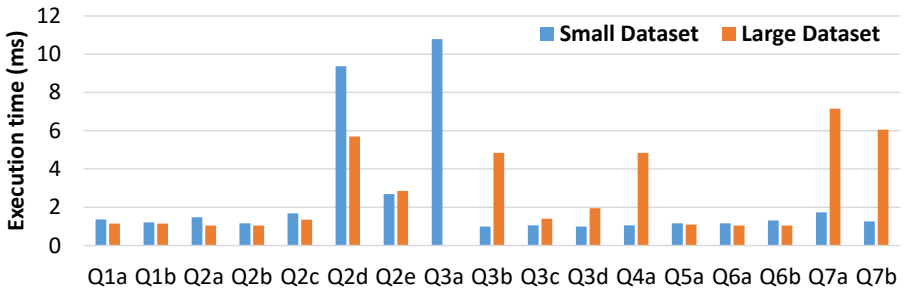

Fig. 4. Execution of path queries after cache warming

for the small benchmark (1.9 ms); more notably, the second query—counting TFPairs for each pair of enhancers—took roughly 19 times longer for the larger benchmark (37.7 ms) than for the small benchmark (1.95 ms).

5 Conclusion

In this paper we demonstrated that several advantages can be achieved for our use case of GRN analysis by loading our data into the Neo4J graph database and expressing our analysis queries in the human-readable query language Cypher. We presented our approach for integration of biological data from different sources. We proved scalability of query execution in the graph database by benchmarking the Neo4J graph database on a query workload using a small and a large data set and investigating the effect of cache warming on the performance.

The growing importance of visualization techniques is reflected in the still growing number of corresponding publications that are registered in the Pubmed database. In 2017 the proportion of visualization related articles has increased by a factor of 17 with respect to the average from the period of 1945 to 1974. This demonstrates the drastically increasing importance of visualization techniques "in the life sciences". Up until just a few years ago publications involving the keyword visualization were typically dealing with topics related to imaging techniques in the medical sciences. Only from the year 2012 on, a substantial number of publications that deal with visualization of big data has been published.

Making big data sets accessible to interpretation is one of the main challenges in Life science now and in the next years. Graph databases (in particular Neo4J) can be a powerful tool to aid researchers with the storage, the integration as well the analysis and visualization of biological, medical and healthcare data.

Acknowledgements. Chimi Wangmo participated in the preparation of this article while visiting the University of Göttingen with a Go International Plus scholarship by the Erasmus+ Key Action of the European Commission.

References

1. Albers, D., Dewey, C., Gleicher, M.: Sequence surveyor: Leveraging overview for scalable genomic alignment visualization. IEEE transactions on visualization and computer graphics 17(12), 2392–2401 (2011)
2. van Arensbergen, J., van Steensel, B., Bussemaker, H.J.: In search of the determinants of enhancer-promoter interaction specificity. Trends in Cell Biology 24(11), 695 – 702 (2014), http://www.sciencedirect.com/science/article/pii/S0962892414001184
3. Baker, C.A., Carpendale, M.S.T., Prusinkiewicz, P., Surette, M.G.: Genevis: visualization tools for genetic regulatory network dynamics. In: Visualization, 2002. VIS 2002. IEEE. pp. 243–250. IEEE (2002)
4. Bozdag, S., Li, A., Wuchty, S., Fine, H.A.: Fastmedusa: a parallelized tool to infer gene regulatory networks. Bioinformatics 26(14), 1792–1793 (2010)
5. Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B., Marchal, K.: Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. BMC Bioinformatics 7(1), 43 (Jan 2006), https://doi.org/10.1186/1471-2105-7-43
6. Chatraryamontri, A., Breitkreutz, B.J., Oughtred, R., Boucher, L., Heinicke, S., Chen, D., Stark, C., Breitkreutz, A., Kolas, N., O'Donnell, L., Reguly, T., Nixon, J., Ramage, L., Winter, A., Sellam, A., Chang, C., Hirschman, J., Theesfeld, C., Rust, J., Livstone, M.S., Dolinski, K., Tyers, M.: The BioGRID interaction database: 2015 update. Nucleic Acids Research (2014), http://nar.oxfordjournals.org/content/early/2014/11/26/nar.gku1204.abstract
7. Fiannaca, A., La Rosa, M., La Paglia, L., Messina, A., Urso, A.: Biographdb: a new graphdb collecting heterogeneous data for bioinformatics analysis. Proceedings of BIOTECHNO (2016)
8. Gomez, J., Garcia, L.J., Salazar, G.A., Villaveces, J., Gore, S., Garcia, A., Martin, M.J., Launay, G., Alcantara, R., del Toro, N., Dumousseau, M., Orchard, S., Velankar, S., Hermjakob, H., Zong, C., Ping, P., Corpas, M., Jimnez, R.C.: Biojs: an open source javascript framework for biological data visualization. Bioinformatics 29(8), 1103–1104 (2013), http://dx.doi.org/10.1093/bioinformatics/btt100
9. Have, C.T., Jensen, L.J.: Are graph databases ready for bioinformatics? Bioinformatics 29(24), 3107 (2013)
10. Jupiter, D., Chen, H., VanBuren, V.: STARNET2: a web-based tool for accelerating discovery of gene regulatory networks using microarray co-expression data. BMC bioinformatics 10(1), 332 (2009)
11. Karolchik, D., Hinrichs, A.S., Furey, T.S., Roskin, K.M., Sugnet, C.W., Haussler, D., Kent, W.J.: The UCSC Table Browser data retrieval tool. Nucleic Acids Research 32(suppl_1), D493–D496 (2004), http://dx.doi.org/10.1093/nar/gkh103
12. Kel-Margoulis, O., Kel, A., Reuter, I., Deineko, I., Wingender, E.: TRANSCompel: a database on composite regulatory elements in eukaryotic genes. Nucleic Acids Res 30, 332–334 (2002)
13. Kerren, A., Kucher, K., Li, Y.F., Schreiber, F.: Biovis explorer: A visual guide for biological data visualization techniques. PLOS ONE 12(11), 1–14 (11 2017), https://doi.org/10.1371/journal.pone.0187341
14. Kharumnuid, G., Roy, S.: Tools for in-silico reconstruction and visualization of gene regulatory networks (GRN). In: Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on. pp. 421–426. IEEE (2015)

15. Kirlew, P.W.: Life science data repositories in the publications of scientists and librarians. Issues in science and technology librarianship 65(April) (2011)
16. Krupp, M., Marquardt, J.U., Sahin, U., Galle, P.R., Castle, J., Teufel, A.: RNA-Seq Atlas – a reference database for gene expression profiling in normal tissue by next-generation sequencing. Bioinformatics 28(8), 1184–1185 (2012), http://dx.doi.org/10.1093/bioinformatics/bts084
17. Lizio, M., Harshbarger, J., Shimoji, H., Severin, J., Kasukawa, T., Sahin, S., Abugessaisa, I., Fukuda, S., Hori, F., Ishikawa-Kato, S., Mungall, C.J., Arner, E., Baillie, J.K., Bertin, N., Bono, H., de Hoon, M., Diehl, A.D., Dimont, E., Freeman, T.C., Fujieda, K., Hide, W., Kaliyaperumal, R., Katayama, T., Lassmann, T., Meehan, T.F., Nishikata, K., Ono, H., Rehli, M., Sandelin, A., Schultes, E.A., 't Hoen, P.A., Tatum, Z., Thompson, M., Toyoda, T., Wright, D.W., Daub, C.O., Itoh, M., Carninci, P., Hayashizaki, Y., Forrest, A.R., Kawaji, H.: Gateways to the FANTOM5 promoter level mammalian expression atlas. Genome Biology 16(1), 22 (Jan 2015), https://doi.org/10.1186/s13059-014-0560-6
18. Longabaugh, W.J., Davidson, E.H., Bolouri, H.: Visualization, documentation, analysis, and communication of large-scale gene regulatory networks. Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms 1789(4), 363 – 374 (2009), http://www.sciencedirect.com/science/article/pii/S1874939908001624
19. Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D., Califano, A.: Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(1), S7 (Mar 2006), https://doi.org/10.1186/1471-2105-7-S1-S7
20. Matharu, N., Ahituv, N.: Minor loops in major folds: Enhancer-promoter looping, chromatin restructuring, and their association with transcriptional regulation and disease. PLOS Genetics 11(12), 1–14 (12 2015), https://doi.org/10.1371/journal.pgen.1005640
21. Meckbach, C., Tacke, R., Hua, X., Waack, S., Wingender, E., Gültas, M.: PC-TraFF: identification of potentially collaborating transcription factors using pointwise mutual information. BMC Bioinformatics 16(1), 400 (Dec 2015), https://doi.org/10.1186/s12859-015-0827-2
22. Mora, A., Sandve, G.K., Gabrielsen, O.S., Eskeland, R.: In the loop: promoter-enhancer interactions and bioinformatics. Briefings in Bioinformatics 17(6), 980–995 (2016), http://dx.doi.org/10.1093/bib/bbv097
23. O'Donoghue, S.I., Gavin, A.C., Gehlenborg, N., Goodsell, D.S., Hériché, J.K., Nielsen, C.B., North, C., Olson, A.J., Procter, J.B., Shattuck, D.W., et al.: Visualizing biological data – now and in the future. Nature methods 7(3), S2 (2010)
24. Petryszak, R., Keays, M., Tang, Y.A., Fonseca, N.A., Barrera, E., Burdett, T., Füllgrabe, A., Fuentes, A.M.P., Jupp, S., Koskinen, S., et al.: Expression Atlas update – an integrated database of gene and protein expression in humans, animals and plants. Nucleic acids research 44(D1), D746–D752 (2015)
25. Ren, J., Lu, J., Wang, L., Chen, D.: Data visualization in bioinformatics. Advances in Information Sciences and Service Sciences 4(22) (2012)
26. Roy, S., Bhattacharyya, D.K., Kalita, J.K.: Reconstruction of gene co-expression network from microarray data using local expression patterns. BMC bioinformatics 15(7), S10 (2014)
27. Schaffter, T., Marbach, D., Floreano, D.: Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics 27(16), 2263–2270 (2011)

28. Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.L., Ideker, T.: Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3), 431–432 (2010)
29. Sonawane, A.R., Platig, J., Fagny, M., Chen, C.Y., Paulson, J.N., Lopes-Ramos, C.M., DeMeo, D.L., Quackenbush, J., Glass, K., Kuijjer, M.L.: Understanding tissue-specific gene regulation. Cell reports 21(4), 1077–1088 (2017)
30. Steuernagel, L., Wiese, L., Gültas, M.: Repository visualization of dynamic biological networks, `https://github.com/azifiDils/Visualization-of-DynamicBiological-Networks-`
31. Tripathi, S., Dehmer, M., Emmert-Streib, F.: NetBioV: an R package for visualizing large network data in biology and medicine. Bioinformatics 30(19), 2834–2836 (2014)
32. Wang, M., Verdier, J., Benedito, V.A., Tang, Y., Murray, J.D., Ge, Y., Becker, J.D., Carvalho, H., Rogers, C., Udvardi, M., et al.: LegumeGRN: a gene regulatory network prediction server for functional and comparative studies. PloS one 8(7), e67434 (2013)
33. Whitfield, T.W., Wang, J., Collins, P.J., Partridge, E.C., Aldred, S.F., Trinklein, N.D., Myers, R.M., Weng, Z.: Functional analysis of transcription factor binding sites in human promoters. Genome Biology 13(9), R50 (Sep 2012), `https://doi.org/10.1186/gb-2012-13-9-r50`
34. Wiese, L.: Advanced Data Management for SQL, NoSQL, Cloud and Distributed Databases. DeGruyter/Oldenbourg (2015)
35. Wiese, L., Schmitt, A.O., Gültas, M.: Big data technologies for DNA sequencing. In: Sakr, S., Zomaya, A. (eds.) Encyclopedia of Big Data Technologies. Springer (2018)