Computational Inference of DNA Folding Principles: From Data Management to Machine Learning

Luca Nanni

Abstract DNA is the molecular basis of life and would total about three meters if linearly untangled. To fit in the cell nucleus at the micrometer scale, DNA has, therefore, to fold itself into several layers of hierarchical structures, which are thought to be associated with functional compartmentalization of genomic features like genes and their regulatory elements. For this reason, understanding the mechanisms of genome folding is a major biological research problem. Studying chromatin conformation requires high computational resources and complex data analyses pipelines. In this chapter, we first present the PyGMQL software for interactive and scalable data exploration for genomic data. PyGMQL allows the user to inspect genomic datasets and design complex analysis pipelines. The software presents itself as a easy-to-use Python library and interacts seamlessly with other data analysis packages. We then use the software for the study of chromatin conformation data. We focus on the epigenetic determinants of Topologically Associating Domains (TADs), which are region of high self chromatin interaction. The results of this study highlight the existence of a “grammar of genome folding” which dictates the formation of TADs and boundaries, which is based on the CTCF insulator protein. Finally we focus on the relationship between chromatin conformation and gene expression, designing a graph representation learning model for the prediction of gene co-expression from gene topological features obtained from chromatin conformation data. We demonstrate a correlation between chromatin topology and co-expression, shedding a new light on this debated topic and providing a novel computational framework for the study of co-expression networks.

1 Introduction

DNA is the molecular basis of life. It stores all the information required for life to reproduce, sustain itself and adapt to the environment. The human genome con-

L. Nanni (✉)

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy

e-mail: luca.nanni@polimi.it; luca.nanni@unil.ch

Department of Computational Biology, Université de Lausanne, Génopode, 1015 Lausanne, Switzerland

© The Author(s) 2022

L. Piroddi (ed.), *Special Topics in Information Technology*,

PoliMI SpringerBriefs, https://doi.org/10.1007/978-3-030-85918-3_7

sists of approximately three billion DNA base pairs [1, 2], which would total about three meters if linearly untangled. Therefore, to fit in the cell nucleus, which has an approximate diameter of six micrometers, DNA has to fold itself several times.

The study of DNA folding is a current biological research frontier and only recently it was possible to reliably probe its characteristics. A major technological milestone, which enabled for the first time to see the whole genome three-dimensional organization, was the invention of the Hi-C technique [3], which maps 3D contacts between all genomic locations. For the first time it was possible to reveal that genomes fold into high-level structures called Topologically Associating Domains (TADs) [4], which are genomic regions of high self-interaction and low interactions across them. TADs are of great biological relevance, since they are speculated to be associated with functional compartmentalization of genomic features like genes and their regulatory elements. For this reason, understanding the mechanisms of TADs and chromatin loops formation is a major biological research problem.

Studying chromatin conformation and Hi-C data requires high computational resources and complex data analyses pipelines. This is due to both the intrinsic noise present in the data and to the big size of the experimental datasets [5]. Hi-C experiments can produce billions of reads and their storage can require several terabytes. For this reason, it is necessary to develop scalable and interactive software for the exploration of big genomic datasets like Hi-C. In addition, studying Hi-C data usually requires the integration of other heterogeneous data sources (like, for example, ChIP-seq and RNA-seq), which implies that any software pipeline designed for complex biological analyses on chromatin conformation has to be able to work with multiple data formats as well as be able to easily integrate and manipulate them.

In the first part of this chapter, we present the design and implementation of a software for interactive and scalable data exploration for genomic data. This system, named PyGMQL, is a easy-to-use Python extension of the GMQL big genomic data engine, developed by the Genomic Computing team at Politecnico di Milano. PyGMQL is a Python package which enables the user to build complex genomic pipelines by relying on the Spark big data engine. Using a carefully designed set of functions, it allows the user to inspect and manipulate arbitrarily big genomic datasets and design the flow of execution of the program. Once the researcher has defined the complete data analysis pipeline, it can then be fully executed on the complete dataset. It also allows the loading and manipulation of heterogeneous data, coming from different experimental procedures, through the adoption of the Genomic Data Model (GDM), developed initially for the GMQL system. We demonstrate the qualities and performance of this software through a series of data analysis scenarios, showing its usability, requirements and execution times. PyGMQL is extremely versatile and can be used both a downstream analysis data analysis tool and as a development framework on top of which bioinformaticians can build new specialized software tools.

In the second part, we apply this software stack to the study of chromatin conformation data. We focus on the study of the epigenetic drivers and determinants of TADs, and we explore the relationship between the positions and motif orientations of CTCF binding sites. CTCF is a structural protein known to be associated with

chromatin looping [6], but the underlying rules determining TAD topologies have never been highlighted. We propose a set of spatial rules, having as elements the binding sites of CTCF, which correlate with TADs and their topological characteristics. We argue that genome conformation can be explained by a set of “grammatical rules” acted out by CTCF.

We finally inspect the relationship between chromatin conformation and gene expression. We ask to what extent chromatin networks determined by Hi-C can explain co-expression relations between genes, both intra- and inter-chromosomically. We model this research question as a machine learning problem, designing a graph representation learning model for the encoding of gene chromatin topological features. The learnt gene embeddings are used then as inputs to a Random Forest classifier. We demonstrate that this model outperforms a set of baselines based on previous works.

2 Interactive and Scalable Data Analysis for Genomics

The analysis of biological data is conventionally divided into three macro-steps, depending on the data processing and the biological question being asked and answered. This categorization is particularly important in the case of next-generation sequencing data.

- **Primary data analysis:** The first step in the analysis of biological data is usually the production and quality assessment of the raw data generated from the sequencing machine. It produces the raw nucleotide sequences for each read obtained from the machine.
- **Secondary data analysis:** Reads are then filtered and *aligned* to a *reference genome*, which gives the researcher the information about the *position* of a sequencing read in the target genome.
- **Tertiary data analysis:** Researchers then deepen into the biological problem they are studying and ask high-level questions about the data. This is the most important and case-specific step of biological data analysis and it usually requires custom pipelines and result evaluation, as well as the integration of different data sources.

2.1 The Genometric Query Language and Its Ecosystem

The Genomic Computing (GeCo) team at Politecnico di Milano decided to address these issue of *managing, extracting and analysing* tertiary biological data by proposing a paradigm shift in genomic data management [7].

The *Genomic Data Model* (GDM) represents genomic information as the combination of *region data* and *metadata* [8]. The *Genometric Query Language* (GMQL) [7, 9] was designed to answer complex biological questions using a declarative query

language. GMQL queries are compiled into a *computation graph*. This enables the implementation of different *backends* of the GMQL system [9]. GMQL has support for Spark [10], Flink [11] and SciDB [12] implementations. The GMQL repository [13] hosts data coming from The Cancer Genome Atlas [14], Roadmap Epigenomics [15], ENCODE [16] and others. The main interface of the GMQL system is through a *web application*, where the user can browse through the publicly available datasets in the GMQL repository, write and compile genomic queries through the *query editor*, and inspect the results of queries in his/her private storage.

2.2 PyGMQL: Scalable Programmatic Data Analysis of Genomic Data

¹ GMQL operates in a batch fashion, which means that the scientists has to write the query, run it and wait for the outcome. This approach does not fit very well the data exploration paradigm. This is due the fact that, in particular at the beginning of a study, scientists do not know the specific research question they want to ask to their data. This approach requires a shift of our framework towards *interactive* computation, where parameters are learnt during the exploration process through trial and error.

With the aim of solving this issue, we designed and implemented PyGMQL [17, 18], a Python library which embeds the GMQL engine.

The library adopts a client-server architecture, where the Python front-end exposes to the user all the dataset manipulation functions and utilities, while a Scala back-end implements all the query operators. As depicted in Fig. 1a, the back-end relies on the implementation of GMQL on Spark.

PyGMQL offers a set of methods which wrap and extend the GMQL language operators. PyGMQL offers methods to manipulate genomic datasets as variables, each associated to structures called *GMQLDataset*. The library stores all the operations performed on a variable through a GMQL directed acyclic graph (DAG) (see [9]). This design enables the back-end to apply query optimizations to the DAG structure [19]. Conforming to the *lazy execution* model of Spark, no actual operation is executed until the user explicitly triggers the computation.

PyGMQL can access data stored in the local machine, like most of the Python libraries. In addition, PyGMQL can also interface with an external GMQL system to interact with his/her private datasets or the public repository. Therefore, queries in PyGMQL can be composed of genomic operations acting both on local and remote datasets. The library can also “outsource” the query computation to an external GMQL service, and then download the result to the user machine (Fig. 1b). The location of the datasets used during the query is therefore orthogonal with respect to the mode of execution. The library keeps tracks of the used datasets and their dependen-

¹ This section is adapted from [17].

Fig. 1 **a** Schematic representation of the software components of PyGMQL. In the front-end, the GMQLDataset is a data structure associated with a query. The GDataframe stores the query result and enables in-memory manipulation of the data. The RemoteManager module is used for message interchange between the package and an external GMQL service. The back-end is implemented in Spark. **b** Deployment modes and executor options of the library. Figure adapted from [17]

cies during the whole Python program execution, minimizing the data transmission between local and remote systems.

We demonstrated the flexibility of the PyGMQL library through three data analysis workflows, available in the PyGMQL GitHub repository.² For a progressive introduction to PyGMQL usage, the applications are increasingly complex both for what concerns the biological analysis and the data size. We also provide a performance evaluation on increasingly large datasets of a significantly complex genomic query, highlighting the scalability of the system [17].

3 The Grammar of Genome Folding

³ We then decided to apply our novel software stack to a concrete biological research problem, having as focus the study of the mechanisms of chromatin folding.

The mechanisms underlying the storage and packing of DNA is a current knowledge frontier in biology. Thanks to the Hi-C technique it was possible to identify genomic regions of having an high level of self-interaction, which were named Topologically Associating Domains (TADs) [4]. TADs are divided by boundaries that are detected as positions in the genome where there is a sharp break from preferential

² <https://github.com/DEIB-GECO/PyGMQL>.

³ This section is adapted from [20].

Fig. 2 **a** Enrichment of the four CTCF pattern classes around highly conserved TAD boundaries in 7 cell types. **b** Enrichment of the same classes along the length of TADs, as a function of percentage of TAD size. **c** Enrichment of the same classes around the center of TADs. **d** Schematic representation of the “grammatical rules” induced by CTCF and its orientation, and their correlation with TAD features. Figure adapted from [20]

left-ward interactions to preferential right-ward interactions [4, 21]. Scientists also tried to understand the underlying factors determining chromatin topological organization. It was shown that the CTCF protein is a fundamental actor in this context. The CTCF protein binds to an asymmetric motif on the DNA [22]. This means that it is possible to find these motifs in two possible *orientations* ($>$ and $<$). It has been shown that convergent CTCF binding sites ($><$) are located at the two extremities of long range interacting DNA regions (loops), explaining multiple features of Hi-C data sets [6, 23].

With the aim to reconcile the notion of chromatin loops and TADs, we created a classification scheme for sets of more than two adjacent CTCF sites. Each cluster is classified based on the relative orientation of the CTCF binding sites composing it [20]. We identified four categories: *same* ($>>>$, $<<<$), *convergent* ($>><$, $><<$), *divergent* ($<>>$, $<<>$) and *convergent-divergent* ($><>$, $<><$). We therefore analysed the distributions of the sizes of CTCF clusters the human genome, revealing that, at length scales ranging from 5 to 100 kb, divergent CTCF site clusters are enriched, while convergent CTCF site clusters are depleted. This suggests that divergent CTCF sites code for TAD boundaries (Fig. 2a, b) and that convergent CTCF sites are involved in the definition of left and right TAD sections (Fig. 2c). We validated this orientation-based grammar as a function of CTCF site strength and TAD boundary strength (Fig. 2a).

These results suggest the presence of a linear “grammar” dictating the relative orientations of CTCF binding sites motifs, and that these simple rules play an impor-

Fig. 3 Schematic representation of the workflow. We generate a Gene Chromatin Network by summarizing Hi-C information of genes and their neighborhood for each gene (1). We then produce a reduced vector representations of genes through network embedding techniques (2). Finally, we take the combined pairs of gene vectors as input for a Random Forest classifier, trained on a subset of the gene co-expression network (3). Figure adapted from [30]

tant role in the definition of key chromatin conformation features, like TADs, their inside and their boundaries (Fig. 2d).

4 Chromatin Conformation and Gene Expression

A key element left out from our previous analysis of chromatin conformation is the role of genes. Gene expression is the key biological mechanism which produces phenotypic differences across cell types, tissues and biological conditions. Gene expression data can be summarized through the computation of co-expression between pairs of genes, building therefore a *co-expression network*.

The relationship between *gene co-expression* and chromatin conformation is of great biological interest [24–26]. Given the high complexity of Hi-C data and the difficult definition of gene coexpression networks [27], the development of proper computational tools to investigate such relationship is rapidly gaining the interest of researchers. One of the most fascinating questions in this context is how chromatin topology correlates with gene coexpression and which physical interaction patterns are most predictive of coexpression relationships [28, 29].

We explored the relationship between chromatin conformation and gene expression using a predictive modeling approach. Specifically, we designed a model to predict co-expression between two genes from the physical set of interactions derived from an Hi-C experiment (see Fig. 3). We used a representation learning approach [31] for embedding the topological features of genes. In our work, the features of the nodes are learnt by solving an optimization problem, which defines the embedding strategy of the physical interaction network extracted from Hi-C data. Therefore, the proper choice of the optimization method is critical.

We compared two different node embedding strategies. The first method is based on Matrix Factorization [32], while the second exploits a random walk procedure to

Fig. 4 Accuracy measure across the 22 single-chromosome networks of the proposed gene embedding strategies and comparison with baselines. Each box plot is derived from the accuracy measures for each cross validation fold and for each chromosome. Figure adapted from [30]

find similar embeddings for genes in the same neighborhood [33]. We then used the learnt embeddings to train a non-linear classifier, based on Random Forest [34]. We compared the performances between the two embedding strategies and against a set of baselines: the first one is based on the computation of a set of predefined measures for each gene/node in the Hi-C network to be used as input of the classifier [24]; the second is a simple distance-based predictor using only the linear distance between genes in the genome as input.

We validated our models on a comprehensive collection of datasets, where Hi-C data and matching gene expression is available, showing consistent results across conditions, cell line and tissues. Our results (see Fig. 4) show that both our embedding methods outperform the considered baselines. This finding shows that it is indeed possible to encode the topological properties of genes, and to use them to predict their co-expression. It is also important to point out that, despite the significant improvement in prediction performance of our proposed methods, we were able to predict correctly only a portion of the co-expression relationships. A possible interpretation of this result is that chromatin conformation and gene expression are linked phenomena, but they also maintain a degree of independence: the gene expression machinery is a very complex system, influenced by several factors, like histone modifications and DNA methylation, together with chromatin conformation.

References

1. International Human Genome Sequencing Consortium et al. Initial sequencing and analysis of the human genome. *Nature*, **409**(6822), 860 (2001)
2. J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt et al., The sequence of the human genome. *Science* **291**(5507), 1304–1351 (2001)
3. E. Lieberman-Aiden, N.L. Van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B.R. Lajoie, P.J. Sabo, M.O. Dorschner et al., Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**(5950), 289–293 (2009)
4. J.R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J.S. Liu, B. Ren, Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485**(7398), 376 (2012)

5. B.R. Lajoie, J. Dekker, N. Kaplan, The hitchhiker's guide to hi-c analysis: practical guidelines. *Methods* **72**, 65–75 (2015)
6. A.L. Sanborn, S.S.P. Rao, S.-C. Huang, N.C. Durand, M.H. Huntley, A.I. Jewett, I.D. Bochkov, D. Chinnappan, A. Cutkosky, J. Li et al., Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. *Proc. Nat. Acad. Sci.* **112**(47), E6456–E6465 (2015)
7. M. Masseroli et al., Genometric query language: a novel approach to large-scale genomic data management. *Bioinformatics* **31**(12), 1881–1888 (2015)
8. M. Masseroli et al., Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying. *Methods* **111**, 3–11 (2016)
9. M. Masseroli et al. Processing of big heterogeneous genomic datasets for tertiary analysis of next generation sequencing data. *Bioinformatics*, pp. 688 (2018)
10. M. Zaharia et al., Apache spark: a unified engine for big data processing. *Commun. ACM* **59**(11), 56–65 (2016)
11. A. Katsifodimos, S. Schelter. Apache flink: stream analytics at scale, in *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*. IEEE (2016), pp. 193–193
12. M. Stonebraker, P. Brown, D. Zhang, J. Becla, Scidb: a database management system for applications with complex analytics. *Comput. Sci. Eng.* **15**(3), 54–62 (2013)
13. A. Bernasconi, A. Canakoglu, M. Masseroli, S. Ceri, *Meta-Base: A Novel Architecture for Large-Scale Genomic Metadata Integration* (IEEE/ACM Trans. Comput. Biol. Bioinf, 2020)
14. J.N. Weinstein, E.A. Collisson, G.B. Mills, K.R. Mills Shaw, B.A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J.M. Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, **45**(10), 1113 (2013)
15. A. Kundaje, W. Meuleman, J. Ernst, M. Bilenky, A. Yen, A. Heravi-Moussavi, P. Kheradpour, Z. Zhang, J. Wang, M.I.J. Ziller et al. Integrative analysis of 111 reference human epigenomes. *Nature*, **518**(7539), 317 (2015)
16. ENCODE Project Consortium et al. The encode (encyclopedia of dna elements) project. *Science* **306**(5696), 636–640 (2004)
17. L. Nanni, P. Pinoli, A. Canakoglu, S. Ceri, Pygmql: scalable data extraction and analysis for heterogeneous genomic datasets. *BMC Bioinformatics* **20**(1), 560 (2019)
18. L. Nanni, P. Pinoli, A. Canakoglu, S. Ceri, Exploring genomic datasets: From batch to interactive and back, in *Proceedings of the 5th International Workshop on Exploratory Search in Databases and the Web, ExploreDB 2018* (ACM, New York, NY, USA 2018), pp. 3:1–3:6
19. P. Pinoli, S. Ceri, D. Martinenghi, L. Nanni, Metadata management for scientific databases. *Inf. Syst.* **81**, 1–20 (2019)
20. L. Nanni, S. Ceri, C. Logie, Spatial patterns of ctcf sites define the anatomy of tads and their boundaries. *Genome Biol.* **21**(1), 1–25 (2020)
21. J.R. Dixon, D.U. Gorkin, B. Ren, Chromatin domains: the unit of chromosome organization. *Molecular Cell* **62**(5), 668–680 (2016)
22. M.H. Nichols, V.G. Corces, A ctcf code for 3d genome architecture. *Cell* **162**(4), 703–705 (2015)
23. S.S.P. Rao, M.H. Huntley, N.C. Durand, E.K. Stamenova, I.D. Bochkov, J.T. Robinson, A.L. Sanborn, I. Machol, A.D. Omer, E.S. Lander et al., A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159**(7), 1665–1680 (2014)
24. S. Babaei, A. Mahfouz, M. Hulsman, B.P.F. Lelieveldt, J. de Ridder, M. Reinders, Hi-C chromatin interaction networks predict co-expression in the mouse cortex. *PLoS Comput. Biol.* **11**(5), e1004221 (2015)
25. O. Delaneau, M. Zazhytska, C. Borel, G. Giannuzzi, G. Rey, C. Howald, S. Kumar, H. Ongen, K. Popadin, D. Marbach et al. Chromatin three-dimensional interactions mediate genetic effects on gene expression. *Science* **364**(6439), eaat8266 (2019)
26. G. Kustatscher, P. Grabowski, J. Rappsilber, Pervasive coexpression of spatially proximal genes is buffered at the protein level. *Molecular Syst. Biol.* **13**(8), 937 (2017)
27. B. Zhang, S. Horvath, A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, **4**(1) (2005)

28. D. Tian, R. Zhang, Y. Zhang, X. Zhu, J. Ma, MOCHI enables discovery of heterogeneous interactome modules in 3D nucleome (2019)
29. N. Zhou, I. Friedberg, M.S. Kaiser, Hierarchical markov random field model captures spatial dependency in gene expression, demonstrating regulation via the 3D genome. *bioRxiv*, page 2019.12.16.878371 (Dec 2019)
30. M. Varrone, L. Nanni, G. Ciriello, S. Ceri, Exploring chromatin conformation and gene co-expression through graph embedding. *Bioinformatics*, **36**(Supplement_2):i700–i708, 2020
31. Y. Bengio, A. Courville, P. Vincent, A review and new perspectives, *Representation Learning* (2012)
32. X. Yue, Z. Wang, J. Huang, S. Parthasarathy, S. Moosavinasab, Y. Huang, M.S. Lin, W. Zhang, P. Zhang, H. Sun, Graph embedding on biomedical networks: methods, applications, and evaluations. *arXiv preprint arXiv:1906.05017* (2019)
33. A. Grover, J. Leskovec, node2vec: scalable feature learning for networks, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 855–864
34. L. Breiman, Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.