Universidad Nacional Autónoma de México
Programa de Doctorado en Ciencias Biomédicas
Centro de Ciencia Genómicas

Caracterización de las redes reguladoras de *Escherichia coli* K-12 por integración bioinformática de datos de alto rendimiento

Tésis que para optar por el grado de Doctor en Ciencias

Presenta:

Claire Marciane Christine Rioualen

Director de tesis:

Dr. Julio Collado-Vides, Centro de Ciencias Genómicas

Comité tutor:

Dra. Alejandra Medina-Rivera, Instituto de Neurobiología
Dr. José Utrilla-Carreri, Centro de Ciencias Genómicas

Cuernavaca, Morelos, junio del 2022.

Agradecimientos

A Julio Collado, por confiar en mí y darme la oportunidad de realizar una gran aventura laboral y humana, desde una colaboración iniciada en el 2016 hasta culminar con la presente tesis, y por haber fomentado un espacio de trabajo tan amigable y colaborativo como el programa de genómica computacional. A Alejandra Medina y José Utrilla, por apoyarme a lo largo de estos 4 años, y por su retroalimentación tan valiosa en las etapas más cruciales del doctorado. To the Galagan and Wade groups from Boston University and SUNY Albany for the fruitful collaboration.

A Jacques van Helden, por ser no solamente un gran investigador científico, sino también un gran ser humano y una inspiración, y por siempre dedicarme tiempo pese a sus innombrables obligaciones.

A Concepción Hernández, por sus pequeñas atenciones y su gran cariño. A César Bonavides, Romualdo Zayas y Víctor del Moral, por su ayuda en los lazos de la administración, por su amistad, por las vueltas salseras y las palabras náhuatl. A Shirley Alquicira y Heladia Salgado, por su dedicación y simpatía. A Socorro Gama, Carlos Méndez, Alberto Santos, Irma Martínez, y todxs lxs integrantes del programa de genómica computacional, presentes o pasados, por su amistad y su buen humor, y por hacerme sentir parte de una familia.

A la UNAM y el CONACyT, por permitirme realizar mi doctorado durante 4 años en México. A la coordinación del posgrado por su apoyo con los trámites. Al Dr. Christian Sohlenkamp, al Centro de Ciencias Genómicas y sus integrantes, por fomentar un entorno laboral dinámico y cálido. A Lúa Castañeda por su invaluable ayuda, y por haber tanto hecho para permitir que todxs pudiéramos crecer en un ambiente incluyente y empático.

Al centro de lenguas de la UAEM y a Patty 先生, ありがとうございます！Al grupo de senderismo de la facultad de biología, a los salseros del instituto de biotecnología, a

Doña Vicky y Doña amor, y a toda la comunidad que hace del campus Chamilpa un lugar lleno de vida, mucho más allá del doctorado.

A mis amigxs y roomies en Cuernavaca, por su amistad y su convivencia, por los días en el cerro y las noches en la Palapa, por su apoyo y su confianza. A Vargas, Tonalli, Andrei, Gustavo, Diana, Hector, Ale, Karen, Frida, y muchxs más que han cruzado mi camino en la uni y afuera. Al barbas y a las mazorcas por las carnes asadas y las posadas. A mis amigxs y compañerxs del club de tocho y de la academia de salsa, por enseñarme lo valioso que es ser parte de un equipo, y que mientras le eche ganas, nunca dejaré de crecer como persona.

À Luisa et Jean-Hugues pour leur amitié fidèle malgré la distance. To Arato, Douglas, Quentin, Maxime, Guillaume, and all the friends I met at the CRCM. À Wilfried, je garde pour toujours ton amitié en moi. À Lucie et Laurence, rayons de soleil du TAGC, et à Myriam pour sa patience sans limite. A Santiago y Jaime por enseñarme el arte del albur, y a echarle Tajín a la vida. A Alberto, Elena, Alejandro, Claudio y todxs lxs integrantes del *Café des Langues*, por transmitirme el amor al reggaetón. A Tannia y David, por su amistad y por ser parte de mi querido mundo marsello-mexicano.

A mis gatijos, Michi y Casimiro, por enseñarme la paciencia y el amor incondicional... y por la serotonina.

À mes parents, pour leur soutien indéfectible. Pour respecter mon désir d'autonomie et me permettre de suivre le chemin que j'ai choisi. À ma sœur pour les conversations honnêtes et l'humour pourri. Qui l'eut cru, pas moi en tout cas. À mes grands-parents, qui auraient été si fiers de me voir accomplir tout cela.

Resumen

Escherichia coli K-12 es un organismo modelo muy importante para investigar los mecanismos de regulación transcripcional microbiana. Su genoma fue de los primeros secuenciados por completo, y sus genes, operones y factores de transcripción han sido ampliamente estudiados y organizados en bases de datos especializadas. Aunque aún falta información por descubrir sobre las redes de regulación transcripcional de *E. coli*, las tecnologías de secuenciación masiva desarrolladas durante los últimos años permiten contemplar su posible caracterización exhaustiva en un futuro cercano. Para lograr este objetivo, se tienen que revisar los conceptos biológicos fundamentales detrás de los mecanismos de regulación, así como las infraestructuras que permiten el manejo y almacenamiento adecuado de los datos.

En el proyecto de doctorado presentado en este manuscrito, se trabajaron estos aspectos recolectando múltiples fuentes de datos y literatura, estableciendo una nomenclatura para el manejo de los genes, formalizando formatos de almacenamiento para los objetos genómicos y reguladores. Además, se desarrollaron herramientas computacionales para realizar el análisis automatizado de datos generados por tecnologías de secuenciación de alto rendimiento. Finalmente, este trabajo culminó con la integración de dichos datos con los datos de referencia generados mediante experimentos clásicos, ofreciendo un nuevo fundamento para entender los mecanismos de regulación genética a escala global en un organismo modelo como *Escherichia coli* K-12.

Comprehensive characterization of *Escherichia coli* K-12 regulatory networks by bioinformatics integration of high-throughput data

---

Table of contents

Abstract

Escherichia coli K-12 is the best studied free-living organism on Earth, which makes it a fundamental model organism in microbiology. It is a reference for the study of transcriptional regulation. Extensive information about its genes, transcription factors, and transcription units has been manually curated and indexed for decades in dedicated databases, and its genome was one of the first to be entirely sequenced and published.

Currently, the wide variety of high-throughput technologies available allows for the acquisition of larger collections of genomic features, regulatory elements or gene expression profiles, and does so with a higher-than-ever accuracy, opening the possibility of comprehensively characterizing the transcriptional regulatory network of a species such as *E. coli*.

However, such a tremendous amount of data triggers new concerns regarding the proper analysis and integration of this new information within the existing frameworks, together with the knowledge established through decades of low-throughput experimentation and manual literature curation.

In this work, I tackled those challenges by working through these issues. I searched public databases and recent literature for relevant datasets; I revised key biological concepts with the aim of fitting a common framework; and I conceived bioinformatics tools for the automatic and reproducible analysis of high-throughput datasets. Finally, I built on those foundations to perform the analysis of dozens of high-throughput datasets, the standardization of thousands of genomic features and regulatory elements, and their integration with reference knowledge from classic experiments. This provides a foundation for further research to understand gene regulation at a global scale in this model organism.

Abbreviations

ChIP-exo	chromatin immunoprecipitation with exonuclease digestion
ChIP-seq	chromatin immunoprecipitation followed by high-throughput sequencing
CTG	co-transcribed genes
DAP-seq	DNA affinity purification sequencing
DNA	deoxyribonucleic acid
ENA	European nucleotide archive
GEO	gene expression omnibus
gSELEX	genomic systematic evolution of ligands by exponential enrichment
HT	high-throughput
LT	low-throughput
mRNA	messenger RNA
ORF	open reading frame
RBS	ribosome binding site
RNA	ribonucleic acid
PSSM	position-specific scoring matrix
RNAP	RNA polymerase
RNA-seq	RNA sequencing
SRA	sequence read archive
sRNA	small RNA
TF	transcription factor
TFBS	transcription factor binding site
TFRS	transcription factor regulatory site
TRN	transcriptional regulatory network
TSS	transcription start site
TTS	transcription termination site
TU	transcription unit

Introduction

Escherichia coli K-12, a fundamental microbial model organism

Escherichia coli is a Gram-negative, facultative anaerobic *gammaproteobacteria* from the *Enterobacteriaceae* family. Though it is mainly known for living in the digestive system of healthy mammals as a commensal species, it also has the capacity of being a free-living organism or being pathogenic.

Escherichia coli was first discovered in 1885 by Theodor Escherich, who would later on give this new species its current name. Over time, it became a model organism for studying and understanding key biological processes, due to its ease of culturing in a laboratory setting, its rapid reproduction and its relative inexpensiveness.

In particular, a strain labeled as "K-12" was isolated in 1922, and was the basis for scientific breakthroughs such as the first description of the mechanism of bacterial conjugation (Lederberg and Tatum, 1946), and the discovery of the transcriptional regulation of the Lac operon (Jacob and Monod, 1961). Finally, the genome of *Escherichia coli* K-12 was one of the first genomes to be completely sequenced (Blattner et al., 1997).

Its anatomy includes a single, circular chromosome encapsulated in the cell envelope along with ribosomes and other proteins and cellular components. The envelope is made of an inner cytoplasmic membrane, a peptidoglycan-rich periplasmic space and an outer membrane. The cell also possesses peritrichous *flagella* and *pili* that enable motility and intercellular communication (Figure 1).

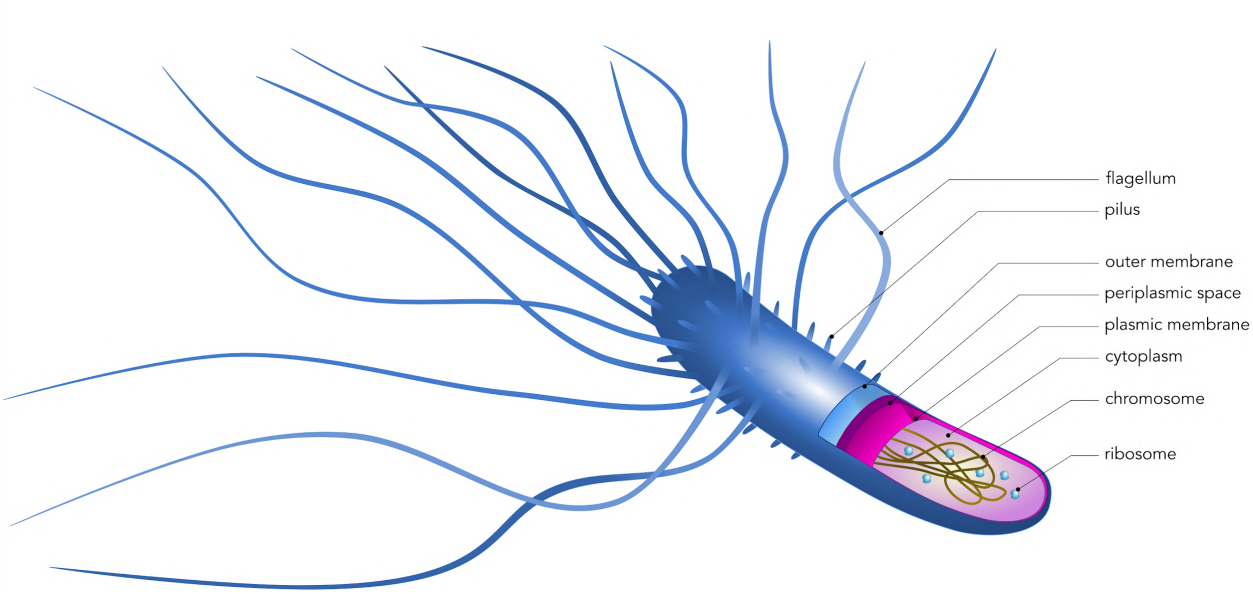


Figure 1. *Escherichia coli* cell structure (simplified).

Bacterial genome and gene expression

Prokaryotes typically possess a single, circular and double-stranded molecule of DNA, and in some cases, one or several smaller plasmids. The main chromosome contains the majority of the genes, finely organized spatially into operons, and expressed either constitutively or under specific growth conditions.

Operons are defined as clusters of genes that share the same orientation and are usually separated by short intergenic segments (Salgado et al., 2000; Moreno-Hagelsieb and Collado-Vides, 2002), and are under the control of a single promoter and co-transcribed together into polycistronic RNA molecules. The transcription mechanism is triggered by the binding of a protein complex called RNA polymerase (RNAP) on a promoter sequence specifically recognized via its sigma subunit (Figure 2a). The RNAP can then open the double-stranded DNA around the transcription start site (TSS), initiate transcription, and slide along the DNA sequence, resulting in the elongation of the transcript until reaching a terminator sequence (Figure 2b). The resulting messenger RNAs can contain one or several ribosome binding sites (RBS) allowing their translation into proteins, while small RNAs can complete other metabolic and/or regulatory functions (Figure 2c).

Escherichia coli K-12 has a circular chromosome of 4,641,652 million base pairs of length with a high density of genes, accounting for about 90% of the total DNA sequence. It contains a total of 4,736 inventoried genes, of which 4,326 are currently reported as protein-coding and another 219 as coding for small RNAs. Those genes are organized into 2,592 operons (Tierrafría, Rioualen et al., 2022; Keseler et al., 2021).

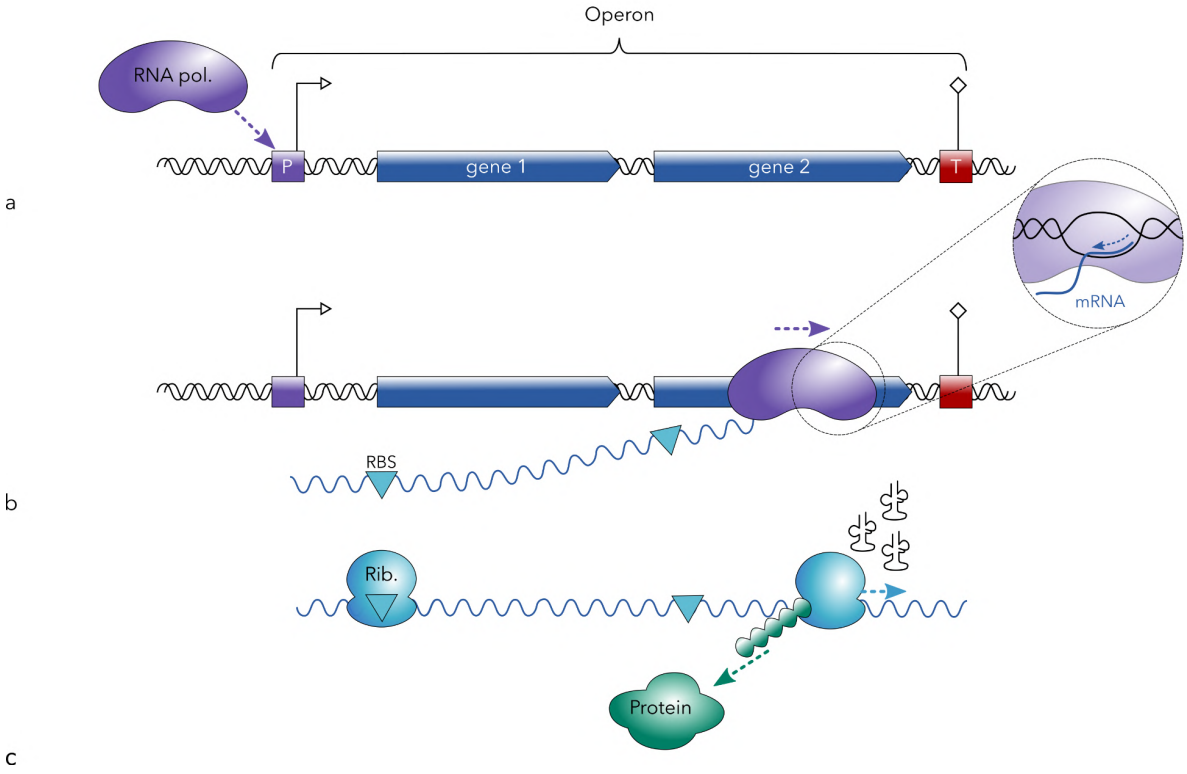


Figure 2. a. Operon structure. An operon is composed of one or several genes that are associated with a promoter. The RNA polymerase complex binds the promoter in order to initiate transcription of the downstream genes. **b. Transcription.** After binding the promoter, the polymerase opens the double-stranded DNA to initiate the transcription, and slides along the DNA sequence to elongate the transcript, until reaching a terminator. **c. Translation.** Ribosomes can bind mRNAs via ribosome binding sites, and translate their downstream sequences into amino acids and proteins.

Gene regulatory networks

Regulation of gene expression is crucial for living organisms in order to be able to adapt to environmental conditions and maintain homeostasis, even more so for a micro-organism such as *E. coli*, which holds the capacity of surviving and even striving

in a wide variety of environments and lifestyles. The mechanisms of adaptation involve several coordinated layers: the signaling network comprises intra- and extra-cellular receptors that detect environmental changes (temperature, osmolarity, etc) and signal transduction mechanisms; the transcriptional regulatory network consists of protein-DNA interactions that can activate or repress the expression of specific genes (genetic switches) and trigger appropriate metabolic responses; and finally the metabolic network is made of interconnected pathways of biochemical reactions that are triggered by specific signals (Ledezma-Tejeida et al., 2017).

The modulation of gene expression can occur at any stage of the process: different growth conditions will affect signal transduction, structural modifications of the DNA can impact the level of transcription of specific regions (eg. DNA supercoiling), transcription initiation can be triggered differentially through alternative sigma subunits of the RNA polymerase holoenzyme, small RNAs can act at the post-transcriptional level to silence mRNA molecules and prevent their translation into proteins, and some mRNAs are able to self-regulate (riboswitches).

But one of the most important mechanisms involved in the regulation of gene expression at the transcriptional level involves DNA-binding proteins called transcription factors (TFs). TFs have the ability to bind to specific sites of the DNA that are typically located upstream of genes and operons, thereby allowing or prohibiting access of the RNA polymerase to promoter regions, in order to positively or negatively regulate the expression of the downstream genes. These mechanisms were first described by Jacob and Monod (Jacob and Monod, 1961) with their work on the lactose operon (Figure 3), and have been shown to be responsible for the direct regulation of more than half of *Escherichia coli* K-12's genes (Pérez-Rueda et al., 2015). The complete set of transcription factors and their respective target genes form the so-called transcriptional regulatory network (TRN).

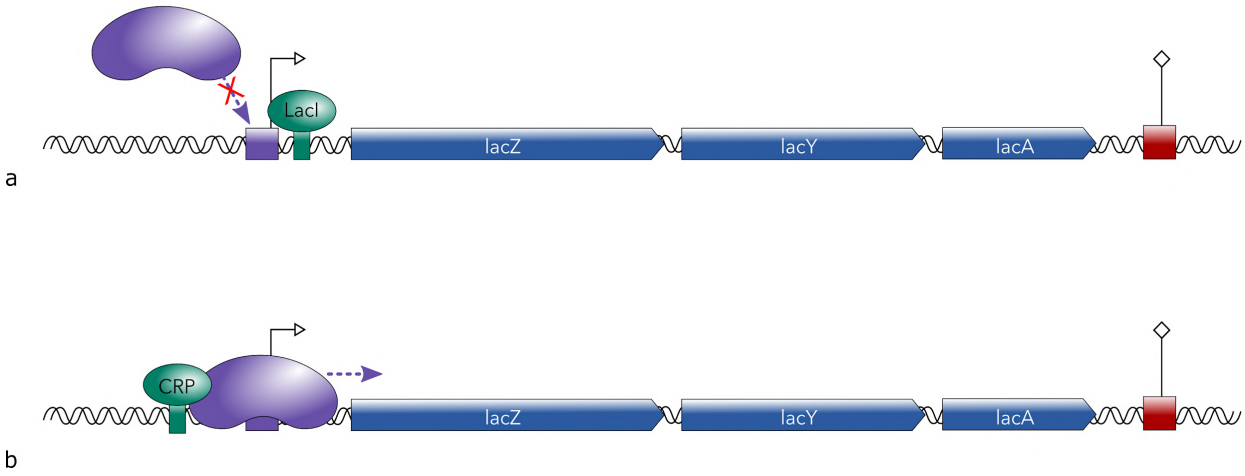


Figure 3. Transcriptional regulation of the lactose operon (Jacob and Monod, 1961). **a. Repression of expression.** In the absence of lactose, the LacI transcription factor is usually bound to a specific site located immediately downstream of the operon promoter, impeding the recruitment of the DNA polymerase and the initiation of the transcription of this operon, coding for lactose metabolism-related genes. **b. Induction of expression.** When lactose is present, allolactose is formed and binds to the repressor, which consequently unbinds from the DNA. Combined with a low level of glucose, this allows CRP to bind its cAMP co-factor and its operator upstream of the LacI operon promoter, contributing to the induction. This phenomenon allows for the recruitment of the RNA polymerase, and thus the transcription of lactose metabolism genes, in order to use lactose as a nutrient. Irrespective of lactose, in the presence of glucose, cAMP levels go down, provoking the unbinding of CRP transcription factor from its activator sites in many operons for carbon utilization such as the lactose operon.

Transcription factors

Transcription factors are defined as DNA-binding proteins that allow or block the transcription genes, and are not part of the RNAP core or holo enzyme (Zhou and Yang, 2006). Meta-analyses have shown that TF-coding genes can make up for up to 10% of all coding genes in bacteria, though this proportion can vary greatly depending on bacterial genome size and lifestyle (Pérez-Rueda et al., 2004). Though a complete and definitive identification of all TFs in *E. coli* K-12 is still lacking, a consensus has been reached over the years around a total estimate of 300 to 350 TFs (Pérez-Rueda and Collado-Vides, 2000; Pérez-Rueda et al., 2015; Gao et al., 2018; Flores-Bautista et al., 2020; Kim et al., 2021), most of which have been shown to perform negative auto-regulation (Pérez-Rueda et al., 2015).

Transcription factors usually comprise a DNA-binding domain (DBD) and a companion domain (CD). The DNA-binding domain is necessary for a TF to bind onto specific sites

of the genome, thus called transcription factor binding sites (TFBS), while the companion domain can have a variety of functions such as ligand binding, protein-protein interactions, or enzymatic activities (Pérez-Rueda et al., 2018). Each TF binds specifically to its own target sites and regulates specific target genes, some of which may be TF-coding themselves, generating a complex network of interactions. Together, a group of genes that are regulated by a common transcription factor form a regulon. A variety of DBDs has been described, however in bacteria about 80% of them contain a "helix-turn-helix" or HTH segment that binds to the DNA (Pérez-Rueda and Collado-Vides, 2000; Flores-Bautista et al., 2020). Protein binding domains have been used to classify bacterial TFs into evolutionary families (Pérez-Rueda et al., 2004), and DNA binding sites have been used to identify TF-specific binding genomic patterns.

Currently, 222 TFs have been characterized and confirmed with experimental evidence (Tierrafría, Rioualen et al., 2022), mostly through binding of purified proteins and site mutation, sometimes combined with additional data of lower confidence such as gene expression analysis and binding of cellular extracts. Additionally, computational methodologies have been developed in order to predict TFs that have not yet been characterized experimentally. Predictions were based on several criteria and methods, such as sequence homology with experimentally characterized TFs, identification of a DBD - preferentially including an HTH structure (Pérez-Rueda and Collado-Vides, 2000; Pérez-Rueda et al., 2015), identification of orthologous proteins (Flores-Bautista et al., 2020), as well as deep-learning methods (Gao et al., 2018; Kim et al., 2021).

Regulatory interactions

Most of the regulatory interactions known to date were identified from *in vitro* experiments through the binding of purified proteins. DNAse footprint uses DNAse-protected fragment isolation to detect the "footprint" of a protein on the DNA sequence with a good accuracy (Galas and Schmitz, 1978). On the other hand, electrophoretic mobility shift assays (EMSA, also called gel shift) consist in the electrophoretic separation of DNA fragments of interest with or without bound proteins (Garner and Revzin, 1981), allowing the identification of transcription factor binding sites. More recently, biotin-DNA affinity purification sequencing (DAP-seq) (O'Malley et al., 2016) and genomic systematic evolution of ligands by exponential enrichment (gSELEX) (Shimada et al., 2018) have also been used.

The identification of TF binding sites took a new turn with recent *in vivo* chromatin immunoprecipitation (IP) techniques combined with high-throughput sequencing technologies. They share the same principle: after a TF of interest is bound to whole-genome DNA via its specific sites, cross-linking of the protein is performed. The whole DNA is then fragmented using a process such as sonication, and an antibody that is specific to the TF is added. DNA fragments that are bound by the TF are isolated through IP, and finally the cross-linking is reversed, leaving free DNA fragments originally bound by the TF. These fragments are then amplified and sequenced, before they are finally mapped to the genome of reference. In the case of ChIP-on-chip (Buck and Lieb, 2004), the sequencing step is performed by using DNA microarrays. As HT sequencing technologies improved, binding sites identification gained resolution while dramatically lowering in cost. The ChIP-seq technology (Johnson et al., 2007) shares the same strategy, but the final sequencing is performed using next-generation sequencing devices, resulting in a better resolution of the binding locations. Finally, ChIP-exo (Rhee and Pugh, 2011) is similar to ChIP-seq, but includes an additional step that consists in trimming DNA from the protein-DNA complexes before the IP is carried out, increasing the precision of protein binding sites identification. In all cases, the resulting reads can be aligned to a genome sequence of reference, and the regions enriched in reads at certain positions of the genome form so-called "peaks", that indicate possible binding positions for the TF of interest (Figure 4).

When a TFBS can be linked to evidence of a change in gene expression of immediate downstream genes under a given growth condition, it can formally be identified as a regulatory sequence, and is then labeled as transcription factor regulatory site (TFRS) (Mejía-Almonte et al., 2020), while the regulated genes are considered as targets of the transcription factor.

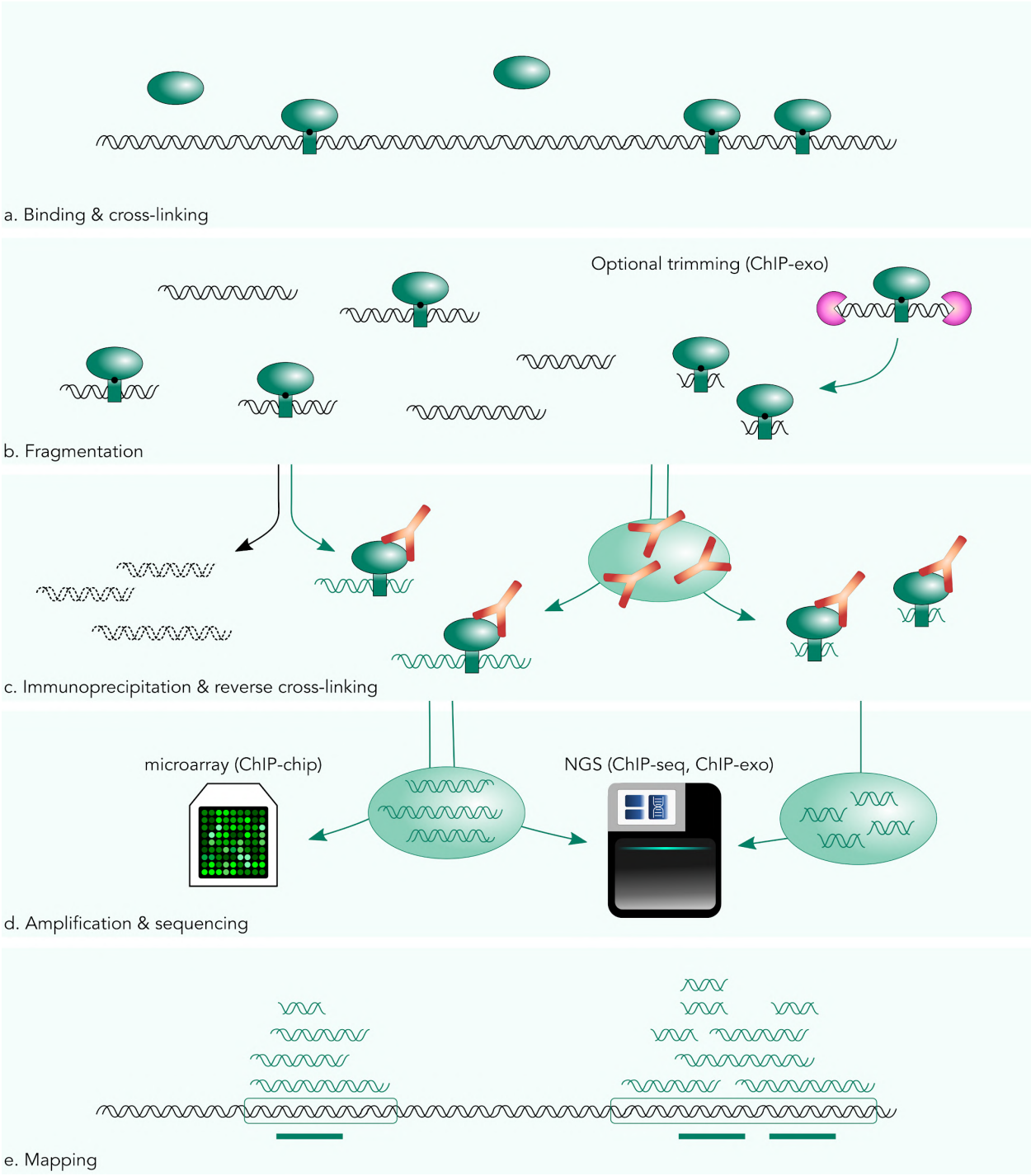


Figure 4. Overview of chromatin immunoprecipitation-based techniques for protein binding sites identification. **a.** The protein of interest is cross-linked to the whole DNA molecule. **b**. The genome is fragmented. **c**. Immunoprecipitation is performed using an antibody specific to the protein of interest. **d**. After reversing the cross-linking, DNA fragments can be amplified and sequenced. **e**. Upon mapping the resulting reads to the whole genomic sequence of the reference organism, regions with a high density of mapped reads, or peaks, indicate protein binding sites.

The relative levels of expression of genes can be measured using transcriptomic technologies. They essentially consist in extracting total mRNAs from a cell and perform their fragmentation, purification, reverse-transcription, and sequencing. The latter step used to be realized using microarrays, and is now routinely done using massive parallel sequencing technologies: this is the RNA-seq technology. Transcriptome analyses can also uncover genomic elements such as transcription units, and transcription start and termination sites. Various protocols based on RNA-seq strategies have been proposed, that allow for the identification of TSSs at single-nucleotide resolution (Conway et al., 2014), and more recently, for the determination of entire transcripts, along with their TSSs and TTSs (Yan et al., 2018; Ju et al., 2019).

Databases on transcriptional regulation in *Escherichia coli* K-12

Most of the current knowledge of *E. coli*'s genome, its features and its regulatory processes, comes from the accumulation of low-throughput experiments realized and published over decades of scientific investigations. Extensive information about *E. coli* K-12 TFs, their binding sites, target genes and operons has been manually curated and indexed for decades by the team at the Program of Computational Genomics at the CCG, and simultaneously described in dedicated databases such as RegulonDB (Tierrafría, Rioualen et al., 2022) and EcoCyc (Keseler et al. 2021).

Since the creation of RegulonDB in 1998, biocurators have gathered information from thousands of original scientific publications, reporting data from classical molecular genetics wet-laboratory experiments. However, genome-scale technologies based on high-throughput sequencing now allow for the accurate identification of genomic features and regulatory elements genome-wide. Additional interactions based on gene expression analyses and computational predictions were also integrated. In order to account for their different level of reliability, a system of classification was implemented in the database, that categorizes the confidence associated with regulatory features as "strong" or "weak", depending on the pieces of evidence they rely on (Weiss et al., 2013). Features that are associated with solid physical and genetic evidence are classified as strong, while those associated with less reliable evidence (i.e. change in expression of a target gene, that could be indirect) are classified as weak.

To date, the total transcriptional regulatory network currently characterized of *E. coli* comprises 222 TFs regulating 1,856 genes, for a total of 4,665 regulatory interactions (2,836 strong- and 1,829 weak-confidence RIs) (Tierrafría, Rioualen et al, 2022). However it is well known that this is just a fraction of the complete transcriptional regulatory network, since nearly a third of the estimated total of TFs lack characterization, and a similar proportion of *E. coli*'s 4,700 genes are not yet functionally characterized (Flores-Bautista et al., 2018; Gao et al., 2018).

Databases of high-throughput datasets

High-throughput datasets are usually made publicly available upon their publication, and uploaded to dedicated databases. The main ones are the European Nucleotide Archive (ENA) and ArrayExpress from the EMBL-EBI, that store nucleotide sequencing information and high-throughput functional genomics experiments respectively (https://www.ebi.ac.uk/); and the Sequence Read Archive (SRA) and Gene Expression Omnibus (GEO), their counterparts from the NCBI (https://www.ncbi.nlm.nih.gov/). Additionally, several databases store more specialized datasets, that are worth mentioning: COLOMBOS offers transcriptomic data from prokaryotic organisms (Moretto et al., 2016) (https://colombos.net), and Transcription Profile of *Escherichia coli* (TEC) offers genomic SELEX data for *E. coli* TFs (Ishihama et al., 2016) (www.shigen.nig.ac.jp/ecoli/tec/).

Objectives

Problematic

Despite decades of investigation and experimentation dedicated to *Escherichia coli* K-12, its transcriptional regulatory network remains far from being exhaustively characterized: nearly 30% of its TFs and genes are not yet characterized, and most of the characterized TFs lack whole-genome profiles that would allow to retrieve exhaustive binding sites and target genes.

Main goal

The main objective of this work is to take advantage of recent, high-throughput-based (HT) published data available for *E. coli* K-12 and combine it with the low-throughput-based (LT) knowledge of reference curated in RegulonDB, in order to complete its known transcriptional regulatory network.

Specific goals

In order to pursue this final aim, a large amount of data from very diverse sources had to be manipulated, which highlighted a recurrent issue: the identification and the mapping of genomic objects and coordinates between sources. In order to circumvent this bottleneck, I developed an R library that performs a number of conversions and operations on genes and other genomic features (Chapter 1: Getting a hang of *E. coli* genes and transcription factors).

The goal of characterizing the regulatory networks of *Escherichia coli* also triggered considerations about genomic features such as promoters, TUs and terminators, and the need to lay out definitions and integrate novel knowledge alongside established

concepts, in order to integrate recent HT-based data with previous knowledge (Chapter 2: Building a comprehensive set of genomic features).

On another hand, I developed a library of bioinformatic workflows that allows for the analysis of high-throughput data in a reproducible manner, with a focus on ChIP-seq and RNA-seq data (Chapter 3: Building tools to analyze high-throughput datasets). This work was published in the form of a protocol (Rioualen et al., 2019).

Building on these founding elements, I worked towards the central goal of my PhD: the completion of the *E. coli* K-12 transcriptional regulatory network by integrating high-throughput data (Chapter 4: Integration of high-throughput data within a reference framework). This work constituted a major upgrade of RegulonDB and was recently published (Tierrafría, Rioualen et al., 2022) as version 11.0.

Finally, I investigated an alternative approach to building transcription factor binding matrices based on *de novo* pattern discovery using the curated binding sites available in RegulonDB, which subsequently enabled me to produce an alternative collection of TF binding motifs for *Escherichia coli* (Chapter 5: An alternative collection of binding motifs).

Chapter 1.

Getting a hang of *E. coli* genes and transcription factors

Problematic

Despite a wide knowledge of the *E. coli* K-12 genome and regulatory networks, the computational manipulation of numerous datasets from a variety of sources can prove to be rather fastidious, due to a lack of congruence in the definition of biological objects, as well as their names or identifiers. Genes and their products can be referred to using a variety of names and synonyms, obsolete or not, different bnumbers (an index specific of *E. coli* genes), coordinates can change over time due to the addition of new knowledge, and frequent updates in genome annotations can lead to discrepancies between sources. Additionally, a significant amount of published datasets are based on obsolete genome assemblies, leading to erroneous genomic coordinates.

In order to overcome these limitations and process datasets containing information on *E. coli* genes, TUs, promoters, or any other genomic features associated with coordinates, I took on the challenge of building a dictionary of genes, TFs and genomic coordinates. After extracting information from several public databases and articles, I built reference tables for genes and transcription factors that allow for an easy translation of inconsistent names or coordinates and created "EcoliGenes", a library of functions that perform verifications and homogenization of *E. coli* genomic datasets (https://github.com/rioualen/EcoliGenes).

Comprehensive table of genes and their attributes

There are numerous names, identifiers and synonyms for most genes and proteins of *E. coli*, as well as outdated annotations, products or coordinates. This complicates the

programmatic manipulation of datasets containing genomic information. In order to be able to process datasets containing any information on *E. coli* genes, TUs, promoters, or any other genomic objects associated with coordinates, I started gathering comprehensive information into one single place. I first retrieved all genes and their products from RegulonDB (Figure 5a), and completed this information with additional data extracted from Ecocyc and Genbank. I merged them first on the basis of their bnumbers, and then using their symbols and coordinates. Finally, I added "reference" columns to this *master* table: reference bnumber, reference symbol, reference start, reference stop, reference strand; and "synonym" columns to store additional names from any source: gene synonyms and product synonyms (Figure 5b).

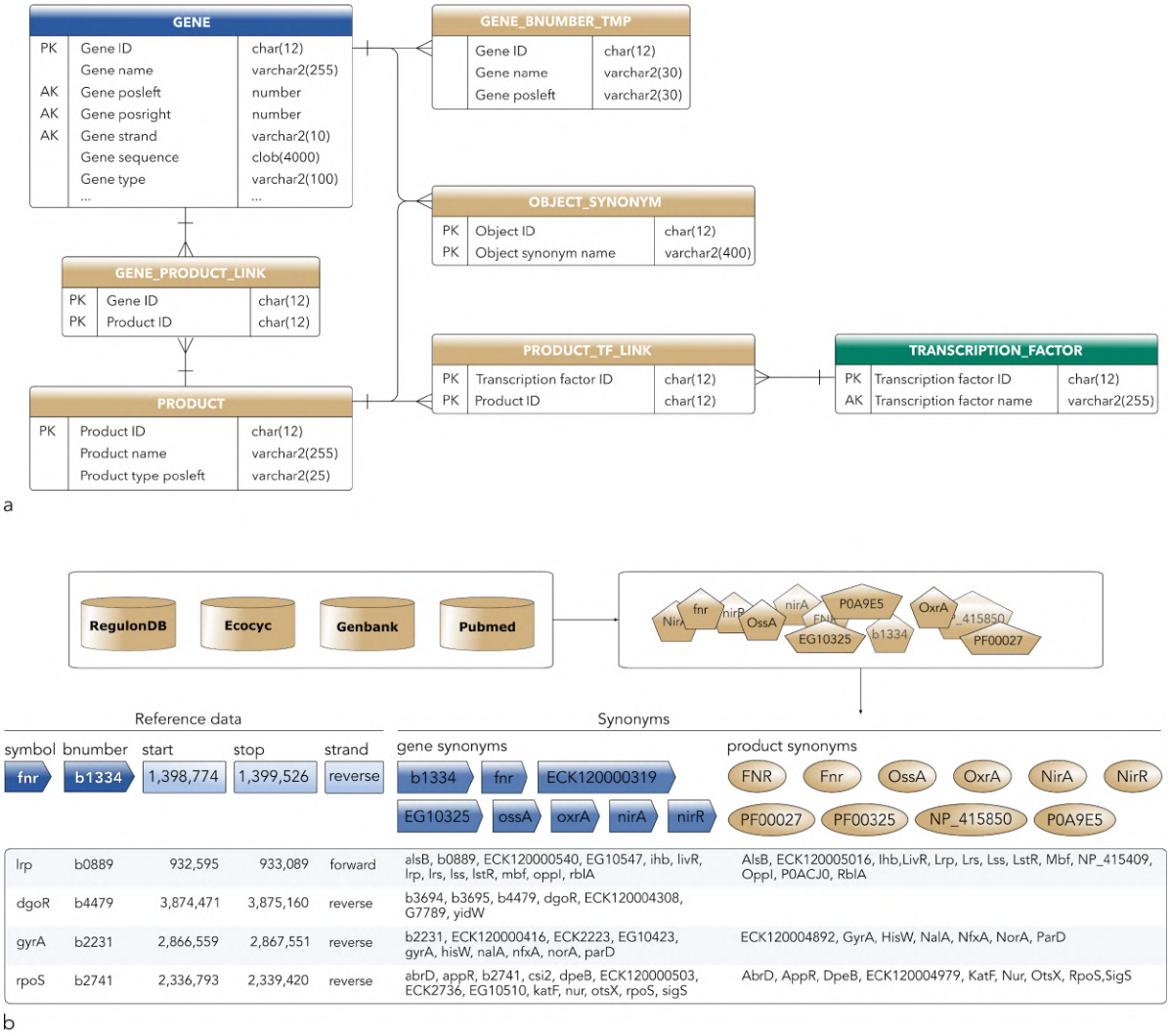


Figure 5. a. Entity-relationship diagram of the gene information retrieved from RegulonDB. **b.** Construction of the gene master table: extraction, classification and organization of gene names, synonyms, attributes and products.

Comprehensive list of transcription factors

To date, there is no single consensual list of confirmed TFs for *E. coli*. RegulonDB 11.0 contains 222 experimentally confirmed TFs (Tierrafría, Rioualen et al., 2022), associated with at least one regulatory interaction, however the total number of TFs in *E. coli* is estimated to be slightly above 300, and several groups have predicted TF candidates based on *in silico* predictions using criteria such as the presence of a DNA binding domain or a significant homology with known TFs (Pérez-Rueda and Collado-Vides, 2000; Pérez-Rueda et al., 2015; Gao et al., 2018; Flores-Bautista et al., 2020; Kim et al., 2021).

By combining those different sources, I built a list of 408 confirmed or proposed transcription factors, and gathered their most relevant attributes. First I retrieved the information available in RegulonDB (Figure 6a), then added gene products annotated in Genbank as transcriptional regulators (putative or not), as well TF predictions published in recent years (Pérez-Rueda et al., 2015; Flores-Bautista et al., 2020; Kim et al., 2021). I added their respective identifiers from external databases such as Uniprot, RefSeq, and Pfam, other existing synonyms, and the following "reference" columns: reference TF name, reference gene symbol, reference gene bnumber. It is worth noting that there are heterodimeric TFs, which are therefore associated with two genes. Proteins that are considered as TFs both individually and as part of a dimer have duplicate entries, while proteins that are known to be regulatory only as part of a dimer will be considered as synonym names for said dimer. Finally, I performed a comparison of the lists of TFs from all the different sources, which shows the existence of several discrepancies between one another (Figure 6b), mainly due to the absence of experimental evidence to back up computational predictions. The most striking difference is observed in the list of TFs predicted by Kim and colleagues, which includes 58 proposed TFs absent from all other datasets (Kim et al., 2021). This can be explained by the fact that they used a deep learning approach that does not rely on homology with known TFs, thus revealing potential new classes of TFs, though they could also be false positives.

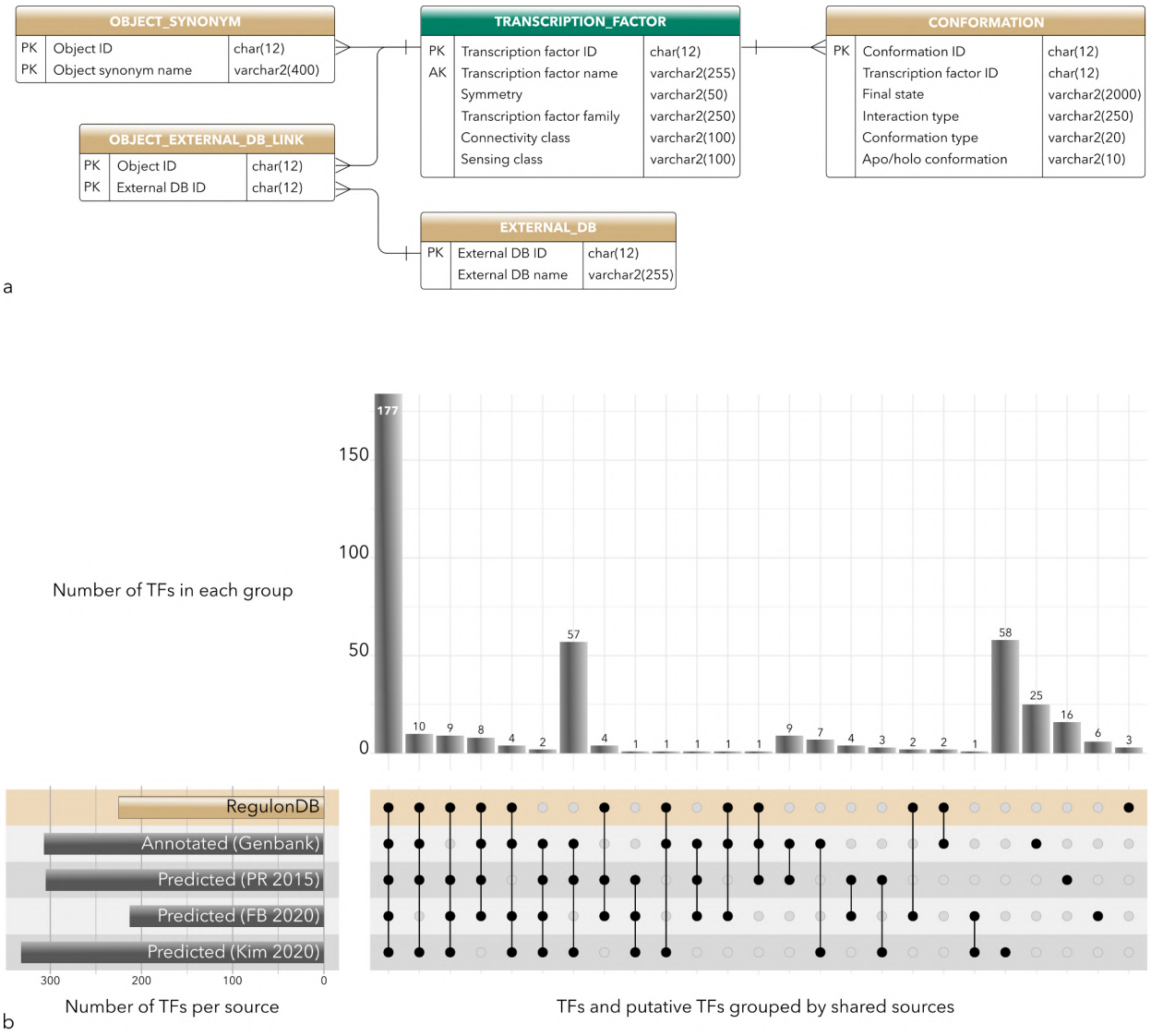


Figure 6. a. Entity-relationship diagram of the TF information retrieved from RegulonDB. **b.** Comparison of the lists of TFs from different sources.

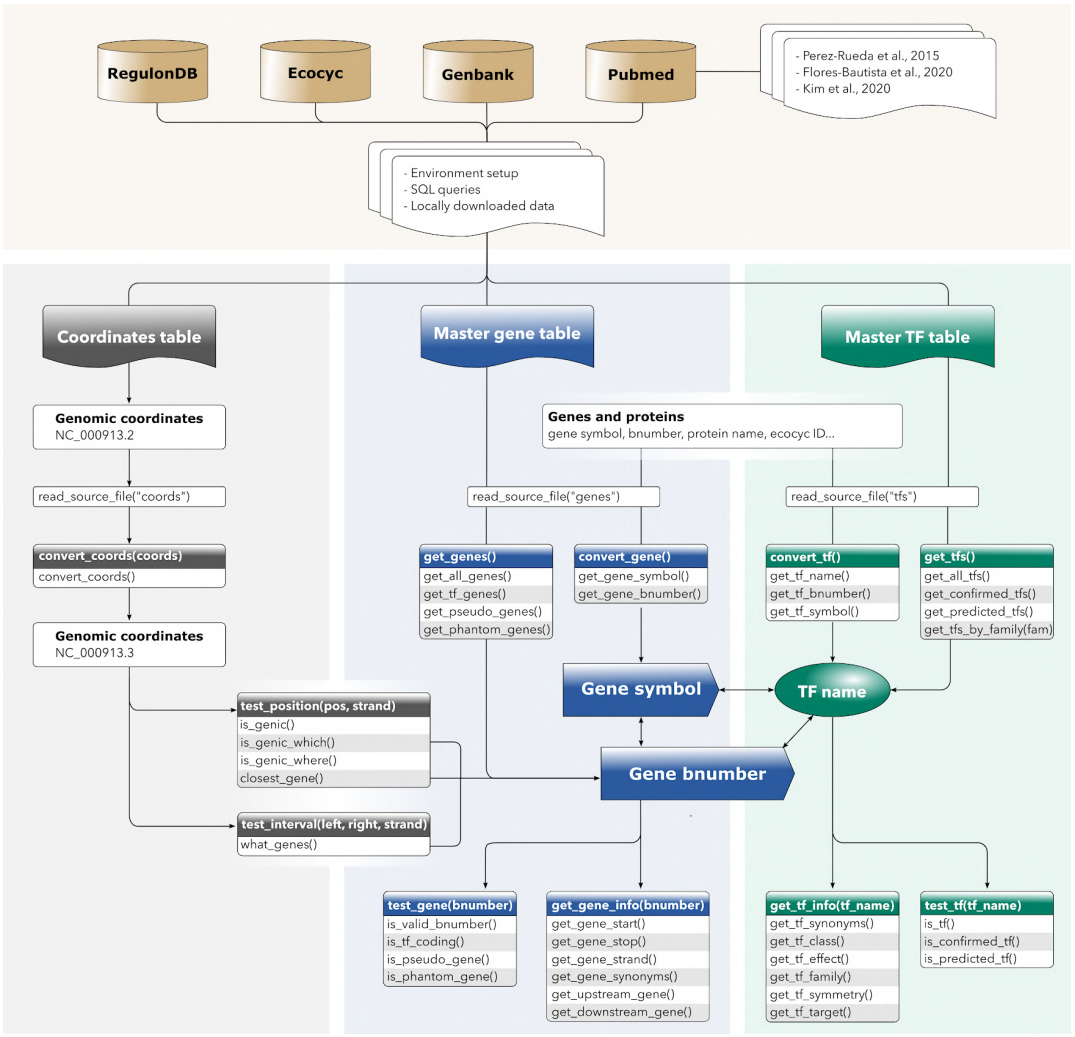
The EcoliGenes library

I built a *tidy* R package designed to process *E. coli* datasets using the master tables described above. It is conceived to manipulate any number of genes, TFs or genomic coordinates using vectorized functions that allow manipulating data frames using basic functions from the tidyverse packages (https://www.tidyverse.org/) (Figure 7).

These tools can be divided into three categories:

- **Coordinate-based tools** (gray-headed boxes). The **convert_coords** function allows to convert genomics coordinates based on *E. coli* genome version NC_000913.2 to the currently used genome version NC_000913.3. The functions **test_position** and **test_interval** perform the extraction of information related to specific genomic positions or regions.
- **Gene-based tools** (blue-headed boxes). Genes reported as symbols or identifiers from any source can be readily converted to symbols or bnumbers of reference with the **convert_gene** function. The **get_genes** functions allow retrieving a list of genes' bnumbers given a specific criteria. Then, bnumbers can be used to perform boolean tests using **test_gene**, or retrieve specific attributes of the genes with **get_gene_info**.
- **TF-based tools** (green-headed boxes). Transcription factors reported as protein names, gene symbols or any other identifiers are converted to their names of reference using **convert_tf**. The **get_tfs** functions allow retrieving a list of TF reference names given specific criteria. Then, they can be used to perform boolean tests using **test_tf**, or retrieve specific attributes of the TF with **get_tf_info**.

With such a simple tool, any number of datasets from a variety of sources and times can be readily uniformized and reliably compared and visualized. This work was heavily used for the processing of numerous datasets presented in Chapter 2 and Chapter 4.



a

```r
gene_list <- sample(get_all_genes_synonyms(), 100)          # get a random list of gene names

gene_table <- data.frame(gene_list) %>%
  mutate(symbol = convert_gene(gene_list, to = "symbol"),   # get symbol of reference
         bnumber = convert_gene(gene_list, to = "bnumber"), # get bnumber of reference
         start = get_gene_start(bnumber),                   #
         stop = get_gene_stop(bnumber),                     # get gene coordinates & strand
         strand = get_gene_strand(bnumber)) %>%             #
  filter(is_tf_gene(bnumber)) %>%                           # select only TF-coding genes
  mutate(tf_name = convert_tf(bnumber),                     # get TF common name
         family = get_tf_family(tf_name))                   # get TF attributes
```

gene_list
ydhM
G7326
yhiT
yeaM
ECK120000526
rtcR
ECK4055
b2087

→

gene_list	symbol	bnumber	start	stop	strand	tf_name	family
ydhM	nemR	b1649	1,726,023	1,726,622	+	NemR	TetR/AcrR
G7326	iscR	b2531	2,661,643	2,662,131	-	IscR	Rrf2
yhiT	gadE	b3512	3,658,366	3,658,893	+	GadE	LuxR/UhpA
yeaM	nimR	b1790	1,874,755	1,875,576	-	NimR	AraC/XylS
ECK120000526	lexA	b4043	4,257,115	4,257,723	+	LexA	LexA
rtcR	rtcR	b3422	3,558,268	3,559,866	+	RtcR	EBP
ECK4055	soxR	b4063	4,277,469	4,277,933	+	SoxR	MerR
b2087	gatR	b4498	2,169,693	2,171,727	-	GatR	NA

b

Figure 7. Framework of the EcoliGenes library. a. Reference data is gathered from several databases and publications in order to build gene and TF master tables (beige frame). Operations can be performed on coordinates, genes and TFs (gray, blue and green frames respectively). **b.** Use case: a random list of gene names is generated, that are translated to their reference names and filtered to get tf-coding genes. Finally, genes and TFs attributes are retrieved.

Reference & availability

Github

The EcoliGenes library is available at Github:
https://github.com/rioualen/EcoliGenes

Poster

This work was presented under the form of a poster at the Bioconductor conference 2022 in Seattle, Washington:

The EcoliGenes library: Solving the never-ending struggle with Escherichia coli K-12 genes

Citation

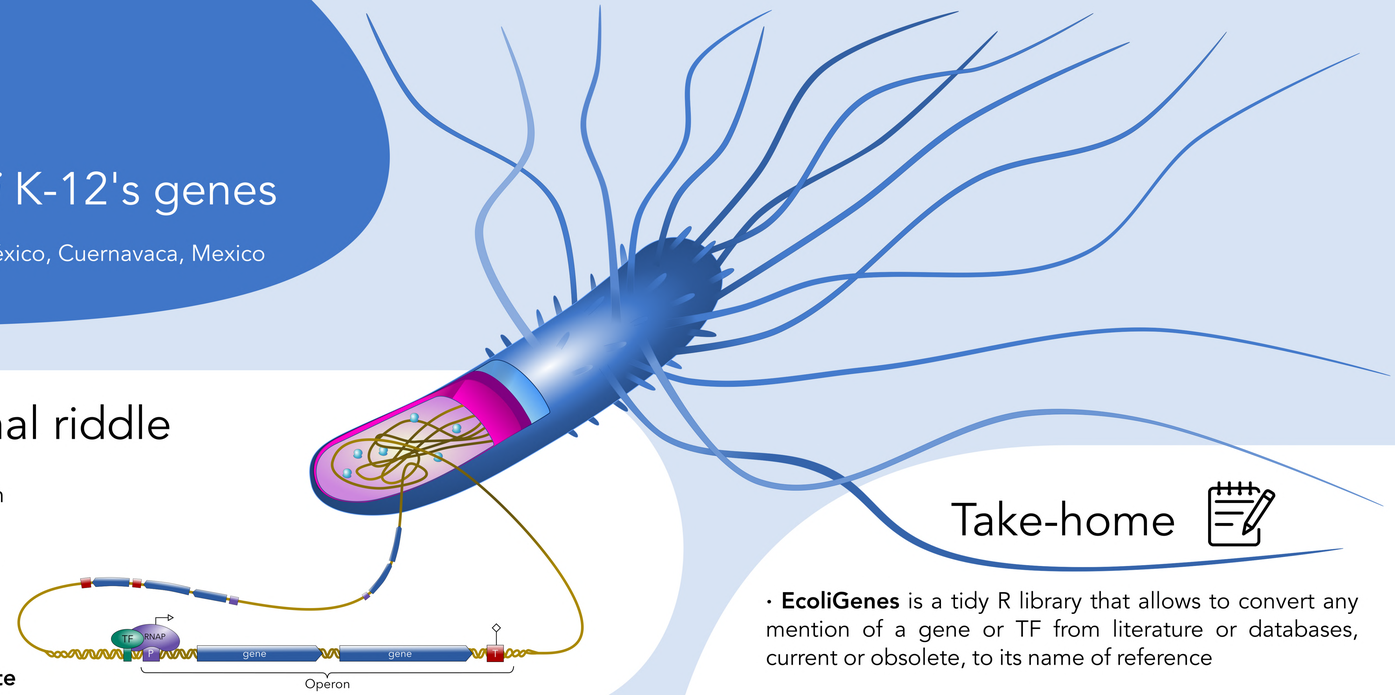
Rioualen C. The EcoliGenes library: Solving the never-ending struggle with *Escherichia coli* K-12 genes [version 1; not peer reviewed]. F1000Research 2022, 11:810 (poster) (https://doi.org/10.7490/f1000research.1119040.1)

The EcoliGenes library

Solving the never-ending struggle with *Escherichia coli* K-12's genes

Claire Rioualen

Centro de Ciencias Genómicas, Universidad Nacional Autónoma de México, Cuernavaca, Mexico



Escherichia coli K-12: a well known model, yet an eternal riddle

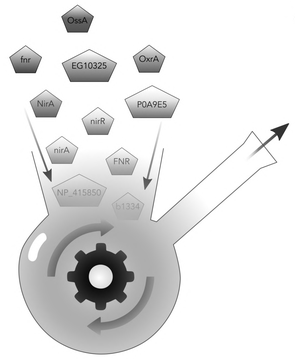
- *E. coli* is a classic **model organism** that has been used for decades in genetics and transcriptomics research
- Genome of 4.6 Mb fully sequenced and published in 1997
- Features more than **4,000 genes** organized in operons, defined by transcription start and termination sites
- An estimated **300 trancription factors** (TFs) control its densely inter-connected regulatory network

- Hundreds of genes are still not characterized, and most TFs have unknown function and targets
- Annotations keep adding up over time, leading to a **large amount of information** that is **tough to navigate**

Take-home

- **EcoliGenes** is a tidy R library that allows to convert any mention of a gene or TF from literature or databases, current or obsolete, to its name of reference

- **Genes, coordinates** and other **genomic features** can be manipulated, converted, and summarized just like the next dataframe, using packages from the tidyverse



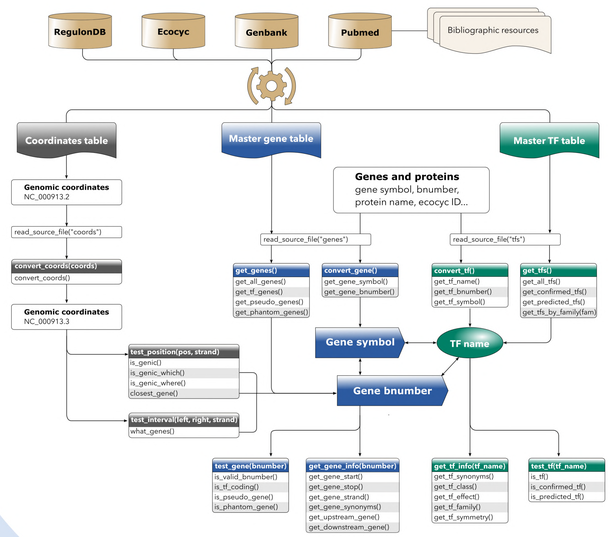
A neat & exhaustive collection of names and synonyms

fnr | b1334 | b1334 fnr ECK120000319 EG10325 ossA oxrA nirA nirR | 1,398,774 | 1,399,526 | reverse | FNR | FNR Fnr OssA OxrA NirA NirR PF00027 PF00325 NP_415850 P0A9E5 | DNA-binding transcriptional dual regulator FNR | Global Regulator | CRP family | covalent

lrp	b0889	alsB, b0889, ECK120000540, EG10547, ihb, livR, lrp, lrs, lss, lstR, mbf, oppI, rblA	932,595	933,089	forward	Lrp	AlsB, ECK120005016, Ihb,LivR, Lrp, Lrs, Lss, LstR, Mbf, NP_415409, OppI, P0ACJ0, RblA	DNA-binding transcriptional dual regulator Lrp Global Regulator AsnC family allosteric
dgoR	b4479	b3694, b3695, b4479, dgoR, ECK120004308, G7789, yidW	3,874,471	3,875,160	reverse	DgoR	DgoR, ECK120008747, G7789, P31460, YidW, YP_026239	DNA-binding transcriptional regulator DgoR Local Regulator
gyrA	b2231	b2231, ECK120000416, ECK2223, EG10423, gyrA, hisW, nalA, nfxA, norA, parD	2,866,559	2,867,551	reverse			
rpoS	b2741	abrD, appR, b2741, csi2, dpeB, ECK120000503, ECK2736, EG10510, katF, nur, otsX, rpoS, sigS	2,336,793	2,339,420	reverse			

- After querying several databases, **gene and TF names, synonyms and IDs** are gathered and organized
- Genes are indexed using a **reference symbol** and bnumber, TFs are indexed by their common name
- Alternative names and synonyms are kept in a **dictionary**, along with additional attributes

The complete framework



Use cases

- A random list of gene names is generated
- Genes are translated to their reference names and filtered
- Some of their attributes are retrieved

```
gene_list <- sample(get_all_genes_synonyms(), 100)              # get a random list of gene names

gene_table <- data.frame(gene_list) %>%
  mutate(symbol = convert_gene(gene_list, to = "symbol"),       # get symbol of reference
         bnumber = convert_gene(gene_list, to = "bnumber"),     # get bnumber of reference
         start = get_gene_start(bnumber),                       #
         stop = get_gene_stop(bnumber),                         # get gene coordinates & strand
         strand = get_gene_strand(bnumber)) %>%                 #
  filter(is_tf_gene(bnumber)) %>%                               # select only TF-coding genes
  mutate(tf_name = convert_tf(bnumber),                         # get TF common name
         family = get_tf_family(tf_name))                       # get TF attributes
```

gene_list
ydhM
G7326
yhiT
yeaM
ECK120000526
rtcR
ECK4055
b2087

→

gene_list	symbol	bnumber	start	stop	strand	tf_name	family
ydhM	nemR	b1649	1,726,023	1,726,622	+	NemR	TetR/AcrR
G7326	iscR	b2531	2,661,643	2,662,131	-	IscR	Rrf2
yhiT	gadE	b3512	3,658,366	3,658,893	+	GadE	LuxR/UhpA
yeaM	nimR	b1790	1,874,755	1,875,576	-	NimR	AraC/XylS
ECK120000526	lexA	b4043	4,257,115	4,257,723	+	LexA	LexA
rtcR	rtcR	b3422	3,558,268	3,559,866	+	RtcR	EBP
ECK4055	soxR	b4063	4,277,469	4,277,933	+	SoxR	MerR
b2087	gatR	b4498	2,169,693	2,171,727	-	GatR	NA

- Other genomic features can be manipulated
- Example: transcription units reported by Conway et al., 2014

```
TUs_Conway <- read.table("Conway_2014.tsv")          # load data

TUs_updated <- TUs_Conway %>%                        # extract TU coordinates and strand
  [some parsing] %>%
  mutate(start = convert_coords(left)                # update coordinates to current
         stop = convert_coords(right)                # genome version
         genes = what_genes(start, stop, strand))    # get genes contained in TU
```

Operon	Genes
P-832279:T-835449	dinG-ybiB
P-835542:T-836826	ybiC
T-1304820:P-1306695	cls-yciU
P-1306788:T-1307015	yciY
P-1504781:T-1506984	ydcP
T-1506717:P-1507161	yncJ

→

start	stop	strand	genes
833,056	836,226	+	dinG, ybiB
836,319	837,603	+	hcxB, ybiE
1,306,796	1,308,671	-	yciU, clsA
1,308,764	1,308,991	+	yciY
1,506,757	1,508,960	+	rlhA
1,508,693	1,509,137	-	yncJ

Coming next...

- A dictionary of *E. coli*'s orthologs in *Salmonella enterica*
- Conversion of genes, TFs and regulatory network b/w both species

Contact
- rioualenclaire@gmail.com
- www.rioualen.github.io
- www.ccg.unam.mx

Acknowledgements
- CR was granted a BioC2022 scholarship award
- CR is a doctoral student from the Programa de Doctorado en Ciencias Biomédicas, UNAM, and has received fellowship 929687 from CONACyT

References

- Conway T, Creecy JP, Maddox SM, Grissom JE, Conkle TL, et al (2014). Unprecedented high-resolution view of bacterial operon architecture revealed by RNA sequencing. mBio 2014;5:e01442-14.
- Wickham H, Girlich M (2022). tidyr: Tidy Messy Data. https://tidyr.tidyverse.org, https://github.com/tidyverse/tidyr

Chapter 2.

Building a comprehensive set of genomic features

Problematic

Despite extensive information in RegulonDB, we don't have an exhaustive panorama of *E. coli* K-12 operons, transcription units, promoters and terminators. Nonetheless, clarifying the genomic structure of *E. coli* is a necessary preliminary step in order to properly connect the pieces together and unravel its transcriptional regulatory network. Recently published high-throughput datasets can help us draw a comprehensive picture of *E. coli* genome composition, but challenges remain as the definitions for these genomic features are somewhat blurry, and handled differently depending on the source.

In this chapter, I show how I gathered high-throughput datasets from recent publications and databases, updated and standardized them together with the classic data from RegulonDB, in order to create a new integrated set of genomic features for *E. coli*.

Revising core concepts

Bacterial genomes possess a characteristic organization of their genes into so-called operons. They are defined as clusters of genes under the control of a single promoter and co-transcribed together into polycistronic RNA molecules. The concept of transcription unit was introduced to account for the existence of distinct transcripts and promoters present in one operon. They were defined as sets of one or several genes co-transcribed as polycistronic units, however their proper description and distinction with operons have remained somewhat unclear (Mejía-Almonte et al., 2020).

As knowledge grew, the necessity arose to revise original concepts and define new ones, in order to fit with the biological complexity of the bacterial genome. Here, I define a transcription unit as a physical entity made of a portion of the genome between given start and end coordinates, that contains a set of contiguous genes that share the same orientation and are co-transcribed into one single transcript. An operon is a conceptual object composed of one or several transcription units that share at least one gene, and consequently, all of the genes contained in said transcription units (Mejía-Almonte et al., 2020). This means that a given gene can be part of several distinct transcription units, but only one operon.

Besides, the concepts of promoter and terminator had to be clarified as well. Here, I consider that a promoter can contain one or several transcription start sites (within a maximum distance of 5 bp), and an operon can contain one or several promoters. A given transcription unit is associated to a specific TSS, which marks its start coordinate. Likewise, terminator regions can contain one or several distinct transcription termination sites, and operons can contain one or several terminator regions.

Transcription unit and co-transcribed genes unit sets

In order to build an exhaustive transcription unit set, the question arose of how to define them in terms of objects and their attributes. TUs are theoretically defined by a promoter and a terminator, however in databases they can be associated with one, several or none of them. TU coordinates can also be defined by the genes they contain and the transcripts they form, and those can be characterized by different experimental methods that do not necessarily have a single-nucleotide resolution. Therefore, multiple coordinates can be considered that potentially refer to the same biological object, and numerous redundant TUs can be generated. For these reasons, I defined two kinds of objects:

- transcription units are defined by their unique start and end coordinates and their direction or strand;
- co-transcribed genes (CTG) units are made of genes that are co-transcribed together as a polycistronic unit, at least once, regardless of coordinates (Figure 8).

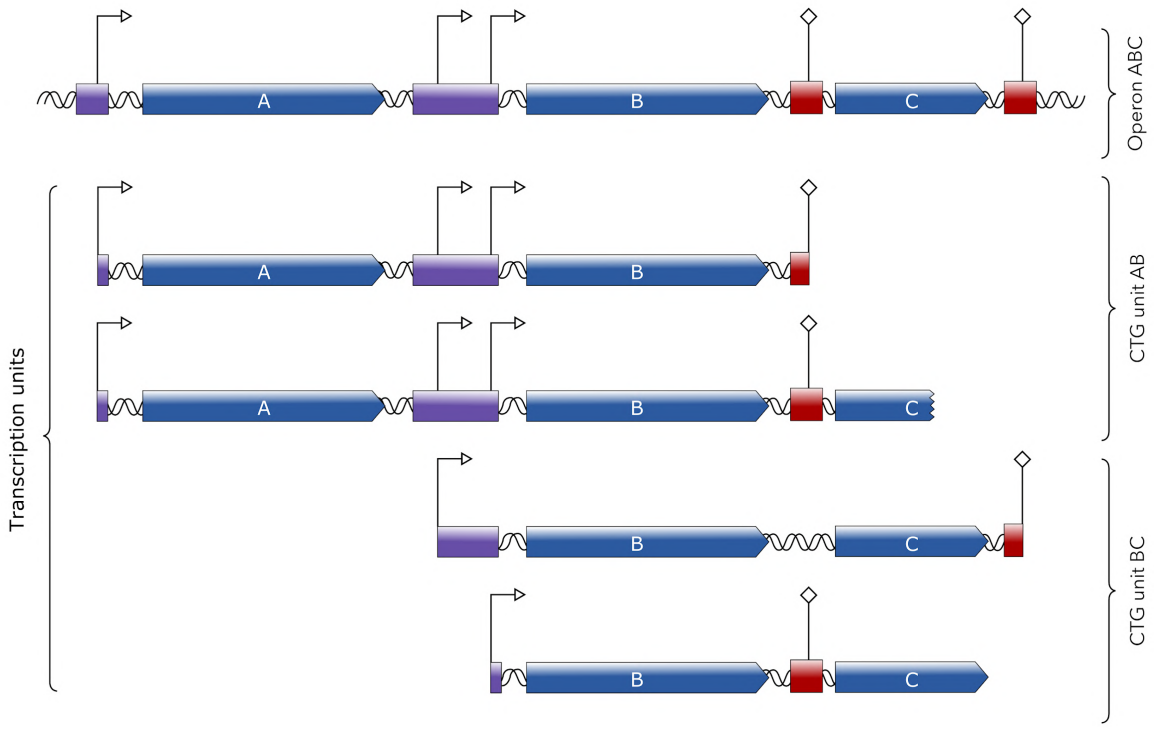


Figure 8. Definitions. A transcription unit is uniquely defined by its genomic coordinates and strand. Coordinates can be those of the associated TSS and TTS, or those of the leading and closing genes. A co-transcribed genes unit can be composed of one or several TUs, and is defined by its gene content. A given gene can be in several CTG sets, but no two CTG sets can contain exactly the same genes.

First, I retrieved 3,652 transcription units from RegulonDB, together with their associated promoters and terminators when available. Theoretically a transcription unit should be associated with a specific TSS and TTS, however in many cases the information is unknown or ambiguous. I defined transcription unit start and end coordinates using their TSS and TTS positions when available, and in their absence, the coordinates of their first and last genes. Second, I added 4,686 transcription units generated through SMRT-Cappable-seq technology (Yan et al., 2018) using two distinct growth conditions and two methods for determining the ending position: formal identification of a TTS (10% of reported TUs) or longest read coordinate (90% of reported TUs). Considering their coordinates, only one TU from RegulonDB was also present in the HT datasets, and 259 TUs from the HT dataset were present in more than one condition (Figure 9a). All of these TUs were given a "gene content" attribute, listing the genes entirely contained in each TU per their respective coordinates. Gene

names were homogenized using the EcoliGenes library (Chapter 1). When the start or end position of a TU fell inside a genic sequence, the corresponding gene was excluded from the TU gene content without impacting its coordinates, and only the entire genes were included. Finally, in order to get a full coverage of all genes of *E. coli*, the genes that were not included in any TU were made into 173 *orphan* TUs, with their start and end coordinates being those of the gene. All of the TUs were merged by coordinates and strand, amounting to a total of 8,221 unique transcription units.

Then, I derived the co-transcribed genes set from the TU set. Every group of CTG is made of genes that are present together in at least one TU, implying they can be co-transcribed together as a polycistronic unit. In practice, TUs that contain exactly the same gene content are grouped into CTG units, and their widest coordinates are kept for reference. This allowed to reduce the redundancy inside each dataset: the collection of TUs from RegulonDB lowered to 3,053 unique CTGs, and the HT collection lowered from 4,686 TUs to 2,326 CTGs, mostly due to the numerous TUs that don't have a precise terminator site associated. Overall, a total of 4,283 CTG units compose the whole set, which dramatically lowers the redundancy observed in the initial TU set (Figure 9b), of which 29% were not initially present in RegulonDB.

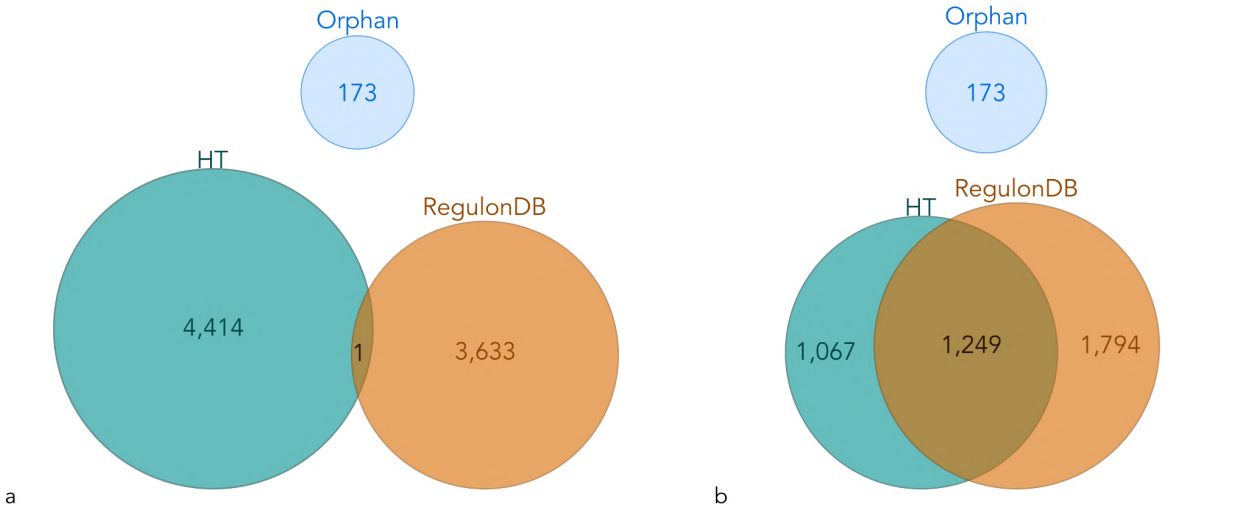


Figure 9. Overlap between classic and high-throughput data. **a.** Overlap between transcription units defined by their unique coordinates. **b.** Overlap between co-transcribed gene units defined by their unique gene content.

Briefly, out of 8,211 unique TUs, I reduced the total number to almost half of this amount by taking into account their gene content, reaching a total of 4,283 CTGs. This drastic change can be explained by technical and biological factors: some experiments don't allow a precise identification of the TSS and/or TTS, leading to ambiguous coordinates, and numerous sets of co-transcribed genes are associated with several distinct TSS and/or TTS. This shows that in *E.coli* gene expression diversity is mostly achieved by alternate regulation more than alternate transcription unit membership, a strategy that is frequently observed in bacterial genomes.

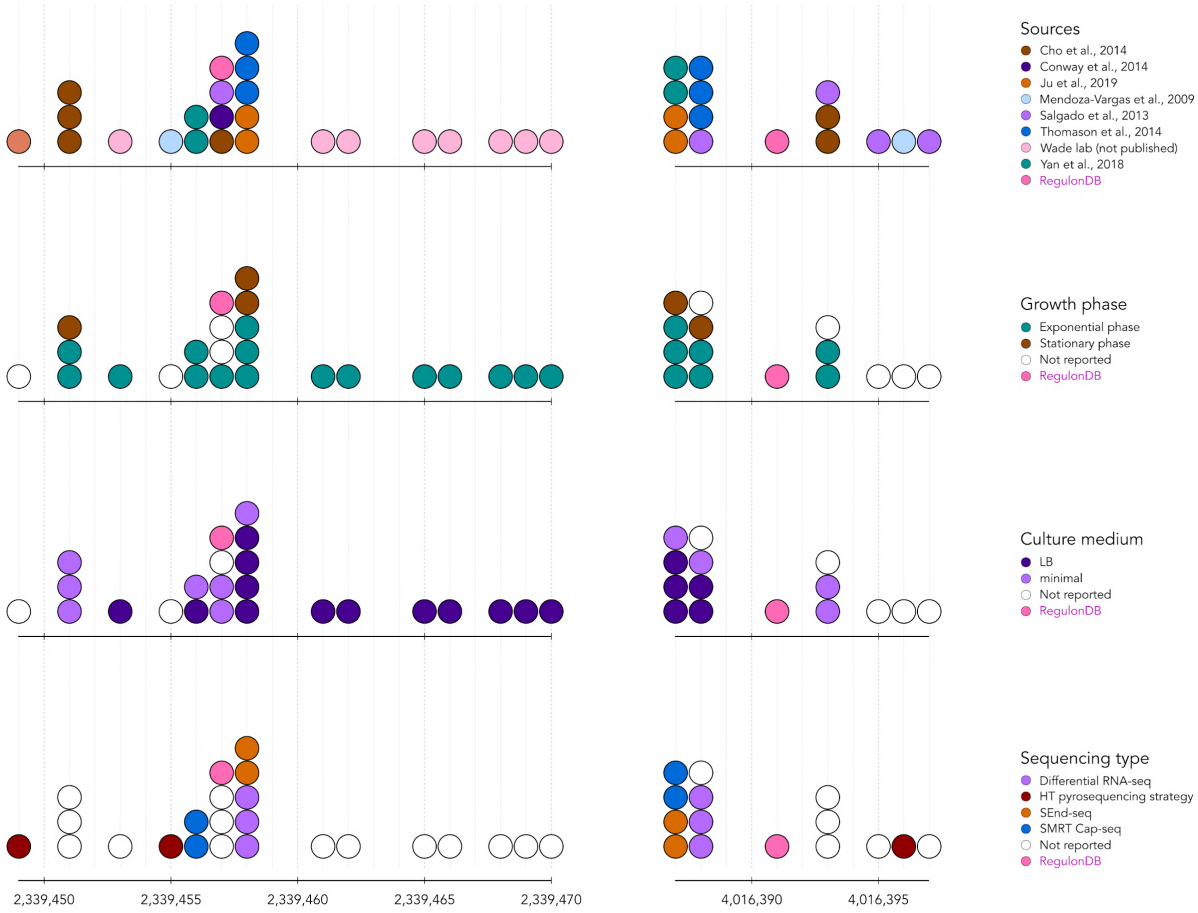
Promoter and TSS sets

As detailed above, transcription start sites are defined by a unique position, while promoters are small regions containing one or several TSSs, usually separated by less than 5 bp from one another. However, those concepts have been interchangeably used in the literature.

The TSS set was built from the data available in RegulonDB, and completed with several HT-based datasets from independent sources (Mendoza-Vargas et al., 2009; Salgado et al., 2013; Cho et al., 2014; Thomason et al., 2015; Yan et al., 2018; Wade laboratory, not published). Datasets prior to 2015, based on an older genome assembly (NC_000913.2) were updated to the latest genomic coordinates (NC_000913.3) using the EcoliGenes library (Chapter 1). TSSs from all of the datasets were merged when they shared the same position and strand, reducing their total number from 65,409 to 28,987; and were homogenized into a common format with the most relevant attributes.

The promoter set was derived from the unmerged TSS set. Associating different TSSs to a single promoter is not a trivial process. Depending on the experimental method used, the precision of a TSS position can vary greatly, and biologically distinct TSSs can be present in the same promoter (Mejía-Almonte et al., 2020). Furthermore, several promoters associated with distinct sigma factors can overlap spatially, and even share TSSs, resulting in promoter *regions*. Here, I built the promoter objects by grouping all of the TSSs that were at most 5 bp away from one another regardless of the associated sigma factor, which is not always reported, using a sliding window. In total, 23,316 promoters were built, with an average length of 1.4 bp and a maximum length of 22 bp. Though it is considered that a promoter should be at most 5 bp long, about 1% of the

collection of promoters obtained is larger, and distinguishing potential overlapping promoters would require further analysis (Figure 10). On average, promoters included 1.24 TSSs, consistent with a previous study that reported an average of 1.6 TSSs per promoter (Cho et al., 2009), and a maximum of 29 TSSs.



a. Promoter gyrAp (24 TSSs, length 22 bp)

b. Promoter udpP (15 TSSs, length 11 bp)

Figure 10. Composition of two of the largest promoters, and source of their respective TSSs (top panel). TSSs can be further distinguished by their associated growth conditions and experimental methods (bottom 3 panels).

Briefly, current knowledge amounts to 28,987 different TSSs in the *E.coli* genome, which might be reduced to around 23,000 functionally distinct promoter regions, making on average around 5 promoter regions per gene. Note that these numbers come from different growth conditions and experimental setups, and therefore the global picture could change as new datasets become available.

Binding sites set

I retrieved all of the transcription factor binding sites curated from RegulonDB version 10.8 with *strong* evidence, and merged them using their coordinates, TF-coding gene bnumber, and effect (positive or negative). When the TF was a heterodimer I created two entries, one per coding-gene bnumber. This resulted in 2,599 TFBSs associated with 185 transcription factors. Those numbers hide significant disparities: most TFs have less than 5 associated binding sites, while 10 TFs have more than 50 binding sites each, and account for a total of 1,252 binding sites, or 48% of the whole set (Figure 11a). Roughly half are reported to be repressors, and half activators (Figure 11b). As it is well known, most binding sites are found in the intergenic regions of the genome, and in particular between -400 and +50 bp relative to gene start codons (Figure 11c-d).

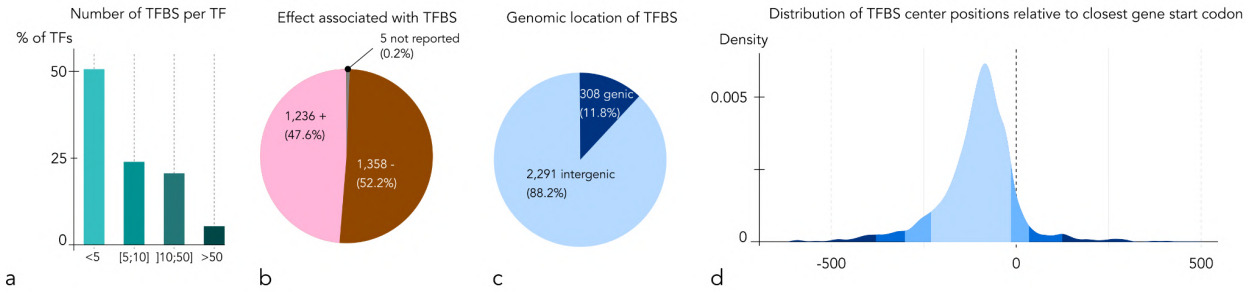


Figure 11. Statistics associated with the binding sites set. a. Number of binding sites associated with TFs. **b.** Effect associated with binding sites (activation in pink, repression in brown). **c.** Genomic location of binding sites. **d.** Binding sites distribution relative to gene start position.

Unified set of genomic features

I generated unique custom identifiers for each object of each set, and created additional tables to connect them with one another, in the form of a small database of its own (Table 1; Figure 12). This database of features can be readily connected to RegulonDB through the gene master table described in Chapter 1.

Type of object	Size of the set
TSS	28,987
promoter	23,316
TU	8,221
CTG	4,283
TFBS	2,599

Table 1. Content of the updated *E. coli* feature set.

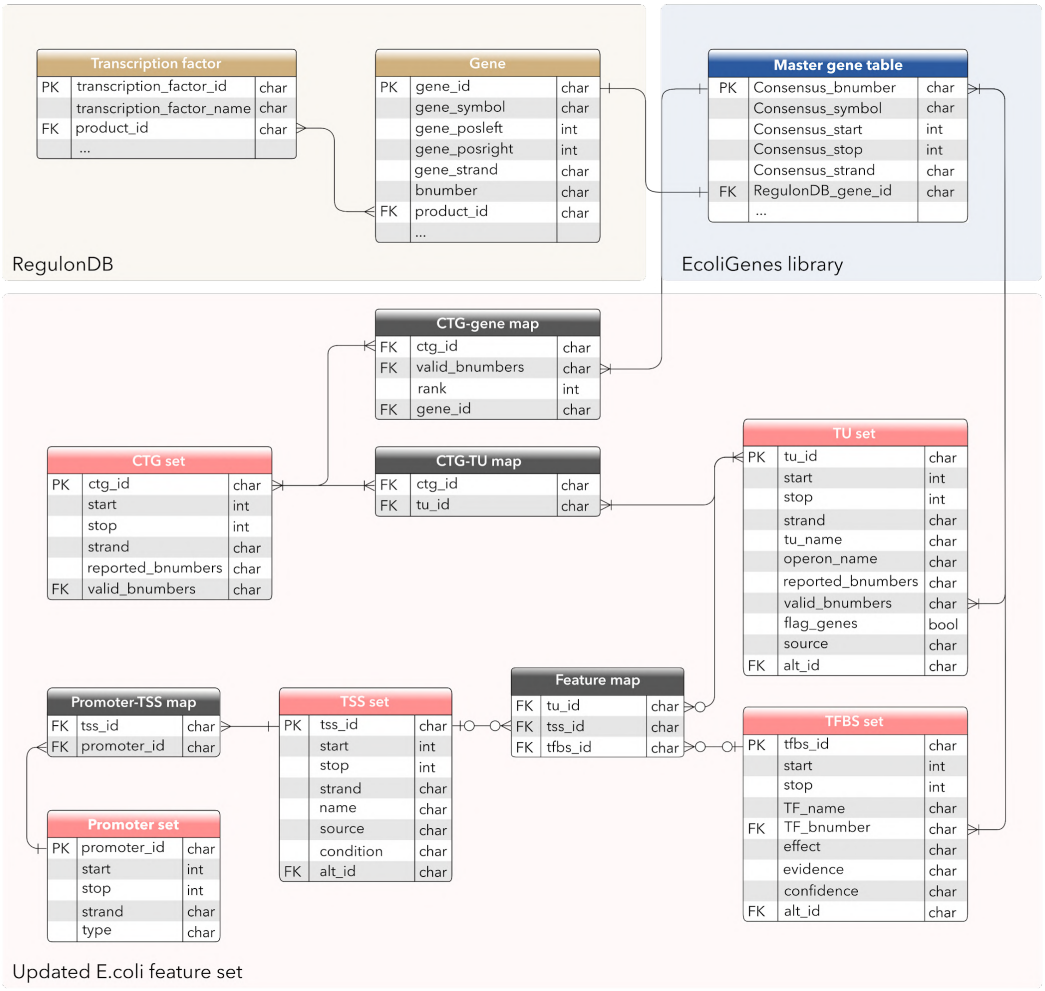


Figure 12. The updated *E. coli* feature set. Five sets of objects are connected together via custom identifiers: transcription units, CTG units, transcription start sites, promoters and TF binding sites (pink frame). This small independent database can be connected to RegulonDB (beige frame) through gene identifiers and EcoliGenes (blue frame).

Availability

Github

The updated *E. coli* feature set is available at Github:
https://github.com/rioualen/Ecoli_feature_set

Chapter 3.

Building tools for high-throughput data analysis

Problematic

Next-generation sequencing technologies enable the characterization of gene regulation mechanisms at an unprecedented scale. Transcription factor binding sites can be identified genome-wide with ChIP-seq, and RNA-seq makes it possible to quantify all transcripts from a given cell, thus revealing gene expression and TUs. However, the analysis of their respective outputs, under the form of sequenced reads, requires multiple processing steps that can be realized using a variety of tools and parameters, and can represent a challenge when dealing with diverse experimental setups and strategies.

I developed a collection of workflows that enable the chaining of the successive steps to be performed. With a proper setup, these workflows can be customized with flexibility to cater to the objective of the analysis to be performed, but also ensure the full traceability and reproducibility of the results. This work was published as a protocol (Rioualen et al., 2019).

Workflows for the analysis of ChIP-seq and RNA-seq data

The framework snakemake (Mölder et al., 2021) was conceived to build pipelines ensuring the full portability and reproducibility of the analyses performed and their subsequent results. Based on the python programming language and GNU make concepts, it defines workflows as sets of rules characterized by their input and output files, or dependencies, and optional parameters. The first rule of a workflow, by

convention, defines the final targets to be produced, and by deduction, the list of rules to be executed according to inner dependencies (Figure 13).

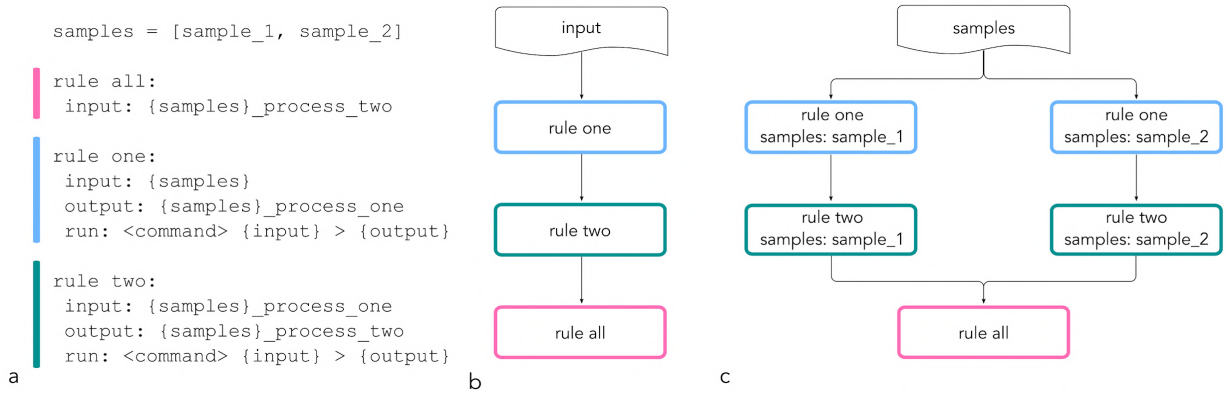


Figure 13. Schematic structure of a snakemake workflow. a. Workflow example. **b.** Dependency graph (also called rulegraph). **c.** Directed acyclic graph of the workflow with parallelization.

I developed workflows and rules organized into a library called "SnakeChunks" and published in the form of a protocol (Rioualen et al., 2019). It comprises more than 60 rules to perform numerous tasks using a variety of tools (Figure 14a). Those rules can be linked to one another via their respective input, output, intermediary files in order to compose workflows (Figure 14b). Workflows can be customized by selecting different tools and optional parameters for each inner step, using external configuration files (Figure 14c). Additionally, the library contains ready-to-use workflows dedicated to perform quality control analyses, read mapping, ChIP-seq analysis, RNA-seq analysis and integration of binding and expression data from RNA-seq and ChIP-seq analyses (Table 2).

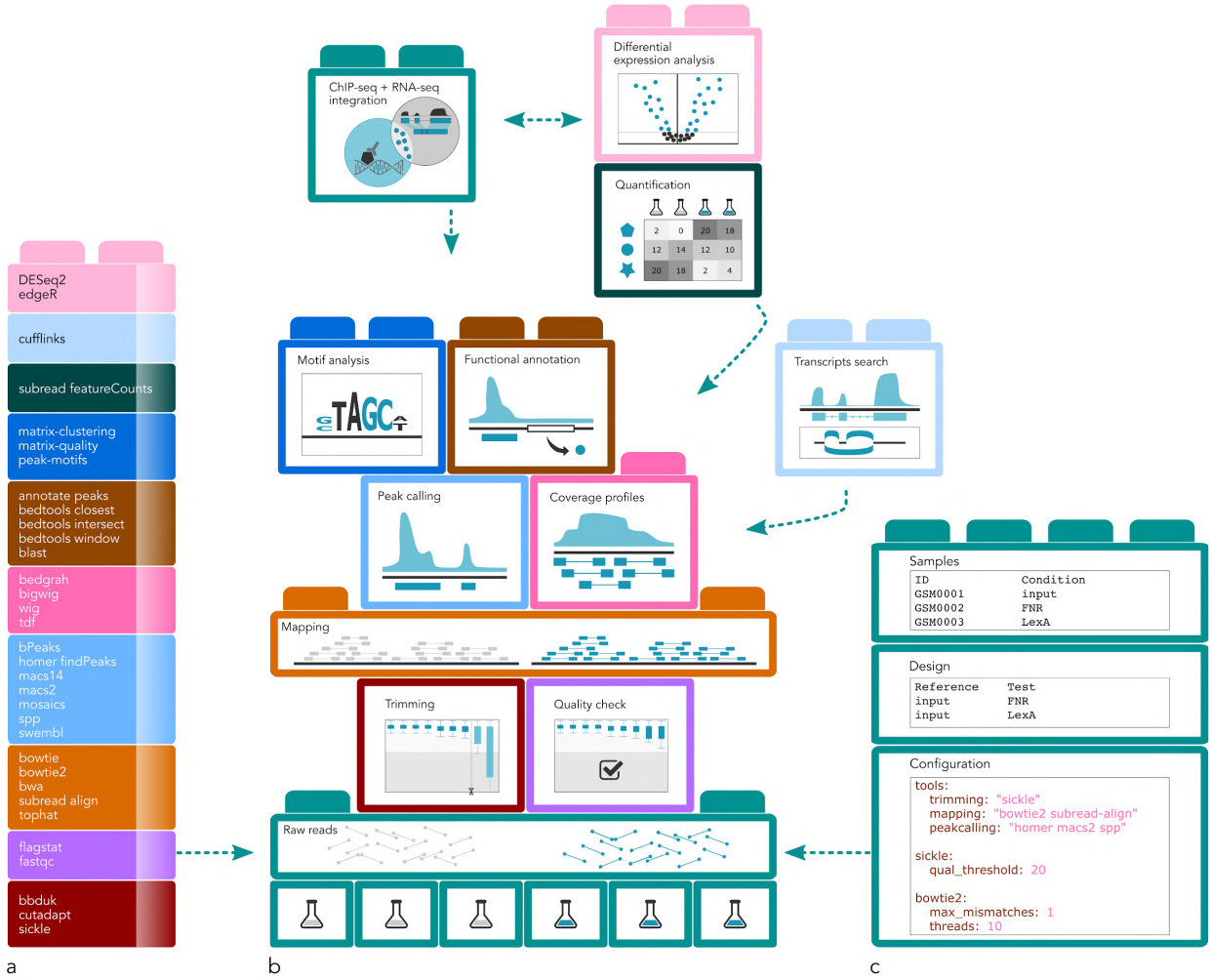


Figure 14. SnakeChunks framework. a. Selected list of rules available by category. **b.** All rules can be assembled to create custom workflow structures. **c.** Metadata and parameters can be specified in separate files for further customization and traceability. Adapted from Rioualen et al., 2019.

workflow	category	rule	input	output
quality control	formatting	sra_to_fastq	sra	fastq
quality control	trimming	sickle	fastq	fastq
quality control	trimming	bbduk	fastq	fastq
quality control	trimming	cutadapt	fastq	fastq
quality control	quality control	fastqc	fastq, bam	html
quality control	quality control	bam_stats	bam	txt
quality control	quality control	multiqc	*	html

workflow	category	rule	input	output
mapping	mapping	bowtie_index	fasta	fai
mapping	mapping	bowtie2_index	fasta	fai
mapping	mapping	bwa_index	fasta	fai
mapping	mapping	hisat2_index	fasta	fai
mapping	mapping	index_fasta	fasta	fai
mapping	mapping	subread_index	fasta	fai
mapping	mapping	bowtie	fastq+fai	bam
mapping	mapping	bowtie2	fastq+fai	bam
mapping	mapping	bwa	fastq+fai	bam
mapping	mapping	hisat2	fastq+fai	bam
mapping	mapping	tophat	fastq+fai	bam
mapping	mapping	subread_align	fastq+fai	bam
mapping	mapping	bam_by_name	bam	bam
mapping	mapping	bam_by_pos	bam	bam
mapping	mapping	split_bam_by_strands	bam	bam
mapping	mapping	index_bam	bam	bai
mapping	coverage	coverage_bedgraph	bam	bedgraph
mapping	coverage	coverage_bedgraph_stranded	bam	bedgraph
mapping	coverage	coverage_bigwig	bam	bigwig
mapping	coverage	coverage_wig	bam	wig
mapping	coverage	bedgraph_to_bigwig	bedgraph	bigwig
mapping	coverage	bedgraph_to_tdf	bedgraph	tdf
mapping	mapping	bam_to_bed	bam	bed
mapping	mapping	sam_to_bam	sam	bam
ChIP-seq	peak calling	bPeaks	bam	bed
ChIP-seq	peak calling	homer	bam	bed
ChIP-seq	peak calling	macs14	bam	bed
ChIP-seq	peak calling	macs2	bam	bed
ChIP-seq	peak calling	mosaics	bam	bed
ChIP-seq	peak calling	spp	bam	bed
ChIP-seq	peak calling	swembl	bam	bed
ChIP-seq	peak annotation	annotate_peaks	bed+fasta+gtf	tab
ChIP-seq	peak annotation	bedops_intersect	bed	bed
ChIP-seq	peak annotation	bedops_peaks_vs_sites	bed	bed
ChIP-seq	peak annotation	bedtools_closest	bed+gff3	bed
ChIP-seq	peak annotation	bedtools_intersect	bed+gff3	bed
ChIP-seq	peak annotation	bedtools_window	bed+gff3	bed
ChIP-seq	peak annotation	peaks_vs_tfbs	bed	bed
ChIP-seq	formatting	bed_to_fasta	bed	fasta
ChIP-seq	formatting	getfasta	bed	fasta
ChIP-seq	motif analysis	dyad_analysis	fasta	html+transfac

workflow	category	rule	input	output
ChIP-seq	motif analysis	peak_motifs	fasta	html+transfac
ChIP-seq	motif analysis	matrix_clustering	transfac	html
ChIP-seq	motif analysis	matrix_quality	transfac+fasta	html
ChIP-seq	RegulonDB	regulondb_download	url	tab
ChIP-seq	RegulonDB	regulondb_get_matrix	<TF name>	transfac
ChIP-seq	RegulonDB	regulondb_get_tfbs	<TF name>	bed
RNA-seq	transcript detection	cufflinks	bam+gtf	gtf
RNA-seq	differential expression	subread_featureCounts	bam+gtf	tab
RNA-seq	differential expression	DESeq2	tab	tab
RNA-seq	differential expression	sartools_targetfile	tab	tab
RNA-seq	differential expression	sartools_DESeq2	tab	html+tab
RNA-seq	differential expression	sartools_edgeR	tab	html+tab
misc.	formatting	get_chrom_sizes	fasta	tab
misc.	formatting	gunzip	zip	*
misc.	formatting	gzip	*	zip
misc.	formatting	md5sum	*	*

Table 2. List of rules and workflows available in the SnakeChunks library, and their respective input and output file formats, defining their mutual dependencies.

This library and its published protocol present a methodological development undertaken before the PhD and finalized during the first year, which I used for subsequent analyses of high-throughput data towards the aim of *E. coli*'s transcriptional regulatory network completion (Chapter 4).

Reference & availability

Github

The SnakeChunks library is available for download and use through github: https://github.com/SnakeChunks/SnakeChunks

Publication

This work was used as part of external collaborations before it was published under the form of a protocol.

Rioualen, C., Charbonnier-Khamvongsa, L., Collado-Vides, J., & van Helden, J. (2019). Integrating Bacterial ChIP-seq and RNA-seq Data With SnakeChunks. Current Protocols in Bioinformatics, 66(1), e72. https://doi.org/10.1002/cpbi.72

Desvillechabrol, D., Legendre, R., **Rioualen, C.**, Bouchier, C., van Helden, J., Kennedy, S., & Cokelaer, T. (2018). Sequanix: A dynamic graphical interface for Snakemake workflows. Bioinformatics (Oxford, England), 34(11), 1934–1936. https://doi.org/10.1093/bioinformatics/bty034

Tsagmo Ngoune, J. M., Njiokou, F., Loriod, B., Kame-Ngasse, G., Fernandez-Nunez, N., **Rioualen, C.**, van Helden, J., & Geiger, A. (2017). Transcriptional Profiling of Midguts Prepared from Trypanosoma/T. congolense-Positive Glossina palpalis palpalis Collected from Two Distinct Cameroonian Foci: Coordinated Signatures of the Midguts' Remodeling As T. congolense-Supportive Niches. Frontiers in Immunology, 8, 876. https://doi.org/10.3389/fimmu.2017.00876

Integrating Bacterial ChIP-seq and RNA-seq Data With SnakeChunks

Claire Rioualen,[1,2] Lucie Charbonnier-Khamvongsa,[1] Julio Collado-Vides,[2,3] and Jacques van Helden[1,4,5]

[1]Aix-Marseille University, INSERM, Laboratory of Theory and Approaches of Genome Complexity (TAGC), Marseille, France
[2]Programa de Genómica Computacional, Centro de Ciencias Genómicas, Universidad Nacional Autónoma de México, Cuernavaca, México
[3]Department of Biomedical Engineering, Boston University, Boston, Massachusetts
[4]Institut Français de Bioinformatique (IFB), UMS 3601-CNRS, Université Paris-Saclay, Orsay, France
[5]Corresponding author: *Jacques.van-Helden@univ-amu.fr*

Next-generation sequencing (NGS) is becoming a routine approach in most domains of the life sciences. To ensure reproducibility of results, there is a crucial need to improve the automation of NGS data processing and enable forthcoming studies relying on big datasets. Although user-friendly interfaces now exist, there remains a strong need for accessible solutions that allow experimental biologists to analyze and explore their results in an autonomous and flexible way. The protocols here describe a modular system that enable a user to compose and fine-tune workflows based on SnakeChunks, a library of rules for the Snakemake workflow engine (Köster and Rahmann, 2012). They are illustrated using a study combining ChIP-seq and RNA-seq to identify target genes of the global transcription factor FNR in *Escherichia coli* (Myers et al., 2013), which has the advantage that results can be compared with the most up-to-date collection of existing knowledge about transcriptional regulation in this model organism, extracted from the RegulonDB database (Gama-Castro et al., 2016). © 2019 by John Wiley & Sons, Inc.

Keywords: ChIP-seq • *Escherichia coli* K-12 • FAIR Guiding Principles • reproducible science • RNA-seq • workflow

How to cite this article:
Rioualen, C., Charbonnier-Khamvongsa, L., Collado-Vides, J., & van Helden, J. (2019). Integrating bacterial ChIP-seq and RNA-seq data with SnakeChunks. *Current Protocols in Bioinformatics*, e72. doi: 10.1002/cpbi.72

INTRODUCTION

Next-generation sequencing (NGS) technologies enable the characterization of biological gene regulation at an unprecedented scale. Transcription-factor binding can be characterized at the genome scale by chromatin immunoprecipitation with DNA sequencing (ChIP-seq), whereas RNA sequencing (RNA-seq) makes it possible to quantify all transcripts.

The analysis of sequenced reads requires a number of successive bioinformatics processing steps, organized into workflows. A workflow, or pipeline, is defined as a chaining of commands and tools applied to a set of data files, such that the output of a given step is used as input for the subsequent one (Fig. 1). Ideally, the experimental design should from the outset take into account a perspective on the bioinformatics analyses that will enable relevant information to be extracted from the raw data. Biological samples are

CURRENT PROTOCOLS
A Wiley Brand
Current Protocols in Bioinformatics e72
Published in Wiley Online Library (wileyonlinelibrary.com).
doi: 10.1002/cpbi.72

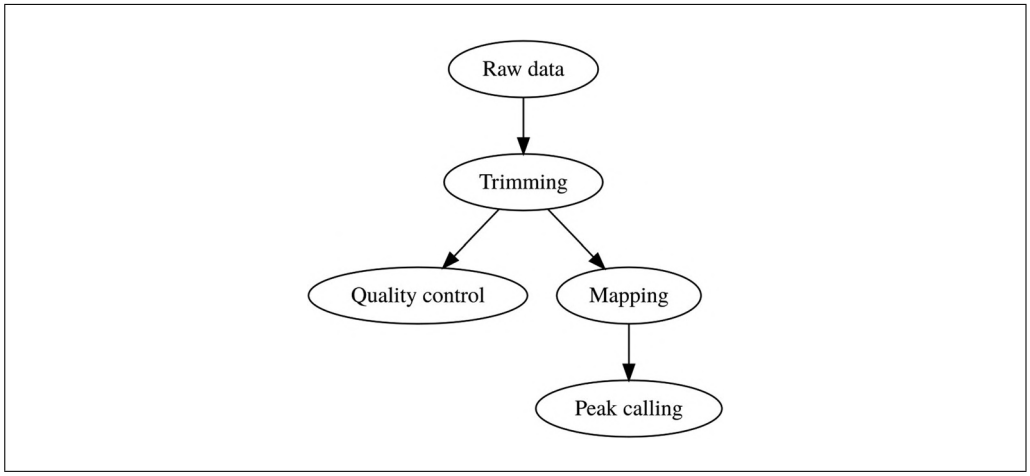


Figure 1 Schematic wiring of a basic workflow for ChIP-seq analysis.

subject to variation, and replication is thus essential to make it possible to estimate the statistical significance of the final results and to ensure an appropriate tradeoff between sensitivity and specificity. It is also necessary, as in any other biological experiment, to carefully define the control conditions that will distinguish signal from noise (see Commentary for more details).

Exploitation of the data by properly implemented bioinformatics workflows (with comprehensive specification of the tools and their versions and selection of parameters) is crucial to ensuring the traceability and reproducibility of the results from the raw data. Following a defined workflow also makes it possible to perform identical operations on dozens of samples, using powerful computing infrastructures when necessary. Snakemake (Köster & Rahmann, 2012) is a software conceived for building such workflows. Based on the Python language, it inherits concepts from GNU make (*https://www.gnu.org/software/make*): a workflow is defined by a set of rules, each defining an operation characterized by its inputs, outputs, and parameters, and a list of target files to be generated through these operations.

SnakeChunks is a library of workflows using the Snakemake framework and designed for the analysis of ChIP-seq and RNA-seq data. It includes rules for the quality control of sequencing reads, removal of adapters and trimming of low-quality bases, read mapping on a reference genome, peak calling to detect local enrichment of reads resulting from the binding of a transcription factor, gene-wise quantification of RNAs, and differential gene expression analysis (Fig. 2A).

The SnakeChunks library has been used to analyze RNA-seq data from *Mus musculus*, *Drosophila melanogaster*, *Saccharomyces cerevisiae*, and *Glossina palpalis* (Tsagmo Ngoune et al., 2017) and from *Desulfovibri desulfuricans* (Cadby et al., 2017), as well as ChIP-seq data from *Arabidopsis thaliana* (Castro-Mondragon, Rioualen, Contreras-Moreira, & van Helden, 2016). We illustrate here its use on combined RNA-seq and ChIP-seq data from *Escherichia coli* (Myers et al., 2013).

Since the initial description of the operon structure (Jacob & Monod, 1961), *E. coli* K-12 has been a model organism of reference for the study of gene regulation, resulting in thousands of publications reporting information about around 200 of the total ∼300 transcription factors (TFs) identified in its genome (Blattner et al., 1997; Pérez-Rueda & Collado-Vides, 2000). Detailed information about TFs and their binding sites, binding motifs, target genes, and operons has been collected for three decades in RegulonDB, the database on the transcriptional regulation in *E. coli* (Gama-Castro et al., 2016), by manual curation of publications based on low-throughput experiments. Nonetheless, a

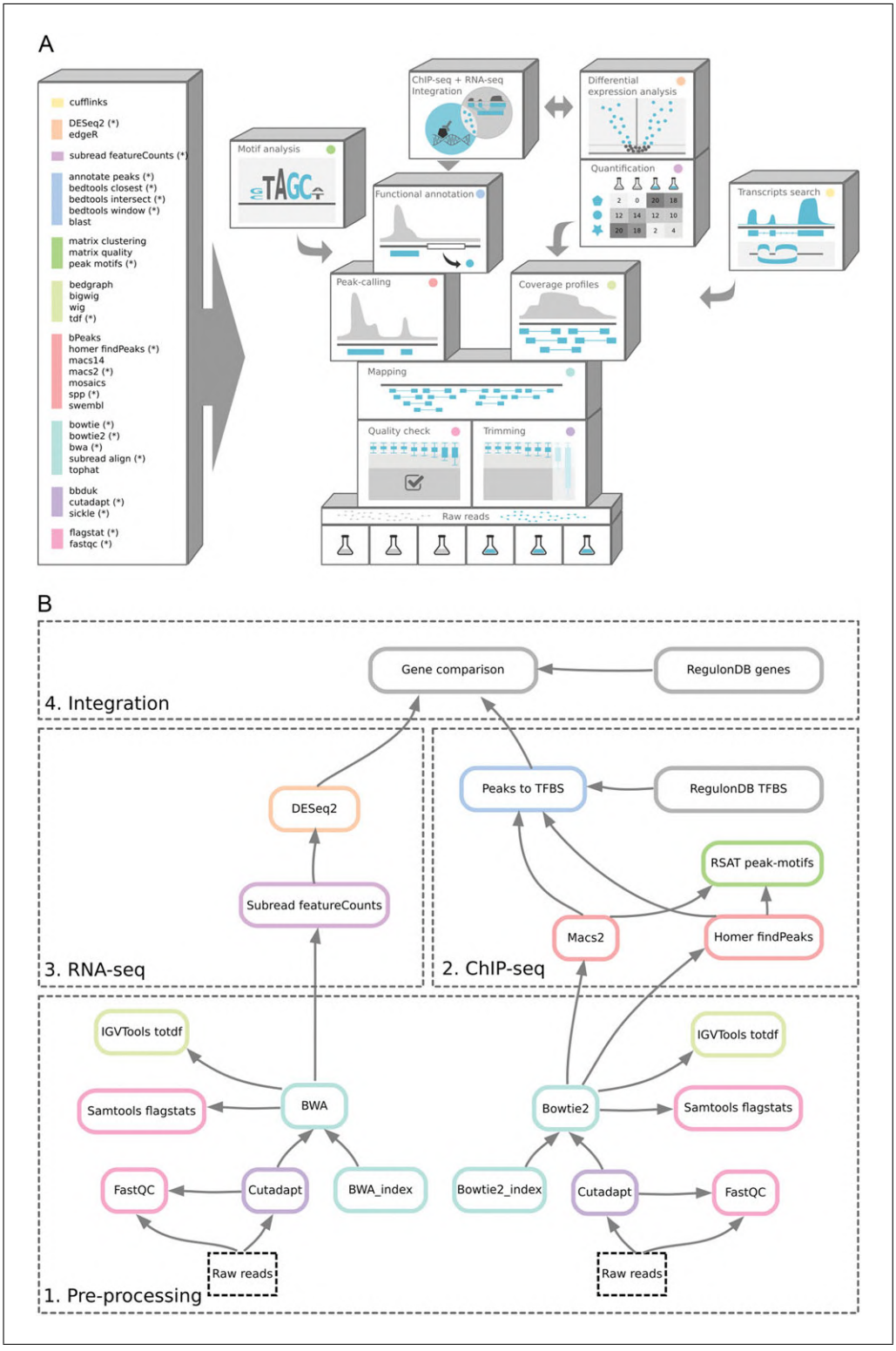


Figure 2 Organization of the SnakeChunks library. (**A**) Principle of the SnakeChunks library. The library is built around a set of Snakemake rules that can be used as building blocks to build workflows in a modular way. Each rule makes it possible to perform a given type of operation with a given tool. A given operation can also be done with alternative tools, as denoted by the color code in list of rules (left side) and on the building bricks. The rules marked with an asterisk (*) are currently supported by Conda. (**B**) Schematic flowchart of the workflows described in this unit.

good deal of information remains to be discovered to provide a global, comprehensive picture of the regulatory network of even this best-characterized model organism. NGS technologies enable the characterization of biological regulation at an unprecedented scale, and have been widely adopted by research communities. ChIP-seq gives insight into regulatory mechanisms by providing genome-wide binding locations for transcription factors, whereas RNA-seq provides information about the functional implications of regulation by measuring the level of transcription of all genes under different conditions.

ChIP-seq publications initially focused on human and metazoan models (PubMed currently returns ∼1,600 ChIP-seq studies for *Homo sapiens* and more than 2,000 for *M. musculus*), and a surprisingly small number of factors were characterized by ChIP-seq in *E. coli* (44 entries in PubMed). However, systematic studies have led to the characterization of 50 transcription factors of *Mycobacterium tuberculosis* (Galagan et al., 2013), and similar projects are on the way for other bacteria, including *E. coli*. The protocols described here address the foreseeable needs of microbiologists undertaking projects based on ChIP-seq, RNA-seq, or both together to analyze bacterial regulation. Those are illustrated by a case study based on a genome-scale analysis of the FNR transcription factor (Myers et al., 2013), a DNA-binding protein that regulates a large family of genes involved in cellular respiration and carbon metabolism during anaerobic cell growth.

This unit is organized as follows.

- Strategic Planning: installation and configuration of the software environment (Conda environment, software tools, SnakeChunks library, and reference genome).
- Basic Protocol 1: preprocessing, which includes quality control, trimming, and mapping of the raw reads on the reference genome. This protocol is illustrated for the case of a ChIP-seq study but can be applied to RNA-seq data as well.
- Basic Protocol 2: analysis of ChIP-seq data: peak calling, assignation of peaks to genes, motif discovery, and comparison between ChIP-seq peaks and sites annotated in RegulonDB.
- Basic Protocol 3: analysis of RNA-seq data: preprocessing (as in Basic Protocol 1), transcript quantification (counts per gene), and detection of differentially expressed genes.
- Basic Protocol 4: integration of ChIP-seq and RNA-seq results: comparison between genes associated with the ChIP-seq peaks, differentially expressed genes reported by transcriptome analysis, and experimentally proven TF target genes annotated in RegulonDB, as well as visualization of the results using a genome browser.
- Alternate Protocol: running of the RNA-seq workflow with the user-friendly graphical interface Sequanix.
- Support Protocol: customization of the ChIP-seq workflow parameters.

The basic protocols are conceived in a modular way (Fig. 2B). In particular, ChIP-seq and RNA-seq analyses can be done separately.

NECESSARY RESOURCES

Computer Resources

This protocol runs on any Unix system (Linux, Mac OS X). Memory and CPU requirements depend on the volumes of data being handled. The study cases have been tested on Ubuntu 14.04, 16.04, and 18.04 (4 CPUs, 16 Gb RAM), on Centos 6.6, and on Mac OSX High Sierra (4 CPUs, 16 Gb RAM).

The full procedure uses ~60 Gb of disk space, including ~5 Gb for the installation of the software environment (Conda, libraries, and tools), ~15 Gb of downloaded raw reads (compressed fastq files, genome annotations), and ~40 Gb for the intermediate and final result files.

The total processing time for all tasks is ~12 h, of which 45% is spent on read mapping and 33% on trimming RNA-seq samples. This time might be further reduced by parallelizing some tasks on a multi-CPU server or cluster (on our four-core configurations, the analyses were completed in ~3 h).

Conda

Conda is an open-source package and environment management system used to automate the installation of all the software components required by the workflows. It greatly facilitates the installation of software tools from multiple sources on different Unix operating systems (Linux and Mac OS X). In addition, the installation and use of all software tools inside a custom environment ensures their isolation from the hosting system and prevents potential clashes with existing tools and libraries.

Conda should be installed prior to the execution of the protocols. It comes in two different versions, Anaconda and Miniconda. We recommend using Miniconda, which takes less disk space and makes it possible to install only the required software. Instructions can be found here: *https://conda.io/docs/user-guide/install/index.html*.

Make sure that the folder containing the Conda executable is added to your $PATH variable. This can be done automatically during the execution of the Miniconda installation script, or later by adding the following command to the bash profile (file `~/.bash_profile`).

```
export PATH=$PATH:~/miniconda3/bin/
```

You now need to log out and open a new terminal session in order for the path to be updated.

Other Software

In the protocols, we use the "tree" software to display the structure of folders and included files in the Unix terminal. This software is not technically required for the analysis, but offers a convenient way to check the proper organization of the files in the shell. Its installation can vary depending on the operating system or Linux distribution. Here are examples of tree installation with some popular package management systems.

Linux Ubuntu: `sudo apt-get install tree`
Linux CentO: `sudo yum install tree`
Mac OS X: `brew install tree`

IMPORTANT NOTE: *Throughout the following protocols, the instructions (text in Courier font) should be typed or copy-pasted in a terminal.*

STRATEGIC PLANNING

Configuration of the Conda Environment

This section provides a succession of Unix commands that enable a user to configure Conda, create a specific environment, install the required software (Snakemake and NGS tools), and download the reference genome and annotations (in our case, *E. coli* K-12 MG1655, release 37). Much of this procedure needs to be done only once, when first

setting up the environment; steps 3, 5, and 7 then need to be repeated for each session (see annotation to step 10 for details).

1. Configure Conda.

```
conda config --add channels r;
conda config --add channels defaults;
conda config --add channels conda-forge;
conda config --add channels bioconda
```

IMPORTANT NOTE: These commands must be typed in the precise order indicated above, which defines the priorities for packages that exist in several channels. Conda may issue warnings, which can be ignored, when some of the channels are already present — we intentionally re-add these channels in order to place them in the right order of precedence.

2. Create an empty SnakeChunks environment using Python version 3.6.

```
conda create --name snakechunks_env python=3.6
```

3. Activate the environment.

   This must be done for each new analysis session.

```
source activate snakechunks_env
```

Check that the environment is active: i.e., that the Unix prompt is prepended by "`(snakechunks_env)`".

4. Install Snakemake and some required software tools in the Conda environment: GNU make software, Python panda library, and the Integrative Genomics Viewer (IGV).

```
conda install make snakemake=5.1.4 igv=2.4.9 pandas=
  0.23.4
```

5. Define an environment variable with the directory for this analysis.

   This must be done for each new analysis session (alternatively, you can declare it in your bash profile).

```
export ANALYSIS_DIR=$HOME/FNR_analysis
```

6. Create the analysis directory.

```
mkdir -p $ANALYSIS_DIR
```

7. Set the current working directory to the analysis directory.

   This must be done for each new analysis session.

```
cd $ANALYSIS_DIR
```

8. Download the SnakeChunks library from GitHub. We recommend keeping a copy of the library in the analysis directory to ensure consistency and reproducibility. The latest version of the SnakeChunks library can be downloaded easily with the following Git command.

```
git clone https://github.com/SnakeChunks/SnakeChunks.
  git
```

IMPORTANT NOTE: The SnakeChunks code will continue evolving with time. For the sake of backward compatibility, we froze the precise version of the library used at the time of publication of this article. This version can be downloaded with the following command.

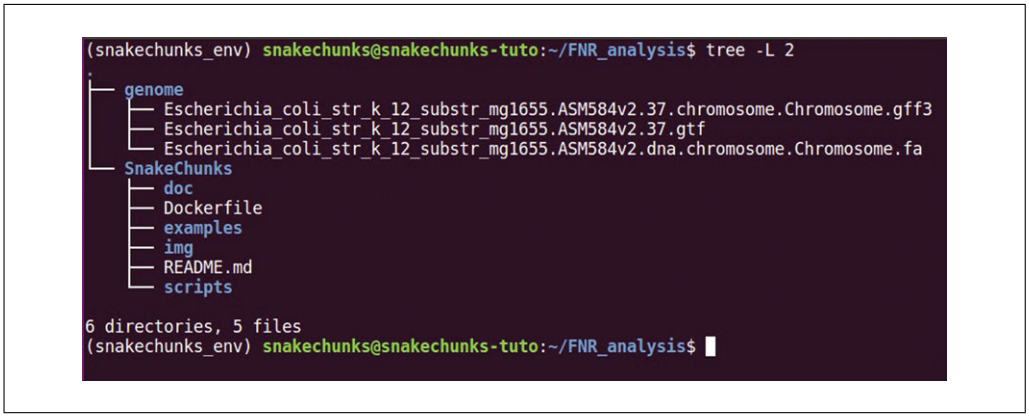


Figure 3 File organization after the Strategic Planning section is completed.

```
wget --no-clobber \
  https://github.com/SnakeChunks/SnakeChunks/archive/
  4.1.4.tar.gz
tar xvzf 4.1.4.tar.gz
mv SnakeChunks-4.1.4 SnakeChunks
```

9. Download the reference genome of *E. coli* K-12 and its annotations.

```
make -f SnakeChunks/examples/GSE41195/tutorial_
  material.mk \
download_genome_data
```

10. Check the organization of the files in the genome directory (Fig. 3).

```
tree -L 2
```

IMPORTANT NOTE: The above steps are used to set up the environment and need to be executed only once, except for steps 3, 5, and 7, which are required for each working session for this project. If you log out of the terminal and want to start a new session later, you will need to reactivate the Conda environment (step 3), redefine the environment variable for the analysis directory (step 5), and set it as the current directory (step 7).

DATA PREPROCESSING AND READ MAPPING

BASIC PROTOCOL 1

Data preprocessing covers the first steps of the analysis, which are common to most NGS workflows. The goal is to make sure that the raw sequencing data are suitable for a proper bioinformatics analysis. This process includes quality control of the sequenced reads, removal of the sequencing adapters, and trimming of the read extremities when needed. These operations are described more thoroughly in the Guidelines for Understanding Results below. We illustrate these steps with a ChIP-seq dataset, but they can be applied similarly to RNA-seq data.

Once the reads are processed and filtered appropriately, a common operation to perform before ChIP-seq and RNA-seq analyses is to map the reads on a reference genome in order to identify their genomic location.

This protocol covers the following steps:

- Quality control of the reads using the program FastQC (Andrews, 2010);
- Removal of the adapters and trimming of the read extremities using the utility cutadapt (Martin, 2011);
- Read mapping using the algorithm bowtie2 (Langmead & Salzberg, 2012).

Figure 4 File organization of the ChIP-seq samples before the analyses are run.

1. Download the ChIP-seq dataset from the GEO series GSE41195 (Myers et al., 2013).

```
make -f SnakeChunks/examples/GSE41195/tutorial_
  material.mk\
download_chipseq_data
```

This creates a subdirectory called "ChIP-seq" in the analysis directory defined in the Strategic Planning section above (Fig. 4), with two fastq files corresponding to the FNR-chipped and control samples, respectively.

```
tree ChIP-seq
```

2. Create a local copy of the metadata folder.

```
make -f SnakeChunks/examples/GSE41195/tutorial_
  material.mk copy_metadata;
tree metadata
```

This creates a local copy of the metadata folder, which contains files describing the samples, the analysis design, and the workflow configuration.

3. Run the workflow for quality control.

```
snakemake -s SnakeChunks/scripts/snakefiles/
  workflows/quality_control.wf \
--configfile metadata/config_ChIP-seq.yml
--config trimming="" -p --use-conda
```

The command above runs a workflow using the "snakemake" command with the following specifications.

The wiring of the workflow is defined in the file `quality_control.wf`, specified with the option `-s`. Modifying this wiring requires some knowledge of the Snakemake language, which is outside the scope of this protocol (Snakemake tutorials can be found in the Snakemake documentation at http://snakemake.readthedocs.io/en/stable/tutorial/tutorial.html). `quality_control.wf` produces quality reports using the FastQC tool (Andrews, 2010), and running this is an essential step to assess the quality of the samples and plan the next steps of the analysis.

The workflow invokes a series of tools, each of which can be tuned with different parameters. All of the parameters of the workflow are specified in a YAML-formatted configuration file, specified with the option `--configfile`. The YAML format is human readable and can be easily edited with a standard text editor (see Support Protocol).

- *The option `--config` is used in order to specify that trimming will not be performed during this run. It overrules the configuration defined in the configuration file mentioned above, which is to perform trimming automatically, as will be done in step 5.*
- *The option `-p` tells Snakemake to print out all the Unix commands that will be executed. This listing is very convenient as a means to check that each*

command is called with the appropriate parameters and to keep a trace of the full process between raw data and final results.

- *When the option `--use-conda` is used, Snakemake creates a separate virtual environment for each rule executed in the workflow, and installs the required tools and their dependencies in a rule-specific subfolder. This ensures compatibility between the different tools invoked. The process can take some time at the first invocation of a given environment, but is faster for subsequent uses of the same environment.*

4. The presence of the two FastQC reports can be checked with the `ls` commands below.

```
ls -l $ANALYSIS_DIR/ChIP-seq/fastq/FNR1/FNR1_fastq.
  gz_qc/FNR1_fastqc.html;
ls -l $ANALYSIS_DIR/ChIP-seq/fastq/input1/input1_
  fastq.gz_qc/input1_fastqc.html
```

   These files can be opened with a Web browser. Insights about these reports can be found in the Guidelines for Understanding Results below.

5. Run the quality control workflow again using the software cutadapt, which performs both read trimming and adapter removal.

```
snakemake -s SnakeChunks/scripts/snakefiles/
  workflows/quality_control.wf \
--configfile metadata/config_ChIP-seq.yml -p
  --use-conda
```

   This time, the workflow will run cutadapt, as defined in the configuration file, before doing a new FastQC check. Note that SnakeChunks can be used to specify several tools for the same step, in order to compare the results. An overview of the options is proposed in Support Protocol.

6. The presence of FastQC reports can be checked with the `ls` commands below.

```
ls -l \
  $ANALYSIS_DIR/ChIP-seq/fastq/FNR1/FNR1_cutadapt_
    fastq.gz_qc/FNR1_cutadapt_fastqc.html;
ls -l \
  $ANALYSIS_DIR/ChIP-seq/fastq/input1/input1_
    cutadapt_fastq.gz_qc/input1_cutadapt_fastqc.html
```

   Open the new FastQC reports with a Web browser. The reports show the improvement in the quality of the reads, as well as the absence of over-represented sequences corresponding to adapters. This is further discussed in the Guidelines for Understanding Results.

7. Run the read-mapping workflow.

```
snakemake -s SnakeChunks/scripts/snakefiles/
  workflows/mapping.wf \
--configfile metadata/config_ChIP-seq.yml -p --use-
  conda -j 2
```

   This workflow essentially performs two operations: read mapping and genome coverage.

   We added the option `-j 2`, which permits Snakemake to parallelize the processing with a maximum of two simultaneous jobs. Because the mapping step can be time consuming, we recommend running it in parallel for the different samples. This option should be adapted to the number of cores of your system. For example, if you analyze a large number of files on a cluster, you could increase the number of simultaneous jobs to 40 or even more (this has to be negotiated with your system administrator).

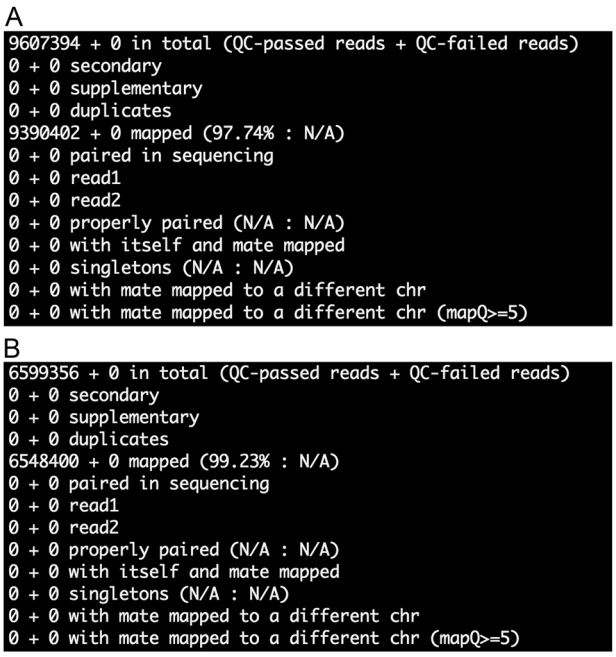


Figure 5 Read mapping statistics. Statistics were computed using the flagstats software from SAMtools for the FNR ChIP-seq sample (**A**) and genomic input (**B**), respectively.

More information about the mapping results can be found in the Guidelines for Understanding Results.

8. Check the contents of the files containing the statistics of the mapping from the shell (Fig. 5).

```
cat \
$ANALYSIS_DIR/ChIP-seq/results/samples/FNR1/FNR1_
  cutadapt_bowtie2_bam_stats.txt;
cat \
$ANALYSIS_DIR/ChIP-seq/results/samples/input1/
  input1_cutadapt_bowtie2_bam_stats.txt
```

These files, generated by the SAMtools program flagstat, display basics statistics for the mapping. As can be seen in Figure 5A and B, here both samples have a very high mapping rate, which confirms that the sequencing data are of good quality and that we are going to dispose of a large quantity of data to perform the ChIP-seq analysis.

BASIC PROTOCOL 2

ChIP-seq

ChIP-seq (Johnson, Mortazavi, Myers, & Wold, 2007; Robertson et al., 2007) is a technology that allows the characterization of DNA binding at a genome scale. The experiment includes the following steps: cross-linking DNA and the bound proteins with a fixative agent, breaking DNA into random fragments by ultrasonication, immunoprecipitating a transcription factor of interest together with its cross-linked DNA, unlinking these DNA fragments, amplifying them by PCR, and sequencing them using massively parallel sequencing technologies. The raw sequences ("reads") are then mapped onto a reference genome, and putative binding regions—regions that contain a large number of reads, usually extending over a few hundred base pairs—are denoted as "peaks." These peaks can then be used to search for precise transcription-factor (TF) binding sites, which can then be associated with nearby genes to infer the potential TF target genes.

Table 1 Descriptions of the ChIP-seq Samples

ID	Condition	GSM identifier	SRR identifier
FNR1	FNR	GSM1010220	SRR576934
input1	Input	GSM1010224	SRR576938

Column headers indicate their contents. the columns ID and Condition are mandatory for the proper use of the workflow. Additional columns can be added at will to document samples.

Table 2 Experimental Design of the ChIP-seq Dataset

Control	Treatment
input1	FNR1

A critical step of a ChIP-seq data analysis is peak calling, which is the detection of these genomic regions with a higher density of mapped reads than would be expected by chance. The choice of a peak-calling algorithm and the tuning of its parameters can drastically affect the number of returned peaks and their sizes. To identify reliable peaks and avoid false positives, it is important to use control samples (see Commentary for more details). Peak callers also have parameters that can be used to tune the rate of false positives by imposing more or less stringent thresholds on peak scores, in order to optimize the tradeoff between sensitivity (the proportion of actual binding regions detected) and specificity (the ability to reject non-binding regions).

Table 1 describes each sample used in the analysis: a test sample resulting from the immunoprecipitation of the FNR transcription factor, and a genomic input. Table 2 specifies the design of the analysis, by indicating the respective status of the samples (control versus treatment).

Although many publications rely on the Macs2 peak caller (Feng, Liu, & Zhang, 2011), generally used with its default parameters, there are actually a variety of tools that can be used and customized in different ways (Pepke, Wold, & Mortazavi, 2009). SnakeChunks currently supports seven of these in a completely interchangeable way (Fig. 2A). We will demonstrate two, Homer (Heinz et al., 2010) and Macs2, which are among the most widely used, maintained, and up-to-date programs for this purpose and which are also supported by Conda.

The main operations performed by the workflow described are the following:

- Peak calling using Homer and Macs2 (Feng et al., 2011; Heinz et al., 2010);
- Motif discovery by remote invocation of the tool peak-motifs (Thomas-Chollier et al., 2012) from the RSAT software suite (Nguyen et al., 2018) via its Web services interface; RSAT peak-motifs also compares discovered motifs with the TF-binding motifs annotated in RegulonDB;
- Comparison between ChIP-seq peaks and known TF binding sites listed in the RegulonDB database (Gama-Castro et al., 2016);
- Assignment of genes to peaks with the tool “annotate peaks” from the Homer suite;
- Gene comparison: comparison between genes associated with peaks and TF target genes (as annotated in RegulonDB).

1. Run the ChIP-seq workflow.

```
snakemake \
-s SnakeChunks/scripts/snakefiles/workflows/
  ChIP-seq_RegulonDB.wf \
```

```
--configfile metadata/config_ChIP-seq.yml -p --use-
  conda -j 2
```

2. The output files can be found here.
   a. Peaks: Because these files are quite large, we use the Unix command `less` to display them page by page (press enter to move one page forward). After inspecting a few pages, type "`q`" to quit the less program.

```
less \
$ANALYSIS_DIR/ChIP-seq/results/peaks/FNR1_vs_
  input1/homer/FNR1_vs_input1_cutadapt_bowtie2_
  homer.bed;
less \
$ANALYSIS_DIR/ChIP-seq/results/peaks/FNR1_vs_
  input1/macs2/FNR1_vs_input1_cutadapt_bowtie2_
  macs2.bed
```

   b. Motifs discovered with RSAT in the peaks: Check that the html files produced by peak-motifs are at the expected place.

```
ls -l \
$ANALYSIS_DIR/ChIP-seq/results/peaks/FNR1_vs_input1/
  homer/peak-motifs/FNR1_vs_input1_cutadapt_bowtie2_
  homer_peak-motifs/peak-motifs_synthesis.html;
ls -l \
$ANALYSIS_DIR/ChIP-seq/results/peaks/FNR1_vs_input1/
  macs2/peak-motifs/FNR1_vs_input1_cutadapt_bowtie2_
  macs2_peak-motifs/peak-motifs_synthesis.html
```

Open the peak-motifs reports with a Web browser. The results of this workflow are further described in the Guidelines for Understanding Results below.

BASIC PROTOCOL 3

RNA-seq

RNA-seq technology, or whole-transcriptome shotgun sequencing, reveals the presence or absence of RNAs from a given sample, at a given moment in time, and also quantifies them if needed. It consists of extracting the total RNA from a cell and filtering out genomic DNA using a deoxyribonuclease (DNase). The RNA is then reverse transcribed to cDNA, which can either be mapped onto a genome of reference or assembled *de novo*. Subsequent analysis options include quantification of gene expression, identification of alternative transcripts, and discovery of single-nucleotide variation.

In this protocol, we will use as a case study an RNA-seq experiment published by Myers et al. (2013), in which the transcriptome of *E. coli* K-12 was measured in two samples from the wild type (WT) and from a mutant strain whose FNR transcription factor activity is inhibited (Lazazzera, Bates, & Kiley, 1993). To perform reliable RNA-seq analyses, it is crucial to dispose of biological replicates (see Commentary). This dataset includes two replicates per genotype (Table 3). Our goal will be to identify genes that are differentially expressed between the FNR mutant (defined as the test condition in Table 4) and the WT (reference condition).

This workflow accomplishes the following steps:

- Quality control and trimming of the reads (for further detail, see Basic Protocol 1);
- Mapping onto a genome of reference using the algorithm BWA (Li & Durbin, 2009) (for further detail, see Basic Protocol 1);

Table 3 Descriptions of the RNA-seq Samples

ID	Condition	GSM identifier	SRR identifier
WT1	WT	GSM1010244	SRR5344681
WT2	WT	GSM1010245	SRR5344682
dFNR1	FNR	GSM1010246	SRR5344683
dFNR2	FNR	GSM1010247	SRR5344684

Column headers indicate their contents. The columns ID and Condition are mandatory for the proper use of the workflow. Additional columns can be added at will to document samples

Table 4 Experimental Design of the RNA-seq Analysis

Test	Reference
FNR	WT

The design file can contain one or several rows, each describing a pair of conditions to be compared. The test and reference conditions must correspond to the values in the Condition column of the sample description table.

- Quantification of transcripts per gene with featureCounts from the Subread package (Liao et al., 2014);
- Detection of differentially expressed genes with DESeq2 (Love, Huber, & Anders, 2014) and edgeR (Robinson, McCarthy, & Smyth, 2010);
- Automatic generation of a report summarizing the results.

1. Copy the example metadata from the SnakeChunks library (can be skipped if already done in Basic Protocol 1, step 2), and check the content of the metadata folder.

```
make -f SnakeChunks/examples/GSE41195/tutorial_
  material.mk copy_metadata; tree metadata
```

2. Download RNA-seq data.

```
make -f SnakeChunks/examples/GSE41195/tutorial_
  material.mk download_rnaseq_data
```

This creates a subdirectory "RNA-seq" in the analysis directory defined in Strategic Planning (Fig. 6), and downloads the raw data. Beware: during our tests, the download takes approximately 8 min per sample. Since the analysis requires eight files, this download can take up to a few hours depending on your connection speed. After the command has been completed, check the organization of the downloaded files.

```
tree -C RNA-seq
```

You should now see four directories (one per sample), each containing two files with the extension .fastq.gz (there is one file per sequencing end).

3. Run the RNA-seq analysis workflow.

```
snakemake -s SnakeChunks/scripts/snakefiles/
  workflows/RNA-seq_complete.wf \
--configfile metadata/config_RNA-seq.yml -p
  --use-conda -j 4
```

Here we use the option `-j 4` in order to parallelize the treatment of the four samples, which is time consuming.

4. Check the organization of the result files in the RNA-seq folder, with a folder depth limit of 3.

```
tree -C -L 3 RNA-seq
```

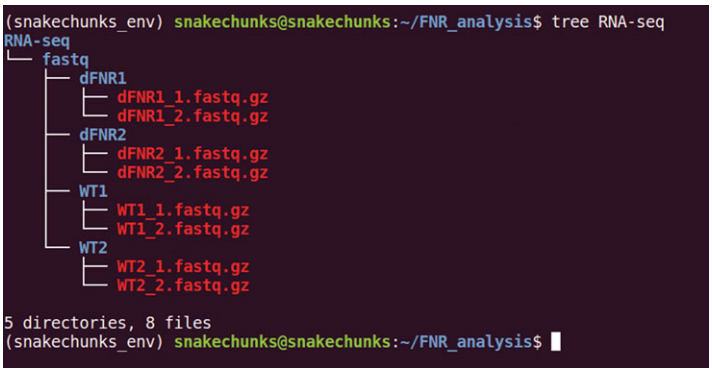


Figure 6 File organization of the RNA-seq samples before the analyses are run.

5. The results of the differential expression analysis performed by this workflow are summarized in an automatically generated HTML report, which can be opened using a web navigator.

```
RNA-seq/results/diffexpr/cutadapt_bwa_featureCounts_
  rna-seq_deg_report.html
```

The elements of this report are further described in the Guidelines for Understanding Results below.

6. Optionally, it is now possible to check the content of the main result files, which can be found here.

```
ls -l RNA-seq/results/diffexpr
```

This folder contains a table with the counts of reads per gene:

```
less RNA-seq/results/diffexpr/cutadapt_bwa_
  featureCounts_all.tsv
```

and a subfolder with the differential analysis results produced by edgeR, DESeq2, and the two together.

```
ls -l RNA-seq/results/diffexpr/FNR_vs_WT
```

It also contains two tables with the differential analysis statistics returned by DESeq2 and edgeR, respectively.

```
less \
RNA-seq/results/diffexpr/FNR_vs_WT/cutadapt_bwa_
  featureCounts_FNR_vs_WT_DESeq2.tsv;
less \
RNA-seq/results/diffexpr/FNR_vs_WT/cutadapt_bwa_
  featureCounts_FNR_vs_WT_edgeR_TMM.tsv
```

The subset of differentially expressed genes (those declared positive because they pass the significance threshold) are exported in an additional file.

```
less \
RNA-seq/results/diffexpr/FNR_vs_WT/cutadapt_bwa_
  featureCounts_FNR_vs_WT_DEG_table.tsv
```

In the tutorial, we retain the union of genes called positive by either DESeq2 or edgeR, but alternatively, the combination rule can be tuned in the YAML configuration file.

7. We can count the rows of this file to get an idea of the number of differentially expressed genes (after subtracting one for the header line).

```
wc -l
RNA-seq/results/diffexpr/FNR_vs_WT/cutadapt_bwa_
  featureCounts_FNR_vs_WT_DEG_table.tsv\
| awk '{print $1 -1}'
```

INTEGRATION

BASIC PROTOCOL 4

We have seen in Basic Protocol 2 that a ChIP-seq experiment followed by peak calling can be used to identify genomic binding locations for a given transcription factor. In Basic Protocol 3, we analyzed results of an RNA-seq experiment to identify genes differentially expressed between two conditions (wild-type versus FNR mutant).

Here, we show how to combine the results of those two types of experiments in order to unravel the links between genome binding data (ChIP-seq) and differential expression data (RNA-seq). This allows to detect not only direct target genes of a factor, i.e., genes whose transcription level is affected in the mutant, and whose upstream region contains a binding peak, but also indirect regulation (absence of a binding peak but presence of an observed effect on the expression of a gene) or binding of the FNR transcription factor without detected effect on the level of transcription of the associated genes. We also compare the NGS results with the list of FNR target genes annotated in the RegulonDB database (Gama-Castro et al., 2016).

1. Run integration workflow.

```
snakemake -p \
-s SnakeChunks/scripts/snakefiles/workflows/
  integration_ChIP_RNA.wf \
--configfile metadata/config_integration.yml
  --use-conda
```

2. Check the first lines of the table summarizing the results for each gene.

```
less $ANALYSIS_DIR/integration/ChIP-RNA-regulons_
  homer_gene_table.tsv
```

   For a better readability, we recommend opening this table with spreadsheet software (e.g., Office Calc or Excel). The table contains annotations for all genes known in E. coli K-12, as well as an indication of whether they are associated with FNR binding (ChIP-seq column), whether their transcription is affected by FNR (RNA-seq column), and whether they have been previously demonstrated to be regulated by FNR (FNR_regulon column).

3. Launch the IGV browser (Robinson et al., 2011; Thorvaldsdóttir, Robinson, & Mesirov, 2013):

   On Linux operating systems: `igv`

   In Mac OS X: open the IGV in the Applications folder.

4. Click on menu File, select Open session..., and select the session file `metadata/igv_session.xml` in the FNR analysis directory.

   This will load an IGV session with our selection of relevant tracks for the interpretation of ChIP-seq and RNA-seq results, which are discussed further in the Guidelines for Understanding Results below.

RUNNING THE WORKFLOW WITH THE USER-FRIENDLY INTERFACE SEQUANIX

ALTERNATE PROTOCOL

Sequanix (Desvillechabrol et al., 2018) is a graphical user interface (GUI) based on PyQt, developed to facilitate the execution of NGS Snakemake pipelines. It was originally designed to run workflows included in the Sequana project (*http://sequana.readthedocs.io*),

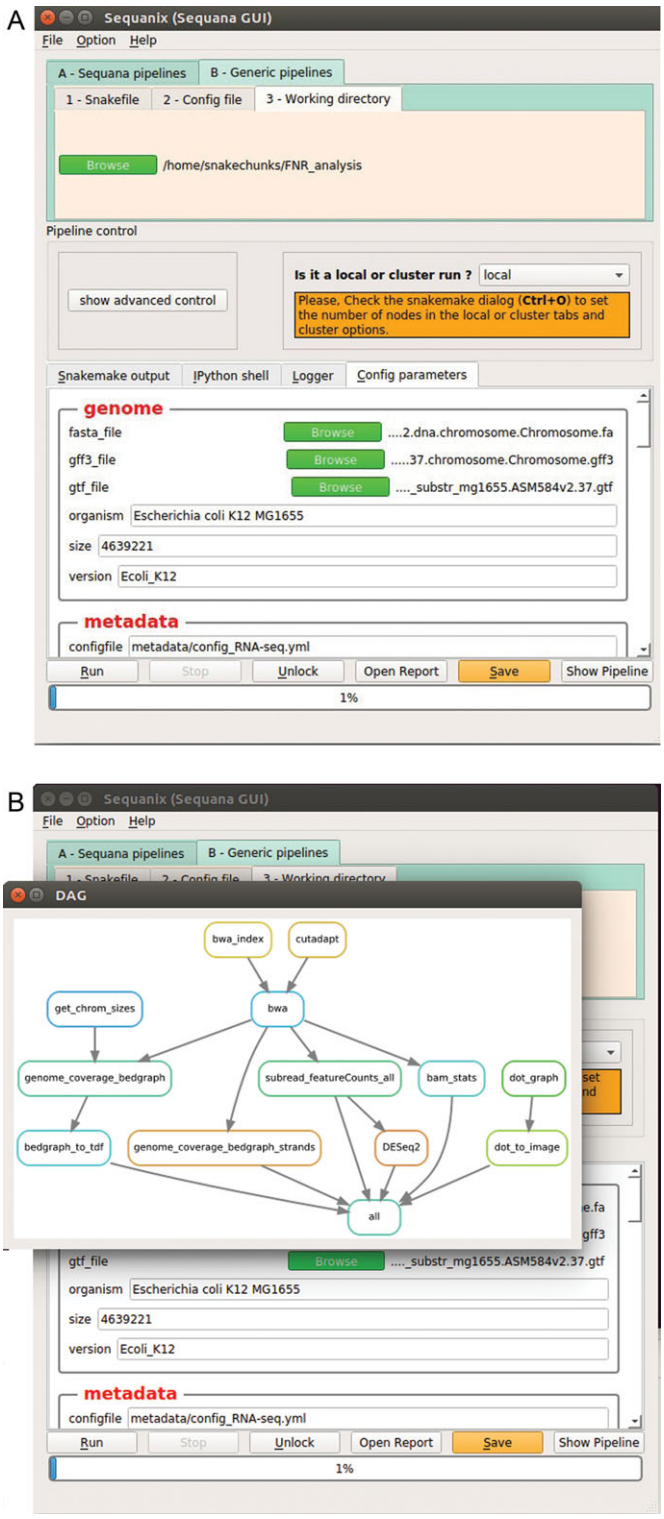


Figure 7 Sequanix graphical user interface. (**A**) Configuration of the workflow parameters. (**B**) Display of workflow wiring. The diagram shows the directed acyclic graph (DAG) of rules automatically generated by Snakemake.

but can also handle any Snakemake pipeline. Thanks to the graphical interface, the parameters can be customized easily and the workflows can be run without using any command line.

Here we demonstrate the execution of the RNA-seq workflow (see Basic Protocol 3) using this interface.

Necessary Resources

Conda: If not already done, create and activate a Conda environment (Strategic Planning, steps 1 to 10)
Sequana: Install Sequana: type `conda install -c bioconda sequana=0.7.1`
RNA-seq dataset: If not already done, download the RNA-seq dataset (Basic Protocol 3, step 1 and 2) to install the metadata and download RNA-seq raw reads

1. Launch Sequanix.

   ```
   sequanix
   ```

2. At the top of the Sequanix window, select the tab "Generic pipelines."
3. Under the Snakefile tab, fetch the workflow file `RNA-seq_complete.wf` in the directory `SnakeChunks/scripts/snakefiles/workflows`.
4. Under the Config file tab, fetch the configuration file `config_RNA-seq.yml` in the directory `metadata`.
5. Under the Working directory tab, select the directory you defined above as `$ANALYSIS_DIR` (Strategic Planning, step 5) (Fig. 7A).
6. In the menu of the application, select Options > Snakemake options ... > General, and type "`--use-conda`" in the bottom box "other options," then press OK.
7. In the Sequanix main window, press Save.
8. Press Show pipeline to check that everything looks reasonable (Fig. 7B).
9. Press Run.

   If you have followed Basic Protocol 3, the Run button should not start any new analysis, because Snakemake will detect that the result files are already present. If not, Sequanix will run the workflow just as in the terminal.

CUSTOMIZATION OF PARAMETERS

SUPPORT PROTOCOL

Each workflow available in SnakeChunks requires three basic files in order to specify the input data files and all the parameters of an analysis. These files have been placed in a directory named "metadata." We explain here how to adapt the ChIP-seq metadata files, but the same principle applies to the RNA-seq and integration workflows. The ChIP-seq workflow runs using three metadata files:

- Sample file: `samples_ChIP-seq.tab`;
- Design file: `design_ChIP-seq.tab`;
- Workflow and tool parameters: `config_ChIP-seq.yml` (Fig. 8A).

The sample file (Table 1) describes each sample to be analyzed (one row per sample), with two mandatory columns (ID and Condition) and optional columns for complementary information such as GSM identifiers. Here, we have two samples: one ChIP-ped with FNR, and a control sample labeled "input" following the ChIP-seq convention.

A

```
############################################################
## WORKFLOW DESIGN
##
  trimming: "cutadapt"
  mapping: "bowtie2"
  peakcalling: "homer macs2"

############################################################
## OPTIONAL PARAMETERS
##
## Parameters used by rules & programs.
## If nothing is mentionned below, all programs will use their default parameters.
  cutadapt:
    qual_threshold: "20"                                    # Optional (def. 20)
    length_threshold: "20"                                  # Optional (def. 20)

  macs2:
    qval: "0.05"                                            # Optional (def. 0.05)
    keep_dup: "all"                                         # Optional (def. 1)
    mfold_min: "2"                                          # Optional (def. 5)
    mfold_max: "50"                                         # Optional (def. 50)
    other_options: "--nomodel"                              # Optional can include --call-summits, --broad...

  homer:
    style: "factor"                                         # Optional (def. factor), can be factor, histone,
    F: "2"                                                  # Optional (def. 4)
    L: "2"                                                  # Optional (def. 4)
    P: "0.01"                                               # Optional (def. 0.0001)
    fdr: "0.01"                                             # Optional (def. 0.001)
```

B

```
############################################################
## WORKFLOW DESIGN
##
  trimming: "sickle"                                        # Available options > sickle, cutadapt
  mapping: "subread-align"                                  # Available options > bwa, bowtie2, subread-align...
  peakcalling: "homer macs2 spp"                            # Available options > homer, macs2, spp

############################################################
## OPTIONAL PARAMETERS
##
## Parameters used by rules & programs.
## If nothing is mentionned below, all programs will use their default parameters.
  sickle:
    qual_threshold: "25"                                    # Optional (def. 20)
    length_threshold: "25"                                  # Optional (def. 20)

  macs2:
    qval: "0.001"                                           # Optional (def. 0.05)
    keep_dup: "all"                                         # Optional (def. 1)
    mfold_min: "2"                                          # Optional (def. 5)
    mfold_max: "50"                                         # Optional (def. 50)
    other_options: "--nomodel"                              # Optional can include --call-summits, --broad...

  homer:
    style: "factor"                                         # Optional (def. factor), can be factor, histone...
    F: "4"                                                  # Optional (def. 4)
    L: "4"                                                  # Optional (def. 4)
    P: "0.0001"                                               # Optional (def. 0.0001)
    fdr: "0.001"                                            # Optional (def. 0.001)

  spp:
    fdr: "0.01"                                             # Optional (def. 0.05)
```

Figure 8 YAML-formatted configuration file for the ChIP-seq workflow. The YAML format enables the user to specify all the parameters of a workflow in a structured way while being human readable and easily editable. (**A**) Default configuration. (**B**) Customized configuration.

The design file (Table 2) defines the samples to be compared in order to perform peak calling. Here, we are going to perform peak calling of the ChIP sample, using the input sample as a background control. For RNA-seq, the design defines the conditions to be compared.

The configuration file (Fig. 8A) is specific to the workflow to be run. It contains three main parts: (1) general information about the reference genome, metadata file, and file organization; (2) general design of the workflow, such as the steps to be performed (trimming, mapping, peak calling, annotation) and the tools to be used at each step; and (3) an optional section enabling to customize the parameters used for each tool (if not specified, their default parameters are used).

Below, we explain how to edit the configuration file in order to generate alternative results, using different tools and parameters.

IMPORTANT NOTE: Be aware that performing alternative trimming and/or mapping can require additional disk space, since FASTQ files (raw reads, trimmed reads) and BAM files (aligned reads) are very space consuming. In the following protocol, that requires about 2 Gb of disk space, but this can go as high as tens of gigabases in the case of larger raw files, such as the RNA-seq files analyzed in Basic Protocol 3.

1. Create a copy of the ChIP-seq config file.

```
cd $ANALYSIS_DIR; \
cp metadata/config_ChIP-seq.yml metadata/config_
  ChIP-seq_custom.yml
```

2. With a text editor, make the following changes to your custom configuration file (`metadata/config_ChIP-seq_custom.yml`).
   a. Change the trimming software from cutadapt to sickle.
   b. Change the mapping software from bowtie2 to subread-align.
   c. Add the SPP peak caller to Homer and Macs2.
   d. Customize the SPP, Homer, and Macs2 parameters in the third section according to the values shown in Figure 8B.

   Alternatively, you can avoid manual editing of parameters by copying the ready-to-use customized configuration file provided in the distribution. To do this, skip step 2 and instead run the following command:

```
cp metadata/config_ChIP-seq_advanced.yml metadata/
  config_ChIP-seq_custom.yml
```

3. Run the commands below, which correspond to steps 5 and 7 of Basic Protocol 1, and step 1 of Basic Protocol 2, 1.5, 1.7, and 2.1 adapted to use the custom configuration file.

```
snakemake \
-s SnakeChunks/scripts/snakefiles/workflows/quality_
  control.wf \
--configfile metadata/config_ChIP-seq_custom.yml -p
  --use-conda -j 2;
snakemake \
-s SnakeChunks/scripts/snakefiles/workflows/mapping.
  wf \
--configfile metadata/config_ChIP-seq_custom.yml -p
  --use-conda -j 2;
snakemake \
-s SnakeChunks/scripts/snakefiles/workflows/
  ChIP-seq_RegulonDB.wf \
--configfile metadata/config_ChIP-seq_custom.yml -p
  --use-conda -j 2
```

4. Visualize the differences in the IGV: load a session as in Basic Protocol 4, steps 3 and 4.

5. Click on the menu File, select "Load from File . . . ," and select the following peak files:

```
$ANALYSIS_DIR/ChIP-seq/results_advanced/peaks/FNR1_
  vs_input1/spp/FNR1_vs_input1_sickle_subread-align_
  spp.bed
$ANALYSIS_DIR/ChIP-seq/results_advanced/peaks/FNR1_
  vs_input1/homer/FNR1_vs_input1_sickle_subread-
  align_homer.bed
$ANALYSIS_DIR/ChIP-seq/results_advanced/peaks/FNR1_
  vs_input1/macs2/FNR1_vs_input1_sickle_subread-
  align_macs2.bed
```

   By running the command `wc -l` on these files, you can note the influence of the choice of peak caller, as well as its parameters.

GUIDELINES FOR UNDERSTANDING RESULTS

Data Preprocessing and Read Mapping (Basic Protocol 1)

Quality control

For each sample, FastQC produces a box plot representing per-base sequence quality. A common phenomenon in high-throughput sequencing is a decrease in sequence quality at the 3′ end of the reads. This can indeed be observed for the input sample in our case study (Fig. 9). Low read quality can reduce the percentage of reads mapped on the reference genome. To avoid this, we recommend performing sequence trimming to remove low-quality read extremities.

Another interesting category of information in FastQC reports is the sequence-duplication levels. The graph outlines read sequences found in an excessive number of copies, which may diagnose an effect of PCR amplification due to poor complexity of the DNA library. Note that duplication is often interpreted in contexts in which the sequence library is much smaller than the genome size (typically ∼50 M reads for a ∼3-Gb mammalian genome), so that reads resulting from a random sampling are not expected to fall on exactly the same genomic position. When studying bacterial regulation, however, library size can exceed genome size (typically 4 Mb) so that multiple matches are expected along

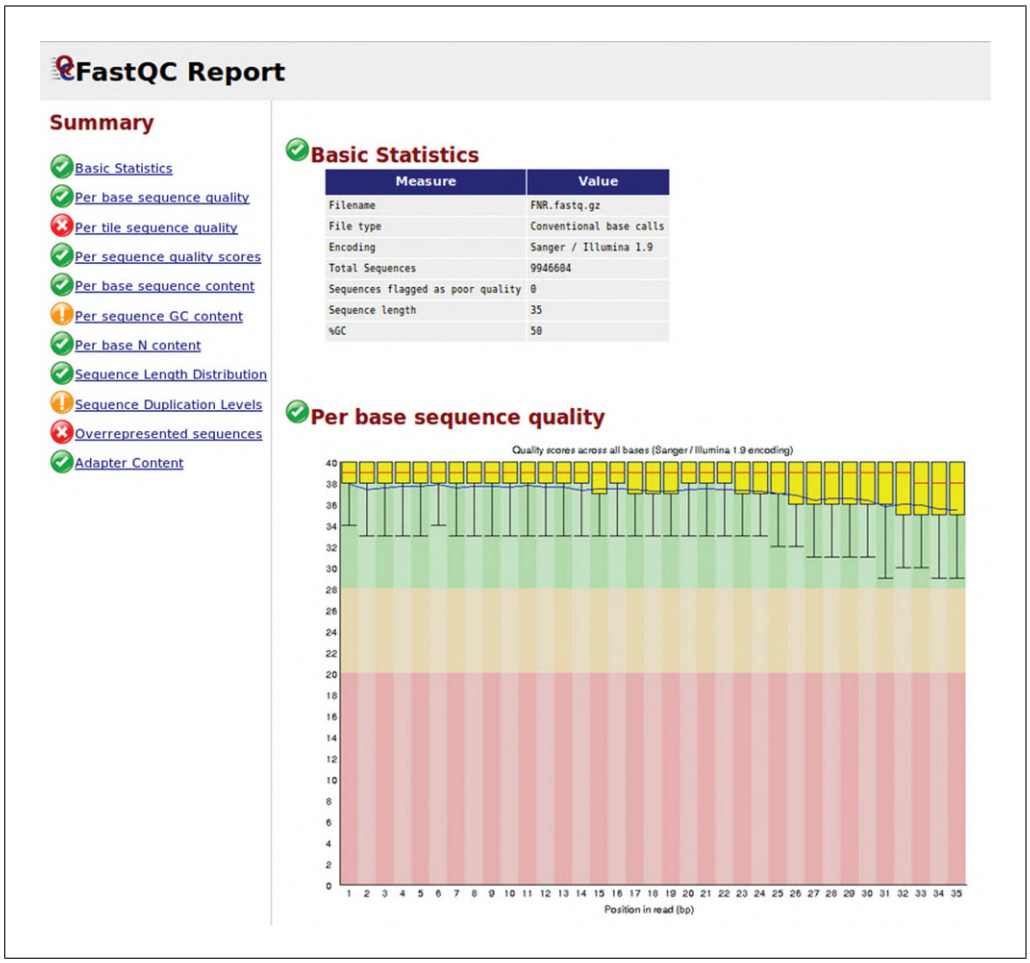


Figure 9 Quality report of the FNR1 ChIP-seq raw reads before trimming. The abscissa (columns) corresponds to nucleotide positions along the mapped reads; the ordinate indicates read quality scores. For each position, statistics are summarized for all the reads of a library: median (red line), interquartile range (yellow box), and quality range (vertical line). Background colors indicate an arbitrary subdivision of quality scores, from red (insufficient) to green (good).

the whole genome. Another section of the FastQC report provides statistics about over-represented sequences. Before removal of the adapters by cutadapt (Basic Protocol 1, step 5), Illumina adapters represent respectively 0.5% and 2.6% of the total number of reads of the FNR1 and input1 samples. After cutadapt is run, these sequences are gone (Basic Protocol 1, step 6). Detailed information on the interpretation of read quality is provided on the FastQC Web site (*http://www.bioinformatics.babraham.ac.uk/projects/fastqc/*).

Read mapping

Using the bowtie2 algorithm, the trimmed reads in FASTQ format are aligned onto a genome of reference, downloaded as described in Strategic Planning. In our case, the reference is *E. coli* K-12. The result of the alignment comes in a BAM format that retains all the information from the fastq files about read sequences and quality, but adds the putative positions of the reads in the reference genome.

Genome coverage

Genome coverage files makes it possible to visualize the mapped reads in a condensed way, by showing the number of reads overlapping each position on each strand of the reference genome (Fig. 10A, pink, gray, and jade tracks in the middle panel) or their sum on both strands (purple track). Coverage profiles can be stored in different file formats (e.g., tdf, bedgraph, bigwig) depending on the size of the dataset and the way to display it. In this protocol, we use the TDF format, which is the recommended format for optimal IGV visualization.

ChIP-seq (Basic Protocol 2)

Peak calling

The peaks detected by Homer and Macs2 can be visualized in IGV as BED files. This file format contains essentially the coordinates of the regions with a high density of mapped reads, which are called "peaks." Although in bacteria it is expected that ChIP-seq peaks will fall into intergenic regions upstream of the regulated genes, it has been shown that a surprisingly high amount of binding may occur into coding or downstream regions (Galagan, Lyubetskaya, & Gomes, 2012). This observation should be interpreted by taking into account the fact that bacteria have a very small proportion of intergenic regions (10% to 15% of the genome).

Figure 10A shows a very clear peak around position 2,344,000, detected by both peak callers, in the noncoding region upstream of the gene *nrdA*. On comparing the ChIP-seq read coverage on the forward and reverse strands (pink tracks in the middle panel), we see a shift between forward and reverse peaks. This typical pattern is consistent with the expectation for ChIP-seq experiments, because immunoprecipitated fragments are sequenced at their extremities, so that the reads are expected to be found either on the forward strand to the left of the binding site, or on the reverse strand to its right.

Different peak-calling tools can produce very different results for the same dataset. In the same region (Fig. 10A), Macs2 detects another peak around position 2,347,000, associated with the gene *nrdB*, which belongs to the same operon as *nrdA*. It is not identified as a peak by Homer, and it is not associated with any known FNR TF binding sites from RegulonDB. However, RegulonDB indicates that *nrdB* is regulated by H-NS and Fis, nucleoid-associated proteins (NAPs) that are known to mask FNR binding sites under anaerobic conditions (Myers et al., 2013). Although barely detected by peak callers, this site is thus supported by some experimental evidence.

Figure 10 Snapshots of ChIP-seq results for selected genomic regions. The figures were generated with the Integrative Genomics Viewer (IGV). (**A**) High-confidence peak in the promoter region of the nrdAB operon. Note the characteristic shift between reads mapped on the plus and minus strands. (**B**) Example of a peak that is likely to be a false positive. For both IGV maps (A and B), the top panels show the coordinates of the displayed genomic region. The middle panels show read density profiles in the input (gray) and ChIP-seq samples (purple for strand-insensitive, pink for strand-sensitive profiles), and RNA-seq data (WT in gray, FNR mutants in turquoise). The lower panels show annotation tracks for genes (yellow), annotated FNR binding sites (none found in the displayed regions), and binding peaks.

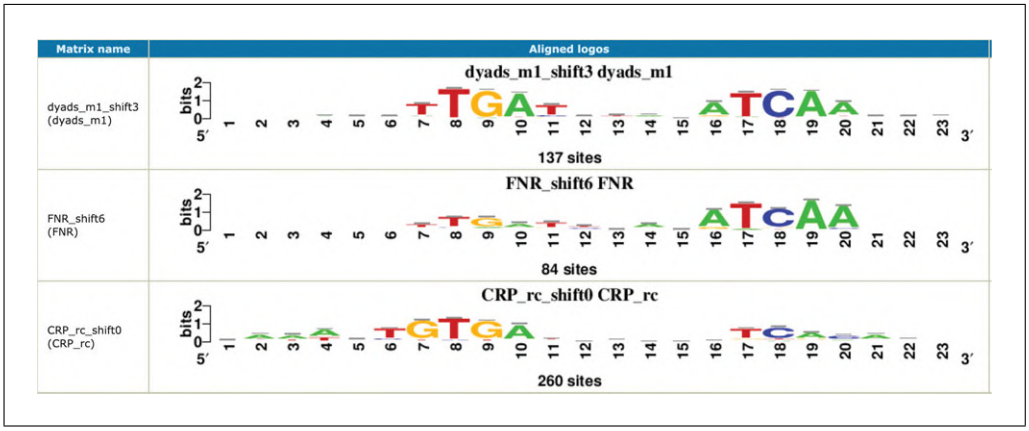


Figure 11 Most significant motif discovered by RSAT peak motifs in the FNR peaks, aligned with matching motifs in RegulonDB.

In contrast, Figure 10B shows a typical example of a peak that is likely to be a false positive. Note that its read enrichment is restricted to the reverse strand and falls within the coding region of a gene. Strand-specific display of read coverage thus makes it possible to assess the reliability of peaks by inspecting their distribution around the putative binding sites.

The number of peaks and their width can vary considerably, hence the need to adapt the tools to a given study and assess the relevance of the downstream results. Under our working conditions, Homer returns 161 peaks of equal width (exactly 177 bp each), whereas Macs2 returns 411 peaks ranging from 200 to 5893 bp (with an average of 475 bp), an obviously excessive size for TF binding sites. The broadest peaks reported by Macs2 correspond to wide regions covering several genes, which are entirely covered by reads in the ChIP-seq sample, and indeed enriched with respect to the genomic input, but which likely do not correspond to TF binding sites. For Macs2, the number of peaks can be strongly modified by tuning the *q*-value threshold and the minimal fold change. For example, the number of peaks drops from 547 with a *q*-value threshold of 0.05 and a minimal fold-change of 2, to 159 with *q*-value threshold of 0.001 and a minimal fold change of 5. The most permissive conditions give fewer relevant peaks, denoted by a drop in the significance of the FNR motif. In summary, the choice of a peak-calling algorithm and the fine-tuning of its parameters crucially affect ChIP-seq results, and should be evaluated case by case.

Motif discovery in peak sequences

The top panel of Figure 11 shows the most significant motif returned by RSAT peak-motifs (Thomas-Chollier et al., 2012) in the sequences of Homer peaks. This motif was discovered by the tool dyad-analysis (van Helden, Ríos, & Collado-Vides, 2000), which detects over-represented pairs of spaced oligonucleotides. This motif discovery approach is particularly relevant for bacteria, where most transcription factors form homodimers that bind spaced motifs. The comparison of this discovered motif with all the TF binding motifs annotated in RegulonDB returns two matches, corresponding to FNR and CRP, respectively. The alignment highlights the strong similarity between the motifs recognized by FNR and CRP (they differ only by one nucleotide at position 7 of the motif alignment), which is consistent with the fact that these two factors are known to co-regulate a number of genes (Gama-Castro et al., 2016; Myers et al., 2013).

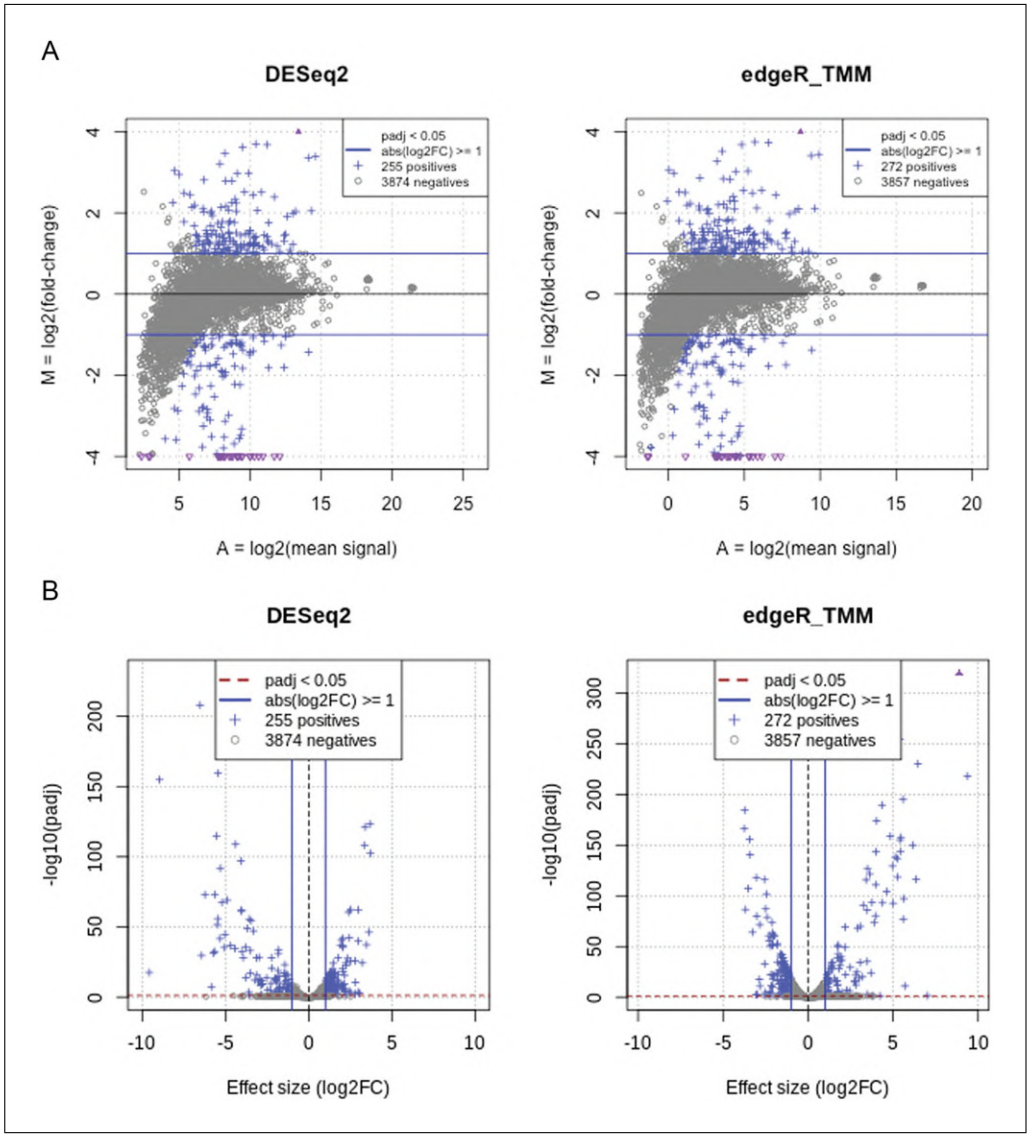


Figure 12 Global views of the results for the detection of differentially expressed genes between FNR mutant versus wild-type. These plots are generated as part of the differential analysis step, using an R script. Left and right panels respectively show the results of DESeq2 and edgeR. (A) MA plots. The abscissa indicates the mean level of expression (average of the log-transformed counts), and the ordinate shows the log fold change between FNR mutant and wild-type strain, which indicates the level of over- (positive values) or underexpression (negative values). Differentially expressed genes (DEGs), i.e., those passing both the effect size and significance thresholds, are highlighted in blue. Triangles indicate genes whose $\log_2$ fold change exceed the plot limits. (B) Volcano plots. The abscissa represents the log fold change, which indicates the size of the effect and its sign (–, downregulation; +, upregulation). The ordinate shows the significance of the differential expression (negative log of the adjusted *P* value).

RNA-seq (Basic Protocol 3)

Differentially expressed genes

The results of the RNA-seq analysis are summarized in an HTML report (`RNA-seq/results/diffexpr/cutadapt_bwa_featureCounts_rna-seq_deg_report.html`), which can be visualized using a web browser. It features information and statistics about the RNA-seq samples, read counts, and differentially expressed genes, detected by using two different tools: DESeq2 (Love et al., 2014) and edgeR (Robinson et al., 2010). Figure 12 shows MA plots and volcano plots that are automatically produced by the workflow to provide a synthetic representation of the

global results of the RNA-seq differential analysis. The MA plots (Fig. 12A) indicate the relationship between the mean level of expression of each gene (abscissa) and its differential expression, measured as the log fold difference between FNR mutant and wild type (ordinate). The genes declared differentially expressed between the two conditions (WT versus FNR) are highlighted as blue crosses. Genes overexpressed and underexpressed in the FNR mutants appear above or below the *x* axis, respectively. The volcano plots (Fig. 12B) provide a combined view of the expression changes (log fold change, on the abscissa) and the statistical significance of these changes (on the ordinate). The significance is computed as the negative logarithm of the adjusted *P* values reported by DESeq2 (left) and by edgeR (right), respectively. High values are indicative of significant differences of expression between FNR mutant and WT strains. To select differentially expressed genes, SnakeChunks combines user-modifiable thresholds on the adjusted *P* value (default: $\alpha = 0.05$) and on the fold change (default: at least twofold over- or underexpression).

In total, these thresholds lead to the retention of 278 differentially expressed genes that were declared positive by either DESeq2 (255 genes) or edgeR (272 genes). This number is consistent with the fact that FNR acts as global regulator in *E. coli*. Note that we chose to keep the union of both lists in order to favor sensitivity, but this can be parameterized in the configuration file by specifying that the detection of differentially expressed genes relies on edgeR, DESeq2, their intersection, or their union.

Integration (Basic Protocol 4)

The Venn diagram generated by the workflow (Fig. 13, file `integration/ChIP-RNA-regulons_venn.png`) shows the number of *E. coli* genes associated with FNR peaks in the ChIP-seq experiment (pink), reported as differentially expressed in the RNA-seq analysis (green), or annotated as FNR targets in RegulonDB (violet), as well as the intersections between these gene sets. Supporting Information Tables S1 and S2 provide the complete data table used to generate these Venn diagrams. Depending on the peak-calling algorithm, the number of genes found at the intersection between the three gene lists (ChIP-seq, RNA-seq, and RegulonDB) will be quite small (38 for Macs2 peaks and 28 for Homer peaks) relative to the respective size of the compared gene sets. It is interesting to consider an interpretive guideline for the pairwise intersections or set memberships. The genes reported by both ChIP-seq (FNR binding) and RNA-seq (FNR transcriptional response) but not annotated in RegulonDB are likely to be direct FNR target genes, and might be considered to be added to RegulonDB, in an annotation track based on combined evidence from complementary high-throughput experiments. This would give 29 genes with Macs2 peaks and 25 with Homer peaks. It would be interesting to furthermore scan their promoter sequences in order to search instances of the FNR binding motif in order to predict binding-site locations, and consolidate the results. The genes detected as differentially expressed (RNA-seq) without any annotated FNR site (RegulonDB) or associated peak (Figure 13, pale green, on the Venn diagrams of Figure 13, covering, respectively, 160 and 167 genes for Macs2 and Homer) include genes located inside the target operons of FNR. Indeed, in bacteria, polycistronic transcripts are regulated by *cis*-acting elements located in the promoter of the operon leader gene. Consistently with this, 38 of these 167 genes (~23% when the analysis is led with Homer) have a very short upstream noncoding region (<55 bp) typical of intra-operon genes, whereas almost all the genes of the triple intersection (28 of 29) have larger upstream sequences typical of operon-leader genes. The remaining 77% of differentially expressed genes without associated ChIP-seq peak are likely to be indirect FNR targets, whose transcription might be affected via intermediate transcription factors that are themselves regulated by FNR. The genes associated with ChIP-seq peaks without transcriptional response (334 for Macs2, 119 for Homer) likely result from different effects:

Figure 13 Integration of ChIP-seq, RNA-seq results, and RegulonDB annotations. Venn diagrams show the intersections of the genes linked to ChIP-seq peaks (pink), those declared differentially expressed by the RNA-seq experiment (green), and those annotated as FNR target genes in RegulonDB (violet). These diagrams are automatically generated by the integration workflow, using the R library VennDiagram. (A) Results with the 411 ChIP-seq peaks reported by Macs2 with $q < 0.01$ and fold change between 2 and 50. (B) Results with the 166 ChIP-seq peaks reported by Homer.

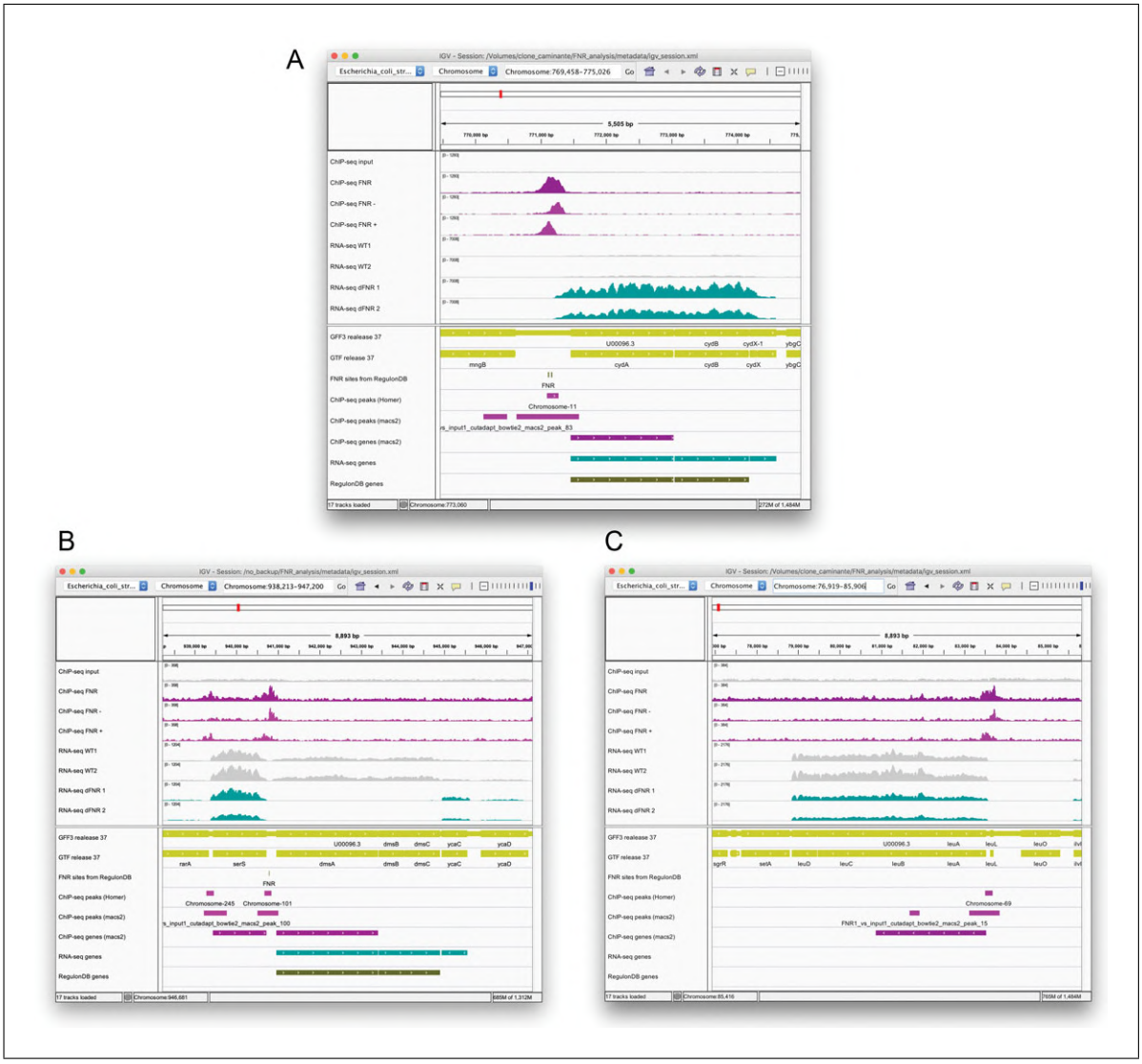


Figure 14 IGV snapshots of RNA-seq results for three illustrative operons. Middle panel, genome coverage profiles for the two replicas of the wild-type (gray) and FNR mutant (jade). Lower panel, genome annotations for the genes (yellow), FNR binding sites from RegulonDB (gray), differentially expressed genes (jade), and FNR target genes annotated in RegulonDB (dark olive). Shown are views of selected regions encompassing (A) the cydABX operon, (B) the dmsABC operon, and (C) the leuLABCD operon.

nonfunctional binding of the FNR factor under the experimental conditions of the study (missing co-activator, co-binding of a repressor); binding between two divergently transcribed transcription units, but regulating only one of them; or false positives from peak calling (e.g., regions with a high density of reads on one strand only, as discussed above).

Figure 14 highlights some illustrative examples of differentially expressed genes detected by DESeq2 or edgeR. For the cydABX operon (Fig. 14A), the FNR mutant (jade tracks on the genome coverage profiles) has an increased level of expression compared to the wild-type (gray tracks). Consistently with that result, this operon is repressed by FNR (Salmon et al., 2003), and it has two annotated FNR binding sites in RegulonDB, which overlap a strong peak detected by both Homer and Macs2 in the ChIP-seq results.

The dmsABC operon also exemplifies the genes found at the triple intersection: it is regulated by FNR (Melville & Gunsalus, 1996), and, consistently, it has one TF binding site listed in RegulonDB, and is reported by both the ChIP-seq and RNA-seq experiments (Fig. 14B).

A more subtle example is the leuLABCD operon (Fig. 14C): RNA-seq coverage profiles also reveal reduced expression, although the differential expression analysis did not report the presence of any significant gene, due to the stringent thresholds applied to both adjusted *P* value (<0.05) and fold change (>2). This operon encodes the enzymes responsible for the biosynthesis of leucine from valine. It has no binding sites annotated in RegulonDB for the FNR transcription factor, and based on the RNA-seq results only, several possibilities could be invoked to explain this inconsistency: the leu operon might (i) be indirectly regulated by FNR via another transcription factor, (ii) be a direct target of FNR whose binding sites have not yet been characterized, or (iii) be a false-positive. This situation can be clarified by analyzing the ChIP-seq profiles, since we observe a clear peak upstream of the operon, detected by both Macs2 and Homer (Fig. 14C), supporting the evidence for a direct regulation of the leu operon by FNR.

In summary, a detailed analysis and human-based interpretation of combined RNA-seq and ChIP-seq data is worthwhile as a means to go beyond the gene lists returned by the automatic comparison of target genes predicted by ChIP-seq and RNA-seq experiments.

COMMENTARY

Background Information

Next-generation sequencing (NGS) technologies (Schuster, 2007) emerged in 2007 with the development of several approaches for massively parallel sequencing of short DNA sequences (a few tens of base pairs per sequence). This unprecedented gain in sequencing speed was mobilized for a wide variety of applications: genome sequencing, transcriptome (RNA-seq), genome-wide binding location analysis (ChIP-seq), chromatin conformation (Hi-C), metagenomics, and many others. Research projects based on NGS typically lead to the situation where the biologist performs experiments, sends the samples to a sequencing center, and receives a link to download several gigabases of raw sequences known as "short reads." Since 2007, a wide variety of software tools has been developed to handle NGS data and extract relevant information (Pepke et al., 2009).

Proper use of such software requires a good understanding of their parameters, strengths, and weaknesses. Beyond the choice and parameterization of each particular tool, it has become crucial to formalize their wiring by implementing workflows that ensure traceability and reproducibility of all the steps used to produce the results from the raw data. Many alternative software systems can be used to manage the development and execution of analysis workflows. Among them, Galaxy (Goecks, Nekrutenko, & Taylor, 2010) became highly popular because it offers an immediate access through a graphical interface to biologists with no experience in the Unix terminal. Snakemake (Köster & Rahmann, 2012) offers a complementary solution to achieve the same goals—developing, managing, and running NGS workflows—in the Unix command-line environment. Snakemake is currently being adopted by a growing number of bioinformaticians as well as experimental biologists willing to get one step further in the analysis of their own data. The goal of SnakeChunks is to facilitate the conception and use of NGS workflows by encapsulating Snakemake commands in a library of modular rules (one per tool) that can be combined in various ways to build and customize workflows (Fig. 2).

Critical Parameters

Control samples

When analyzing binding signals (ChIP-seq) or transcription signals (RNA-seq), it is crucial to generate appropriate control experiments, in order to measure differences in signal against a proper background signal, and thus avoid the detection of false positives. This is especially important when analyzing ChIP-seq data, since false peaks can arise from biases in the experiments: nonhomogeneous sonication of DNA due nonhomogeneous aperture of the chromatin, GC biases arising during PCR amplification of the fragments, low-complexity regions of the genome, and so on. Different types of controls can be used to estimate the background probabilities of read mapping in the different regions of the genome, including (1) sequencing genomic DNA without immunoprecipitation; (2) using "mock IP," i.e., performing the immunoprecipitation with a nonspecific antibody; or (3) artificially knocking out the expression of the TF of

interest. Irrespective of the method used, the control sequences are generally denoted as "input" for the peak-calling programs. In the study by Myers et al. (2013), genomic DNA was used as input. In the case of RNA-seq, knocked-out TFs or overexpressed TFs can be compared against WT samples. In this study, samples with an inactivated FNR protein were compared against WT strains.

Number of replicates

When performing biological experiments, it is crucial to account for the unavoidable variability intrinsic to living organisms. RNA-seq experiments are no exception, and it has been demonstrated that the greater the number of replicates, the more sensitive the detection of differentially expressed genes (Schurch et al., 2016). Designing experiments with a high number of replicates enables the analysis to distinguish subtle but relevant changes in expression from spurious fluctuations due to biological variability.

Choice of a read mapper

Read mapping is generally the most time- and resource-consuming task of RNA-seq and ChIP-seq data analysis. For the FNR study case developed in this article, the complete ChIP-seq workflow runs in a few minutes, whereas the RNA-seq workflows takes several hours. The modularity of the SnakeChunks library enabled us to run the same workflow with three alternative read-mapping tools: BWA (Li & Durbin, 2009), bowtie2 (Langmead & Salzberg, 2012), and subread-align (Liao, Smyth, & Shi, 2013). For this particular dataset, BWA runs approximately three times as fast as the two other algorithms, while giving very similar mapping rates. However, we experienced opposite rankings of tool performance with other datasets and reference genomes. The choice and parameterization of a read mapper should thus be considered as critical step, which has to be tuned in a case-specific way to optimize a workflow.

Troubleshooting

The Snakemake workflow management system is equipped with its own mechanisms for detecting, reporting, and fixing problems. Trouble is reported by red messages displayed on the terminal indicating the kind of problems and—when possible—suggested ways to fix them.

Advanced Parameters

Proper parameterization of the workflow is the key to optimize both computing efficiency and the biological relevance of the results.

Parameters can be changed either by modifying the YAML-formatted configuration file in the metadata (see Support Protocol) or with the option `--config` in the Snakemake command line (see example in Basic Protocol 1, step 3).

With the popularization of RNA-seq for transcriptome studies, the number of samples per research project has been expanding in recent publications. A crucial parameter will be the ability to keep up with increasing storage needs and to parallelize computation for large studies. The FNR case study discussed in this unit was intentionally selected for its small number of replicates per condition, but for wider-scale studies the number of simultaneous jobs handled by Snakemake should be adapted to the number of CPUs of the computing system (option `-j` option).

We also make a frequent use of the Snakemake option `-n`, which prints out all the commands required to complete a workflow, without actually executing them (as a dry run). This gives the user the ability to check that a command is properly parameterized before running it, which can be valuable when applying hours-long tasks to multiple samples.

Suggestions for Further Analysis

The main goal of the SnakeChunks library is to ensure the reproducibility of the analyses. This is why we recommend keeping a copy of the library with each dataset analyzed in order to ensure consistency between the results and the precise version of the library used to generate them. This is particularly crucial in the case of publication, so that readers can actually reproduce the analyses performed.

The use of Conda also enables the user to keep control over the software environment, and is in accordance with the FAIR Principles (Wilkinson et al., 2016).

A natural extension of this work will be to take advantage of SnakeChunks' flexibility in order to assess the impact of tool and parameter choice on the biological relevance of the results, and to optimize workflows by evaluating the correspondence between the lists of genes returned by combining ChIP-seq and RNA-seq results and those already annotated in RegulonDB for well-characterized transcription factors.

Acknowledgments

This work is funded by France Génomique, US National Institutes of Health grant

GM0110597, and FOINS-CONACYT–Fronteras de la Ciencia 2015, ID 15. L.C.K. is funded by a PhD grant from the Ecole Doctorale des Sciences de la Vie et de la Santé, Aix-Marseille Université.

We acknowledge the Institut Français de Bioinformatique (IFB) and Christophe Blanchet for the use of virtual machines on the IFB cloud, which enabled us to assess the portability and reproducibility of the workflows, as well as the Sequanix development team (Thomas Cokelaer, Dimitri Desvillechabrol, and Rachel Legendre), who helped us to port SnakeChunks to Conda and Sequanix.

Literature Cited

Andrews, S. (2010). FastQC: A quality control tool for high throughput sequence data. Retrieved from http://www.bioinformatics.babraham.ac.uk/projects/fastqc.

Blattner, F. R., Plunkett, G., Bloch, C. A., Perna, N. T., Burland, V., Riley, M., … Shao, Y. (1997). The complete genome sequence of *Escherichia coli* K-12. *Science*, *277*(5331), 1453–1462. doi: 10.1126/science.277.5331.1453.

Cadby, I. T., Faulkner, M., Cheneby, J., Long, J., van Helden, J., Dolla, A., & Cole, J. A. (2017). Coordinated response of the *Desulfovibrio desulfuricans* 27774 transcriptome to nitrate, nitrite and nitric oxide. *Scientific Reports*, *7*(1), 16228. doi: 10.1038/s41598-017-16403-4.

Castro-Mondragon, J. A., Rioualen, C., Contreras-Moreira, B., & van Helden, J. (2016). RSAT::Plants: Motif discovery in ChIP-seq peaks of plant genomes. *Methods in Molecular Biology 1482*, 297–322. doi: 10.1007/978-1-4939-6396-6_19.

Desvillechabrol, D., Legendre, R., Rioualen, C., Bouchier, C., van Helden, J., Kennedy, S., & Cokelaer, T. (2018). Sequanix: A dynamic graphical interface for Snakemake workflows. *Bioinformatics*, *34*(11), 1934–1936. doi: 10.1093/bioinformatics/bty034.

Feng, J., Liu, T., & Zhang, Y. (2011). Using MACS to identify peaks from ChiP-seq data. *Current Protocols in Bioinformatics*, *34*, 2.14.1–2.14.14. doi: 10.1002/0471250953.bi0214s34.

Galagan, J., Lyubetskaya, A., & Gomes, A. (2012). ChIP-Seq and the complexity of bacterial transcriptional regulation. In M. G. Katze (Ed.), *Systems biology* (pp. 43–68). Berlin, Heidelberg: Springer. Retrieved from https://link.springer.com/chapter/10.1007/82_2012_257.

Galagan, J. E., Minch, K., Peterson, M., Lyubetskaya, A., Azizi, E., Sweet, L., … Schoolnik, G. K. (2013). The *Mycobacterium tuberculosis* regulatory network and hypoxia. *Nature*, *499*, 178–183. doi: 10.1038/nature12337.

Gama-Castro, S., Salgado, H., Santos-Zavaleta, A., Ledezma-Tejeida, D., Muñiz-Rascado, L., García-Sotelo, J. S., … Collado-Vides, J. (2016). RegulonDB version 9.0: High-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Research*, *44*(D1), D133–143. doi: 10.1093/nar/gkv1156.

Goecks, J., Nekrutenko, A., & Taylor, J., & Galaxy Team. (2010) Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, *11*(8), R86. doi: 10.1186/gb-2010-11-8-r86.

Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y. C., Laslo, P., … Glass, C. K. (2010). Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular Cell*, *38*(4), 576–589. doi: 10.1016/j.molcel.2010.05.004.

Jacob, F., & Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, *3*, 318–356. doi: 10.1016/S0022-2836(61)80072-7.

Johnson, D. S., Mortazavi, A., Myers, R. M., & Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science*, *316*(5830), 1497–1502. doi: 10.1126/science.1141319.

Köster, J., & Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, *28*, 2520–2522. doi: 10.1093/bioinformatics/bts480.

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 2012 Mar 4;*9*(4), 357–359. doi: 10.1038/nmeth.1923.

Lazazzera, B. A., Bates, D. M., & Kiley, P. J. (1993). The activity of the *Escherichia coli* transcription factor FNR is regulated by a change in oligomeric state. *Genes & Development*, *7*(10), 1993–2005. doi: 10.1101/gad.7.10.1993.

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, *25*, 1754–1760. doi: 10.1093/bioinformatics/btp324.

Liao, Y., Smyth, G. K., & Shi, W. (2013). The Subread aligner: Fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Research*, *41*(10), e108.

Liao, Y., Smyth, G. K., & Shi, W. (2014). Featurecounts: An efficient general-purpose program for assigning sequence reads to genomic features. *Bioinformatics*, *30*(7), 923–930. doi: 10.1093/bioinformatics/btt656.

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*, 550. doi: 10.1186/s13059-014-0550-8.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.Journal*, *17*(1), 10–12.doi: 10.14806/ej.17.1.200.

Melville, S. B., & Gunsalus, R. P. (1996). Isolation of an oxygen-sensitive FNR protein of

Escherichia coli: Interaction at activator and repressor sites of FNR-controlled genes. *Proceedings of the National Academy of Sciences of the United States of America*, *93*(3), 1226–1231. doi: 10.1073/pnas.93.3.1226.

Myers, K. S., Yan, H., Ong, I. M., Chung, D., Liang, K., Tran, F., … Kiley, P. J. (2013). Genome-scale analysis of *Escherichia coli* FNR reveals complex features of transcription factor binding. *PLoS Genetics*, *9*(6), e1003565. doi: 10.1371/journal.pgen.1003565.

Nguyen, N. T. T., Contreras-Moreira, B., Castro-Mondragon, J. A., Santana-Garcia, W., Ossio, R., Robles-Espinoza, C. D., … Thomas-Chollier, M. (2018). RSAT 2018: Regulatory sequence analysis tools 20th anniversary. *Nucleic Acids Research*, *46*(W1), W209–W214. doi: 10.1093/nar/gky317.

Pepke, S., Wold, B., & Mortazavi, A. (2009). Computation for ChIP-seq and RNA-seq studies. *Nature Methods*, *6*, S22–S32. doi: 10.1038/nmeth.1371.

Pérez-Rueda, E., & Collado-Vides, J. (2000). The repertoire of DNA-binding transcriptional regulators in Escherichia coli K-12. *Nucleic Acids Research*, *28*(8), 1838–1847. doi: 10.1093/nar/28.8.1838.

Robertson, G., Hirst, M., Bainbridge, M., Bilenky, M., Zhao, Y., Zeng, T., … Jones, S. (2007). Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods*, *4*(8), 651–657. doi: 10.1038/nmeth1068.

Robinson, J. T., Thorvaldsdóttir, H., Winckler, W., Guttman, M., Lander, E. S., Getz, G., & Mesirov, J. P. (2011). Integrative genomics viewer. *Nature Biotechnology*, *29*(1), 24–26. doi: 10.1038/nbt.1754.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. doi: 10.1093/bioinformatics/btp616.

Salmon, K., Hung, S., Mekjian, K., Baldi, P., Hatfield, G. W., & Gunsalus, R. P. (2003). Global gene expression profiling in *Escherichia coli* K12: The effect of oxygen availability and FNR. *Journal of Biological Chemistry*, *278*(32), 29837–29855. doi: 10.1074/jbc.M213060200.

Schuster, S. C. (2007). Next-generation sequencing transforms today's biology. *Nature Methods*, *5*(1), 16–18. doi: 10.1038/nmeth1156.

Schurch, N. J., Schofield, P., Gierliński, M., Cole, C., Sherstnev, A., Singh, V., … Barton, G. J. (2016). How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA*, *22*(6), 839–851. doi: 10.1261/rna.053959.115.

Thomas-Chollier, M., Herrmann, C., Defrance, M., Sand, O., Thieffry, D., & Van Helden, J. (2012). RSAT peak-motifs: Motif analysis in full-size ChIP-seq datasets. *Nucleic Acids Research*, *40*(4), e31. doi: 10.1093/nar/gkr1104.

Thorvaldsdóttir, H., Robinson, J. T., & Mesirov, J. P. (2013). Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, *14*(2), 178–192. doi: 10.1093/bib/bbs017.

Tsagmo Ngoune, J. M., Njiokou, F., Loriod, B., Kame-Ngasse, G., Fernandez-Nunez, N., Rioualen, C., … Geiger, A., (2017). Transcriptional profiling of midguts prepared from *Trypanosoma/T. congolense*-positive *Glossina palpalis palpalis* collected from two distinct Cameroonian foci: Coordinated signatures of the midguts. remodeling as *T. congolense*-supportive niches. *Frontiers in Immunology*, *8*, 876. doi: 10.3389/fimmu.2017.00876.

van Helden, J., Ríos, A. F., & Collado-Vides, J. (2000). Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research*, *28*(8), 1808–1818 doi: 10.1093/nar/28.8.1808.

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., … Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship [Comments and Opinion]. Retrieved December 1, 2017, from https://www.nature.com/articles/sdata20161.

Key References

Gama-Castro et al. (2016). See above.
Describes RegulonDB version 9

Köster & Rahmann (2012). See above.
Describes the Snakemake workflow engine.

Myers et al. (2013). See above.
Publication associated to the dataset used in this protocols.

Internet Resources

Snakemake: Retrieved from http://snakemake.readthedocs.io

SnakeChunks GitHub repository: Retrieved from https://github.com/SnakeChunks/SnakeChunks

SnakeChunks documentation & tutorials: Retrieved from http://snakechunks.readthedocs.io

FastQC: Retrieved from http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

UCSC file format description: Retrieved from https://genome.ucsc.edu/FAQ/FAQformat.html

Chapter 4.

Integration of high-throughput data within a reference framework

Problematic

E. coli K-12 is to date the best characterized prokaryotic organism, and a significant portion of its transcriptional regulatory network is known and available for display and for use through the RegulonDB portal. Still, it remains incomplete: about a third of its predicted transcription factors are not experimentally proven to perform actual regulation, and most of those that do have evidence for regulation were not studied genome-wide. High-throughput technologies now allow for genome-wide detection of binding sites, for instance via ChIP-seq, ChIP-exo, gSELEX or DAP-seq; and transcriptional profiling is now routinely performed using RNA-seq. However, until recently, there was no online resource that would allow one to consult or make use of those data, together with the classic data.

In this chapter, I present my contributions to an article that undertakes the task of gathering, standardizing more than 2,000 datasets of high-throughput data that are relevant to *E. coli* genomic organization and regulation, and integrating them with the data resulting from classic low-throughput experiments and literature curation on a single portal: RegulonDB HT (Tierrafría, Rioualen et al., 2022).

Definition of the framework

The diversity of data to be integrated and objects to be manipulated proved to be a challenge, despite the well-established standards that have been developed in RegulonDB over the years, and have been evolving with the constant addition of new biological knowledge. A lot of thought was put into a new framework that would allow

us to gather and process data from a variety of technologies, and produce uniform datasets.

We defined *collections* as sets of data defining objects of distinct types, namely transcription start sites, transcription termination sites, transcription units, gene expression and transcription factor binding. Additionally, we distinguished *subcollections* of TF binding datasets that were produced using different technologies (ChIP-seq, ChIP-exo, gSELEX and DAP-seq). Each collection and subcollection is composed of a certain number of *datasets* (Figure 15).

We defined a *dataset* as a piece of data generated using a given technology, producing a certain type of object (and thus pertaining to a given collection), and associated to specific growth conditions as defined by the Microbiological Condition Ontology (Tierrafría et al., 2019), and by additional metadata.

We designed *metadata* tables of datasets based on a common format for each collection and subcollection of objects. Each table contains one dataset per row, and one column per attribute of the dataset: technology used, growth conditions, author and publication information, database identifiers, and many more.

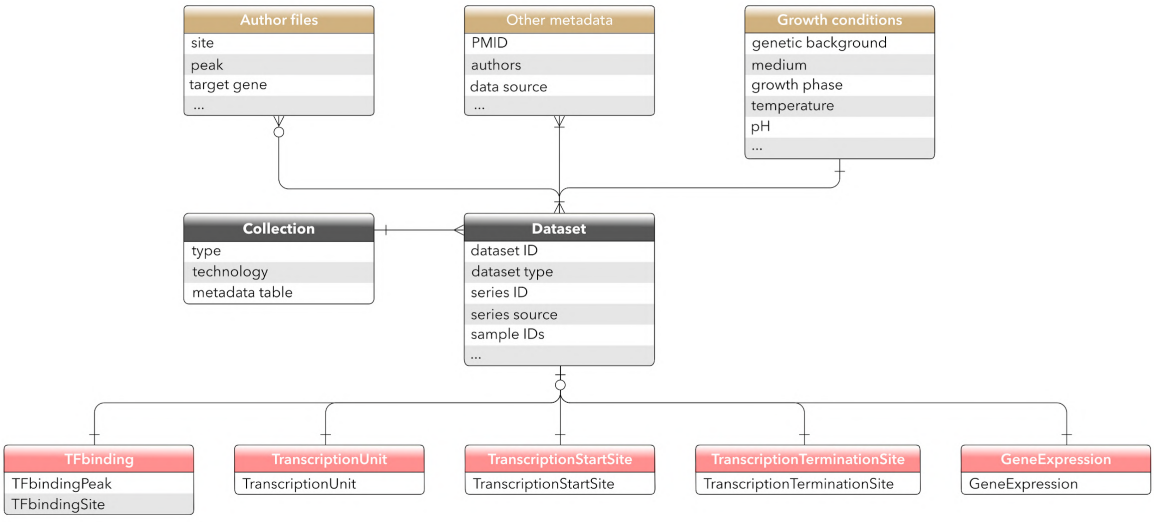


Figure 15. Data model in RegulonDB HT. Adapted from figure 1 (Tierrafría, Rioualen et al., 2022).

Uniform datasets of genomic features

Building on the work described in Chapter 2, I took on the task of gathering datasets from a number of distinct sources and publications, and processing them in order to generate uniform datasets of TSSs, TUs and TTSs. This presented challenges, given the variety of formats used in the original sources, and the presence of obsolete information.

I updated the TSS and TU collections previously generated (Chapter 2) with new sources (Conway et al., 2014; Ju et al., 2019), and generated 16 datasets containing 68,049 TSSs and 5 datasets containing 5,326 transcription units. I created a TTS collection using the information available in the TU collection, and generated 5 datasets containing 12,347 TTSs (Table 3). All of these datasets were formatted to standard bed files. Given the case, genes and coordinates were updated to the latest genome version using the EcoliGenes library (Chapter 1).

Finally, I mapped the TSS collection against the original collection from RegulonDB. The set of reference used for the comparison was generated by extracting all of RegulonDB's promoters associated with classic strong evidence (Figure 7 from the article).

Dataset ID	Growth Condition	Features	Reference
Transcription Units			
TU0001	ORGANISM:Escherichia coli |MEDIUM:LB medium |GROWTH_PHASE:Exponential phase	3,179	Ju et al., 2019
TU0002	ORGANISM:Escherichia coli |MEDIUM:LB medium |GROWTH_PHASE:Stationary phase	1,916	Ju et al., 2019
TU0003	ORGANISM:Escherichia coli BW38028 |MEDIUM:MOPS |OPTICAL_DENSITY:OD600 of 0.4 |GROWTH_PHASE:Exponential phase	2,566	Conway et al., 2014
TU0004	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:wild type |MEDIUM:M9 minimal medium |GROWTH_PHASE:Exponential phase	2,458	Yan et al., 2018
TU0005	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:wild type |MEDIUM:rich medium |GROWTH_PHASE:Exponential phase	2,210	Yan et al., 2018
Transcription Start Sites			
DS0001	ORGANISM:'Escherichia coli str. K-12 substr. MG1655' |GENETIC_BACKGROUND:'wild type' |MEDIUM:LB |TEMPERATURE:'37.0 C' |OPTICAL_DENSITY:'OD600 of 2' |GROWTH_PHASE:'stationary phase'	12,016	Thomason et al., 2014

Dataset ID	Growth Condition	Features	Reference
DS0002	ORGANISM:'Escherichia coli str. K-12 substr. MG1655' |GENETIC_BACKGROUND:'wild type' |MEDIUM:M63 |MEDIUM_SUPPLEMENTS:glucose 0.2%' 'thiamine(1+)' |TEMPERATURE:'37.0 C' |OPTICAL_DENSITY:'OD600 of 0.4' |GROWTH_PHASE:'exponential phase'	11,945	Thomason et al., 2014
DS0003	ORGANISM:'Escherichia coli str. K-12 substr. MG1655' |GENETIC_BACKGROUND:'wild type' |MEDIUM:LB |TEMPERATURE:'37.0 C' |OPTICAL_DENSITY:'OD600 of 0.4' |GROWTH_PHASE:'exponential phase'	8,504	Thomason et al., 2014
DS0004	ORGANISM:'Escherichia coli str. K-12 substr. MG1655' |MEDIUM:LB |AERATION:aerobic |TEMPERATURE:'37.0 C' |pH:'pH 7.4' |OPTICAL_DENSITY:'OD600 from 0.4 to 0.6' |GROWTH_PHASE:'exponential phase'	4,353	Ju et al., 2019
DS0005	ORGANISM:'Escherichia coli str. K-12 substr. MG1655' |MEDIUM:LB |AERATION:aerobic |TEMPERATURE:'37.0 C' |pH:'pH 7.4' |OPTICAL_DENSITY:OD600 above 2.0 |GROWTH_PHASE:'stationary phase'	4,033	Ju et al., 2019
DS0006	ORGANISM:'Escherichia coli BW38028' |GENETIC_BACKGROUND:'wild type' |MEDIUM:'MOPS minimal medium' |MEDIUM_SUPPLEMENTS:'glucose 0.2%' |AERATION:dissolved oxygen above 40% of saturation |TEMPERATURE:'37.0 C' |pH:'pH 7.4' |VESSEL_TYPE:fermenter	2,122	Conway et al., 2014
DS0007	ORGANISM:'Escherichia coli str. K-12 substr. MG1655' |MEDIUM:'DSMZ Medium 382' |MEDIUM_SUPPLEMENTS:'glucose 0.2%' |TEMPERATURE:'37.0 C' |OPTICAL_DENSITY:OD600 from 0.55 to 0.6 |GROWTH_PHASE:'late exponential phase'	2,186	Yan et al., 2018
DS0008	ORGANISM:'Escherichia coli str. K-12 substr. MG1655' |MEDIUM:'LB medium, Lennox' |TEMPERATURE:'37.0 C' |pH:pH 7.2 |OPTICAL_DENSITY:OD600 from 0.55 to 0.6 |GROWTH_PHASE:'late exponential phase'	1,902	Yan et al., 2018
DS0009	ORGANISM:Escherichia coli str. K-12 substr. MG1655' |TEMPERATURE:'30.0 C'	1,468	Mendoza-Vargas et al., 2009
DS0010	ORGANISM:Escherichia coli str. K-12 substr. MG1655' |TEMPERATURE:'30.0 C'	296	Mendoza-Vargas et al., 2009
DS0011	ORGANISM:Escherichia coli str. K-12 substr. MG1655' |TEMPERATURE:'37.0 C' |AGITATION_SPEED:300 rpm	5,197	Salgado et al., 2013
DS0012	[M9 + 0.2% glycerol, cells grown with shaking at 30°C]	5,647	Wade lab
DS0013	GROWTH_PHASE: Exponential phase	1,926	Cho et al., 2014
DS0014	[Glutamine as source of nitrogen]	2,230	Cho et al., 2014
DS0015	[Heat shock]	1,900	Cho et al., 2014
DS0016	GROWTH_PHASE:Stationary phase	2,533	Cho et al., 2014
Transcription Termination Sites			
TR0001	ORGANISM:Escherichia coli |MEDIUM:LB medium |GROWTH_PHASE:Exponential phase	1,473	Ju et al., 2019
TR0002	ORGANISM:Escherichia coli |MEDIUM:LB medium |GROWTH_PHASE:Stationary phase	1,352	Ju et al., 2019

Dataset ID	Growth Condition	Features	Reference
TR0003	ORGANISM:Escherichia coli BW38028 |MEDIUM:MOPS |OPTICAL_DENSITY:OD600 of 0.4 |GROWTH_PHASE:Exponential phase	1,774	Conway et al., 2014
TR0004	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:wild type |MEDIUM:M9 minimal medium |GROWTH_PHASE:Exponential phase	352	Yan et al., 2018
TR0005	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:wild type |MEDIUM:rich medium |GROWTH_PHASE:Exponential phase	375	Yan et al., 2018

Table 3. Summary of the HT datasets generated and their associated growth conditions and references.

Transcription factor comparison

I performed a comparison of the transcription factors present in each subcollection of TF binding datasets and those present in RegulonDB. I used the EcoliGenes library (Chapter 1) in order to translate TF names and synonyms into their reference name, and properly manage hetero-dimeric TFs and their subunits (Figure 5a in the article, see below).

I integrated those TFs with the putative TFs obtained through computational predictions and presented in Chapter 1, and summarized the result by grouping all TFs into 3 categories: confirmed TFs from RegulonDB, predicted TFs from various sources (Pérez-Rueda et al., 2015; Flores-Bautista et al., 2020; Kim et al., 2021), and potential TFs associated with HT experiments (Tierrafría, Rioualen et al., 2022) (Figure 16ab). 46 of the previously predicted TFs are associated with at least one peak in one HT dataset, bringing new pieces of evidence to confirm their regulatory role. Additionally, 4 potential TFs that were neither confirmed RegulonDB TFs nor predicted TFs were associated with HT experiments, where all of them were assigned several binding peaks. Finally, 144 predicted TFs remain without HT datasets that would back up their potential regulatory role. They include all of the 58 predicted TFs from the deep learning approach (Kim et al., 2021), which remain without evidence to back up the predictions.

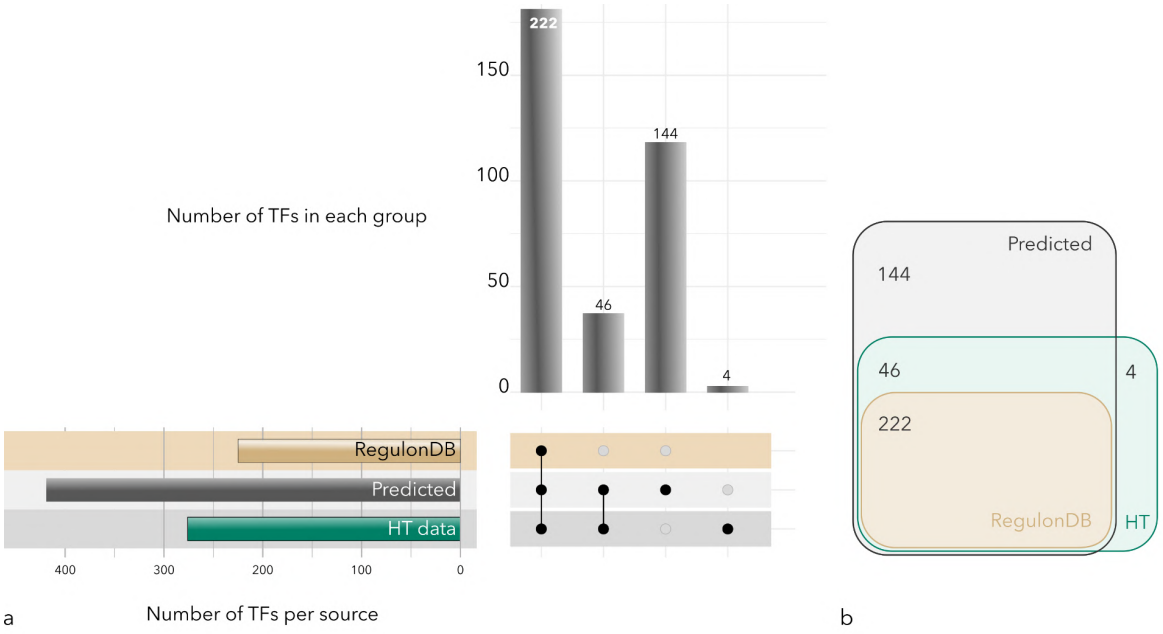


Figure 16. Comparison of TFs from RegulonDB, proteins predicted or annotated as putative TFs and putative TFs with associated high-throughput data.

Considering this newly available data, new pieces of evidence should allow to greatly increase the collection of confirmed TFs in the near future, getting closer to a total of 300 TFs, a common estimate of the total number of TFs in *E. coli* K-12.

Uniformly-processed ChIP-seq datasets

I processed 28 datasets from the ChIP-seq subcollection using the Snakechunks library of workflows (Chapter 3). I built a workflow that takes no more than the ChIP-seq *metadata* table as an input, using the following attributes: source database name, series ID, samples replicates experiment ID, samples replicates control ID, library layout and TF name (Figure 17a).

For each dataset, I extracted the full original metadata from their source database, and given the case, merged or completed them using the tools ffq and pyrasdb (Choudhary, 2019; Gálvez-Merchán et al., 2022). Using this information, I built a common directory structure for all datasets, custom workflow configuration files for each, and I downloaded all of the raw sequencing files in fastq format. For each dataset, I then ran the "quality control" and "mapping" workflows available in SnakeChunks (Rioualen et al., 2019; Chapter 3) using cutadapt for read-trimming (Martin, 2011), bowtie 2 for the

alignment (Langmead and Salzberg, 2012), and multiQC to generate complete quality reports before and after the preprocessing (Ewels et al., 2016). Finally, I designed a new workflow to perform peak-calling using macs3 (Feng et al., 2011), identify sites in peaks with RSAT matrix-scan (Turatsinze et al., 2008), and build a dataset-specific TF motif using the sites identified and the RSAT convert-matrix tool (Santana-Garcia et al., 2022). Each dataset resulted in one peak file, one site file (both bed-formatted) and one PSSM file, as well as several graphical reports (Figure 17b). Additionally, I mapped peaks and sites with the reference binding site set from RegulonDB (Figure 6 from the article).

The metadata table for the ChIP-seq subcollection allowed to customize each dataset processing depending on the TF and the library layout used in each experiment, while using common tools and cutoffs for the different steps of the analysis performed using the SnakeChunks workflows, ensuring flexibility as well as congruence. The output files were integrated into the RegulonDB HT portal, and displayed as tables as well as in a genome browser, together with classic data for easy comparison (Figure 17c).

Overall, the ChIP-seq collection includes 29 datasets corresponding to 12 different TFs (Table 4), of which 28 were processed using my pipeline (one dataset does not come with raw data), and 27 are associated with curated author files (two datasets are not associated with a publication). All of this data is available on the RegulonDB-HT portal.

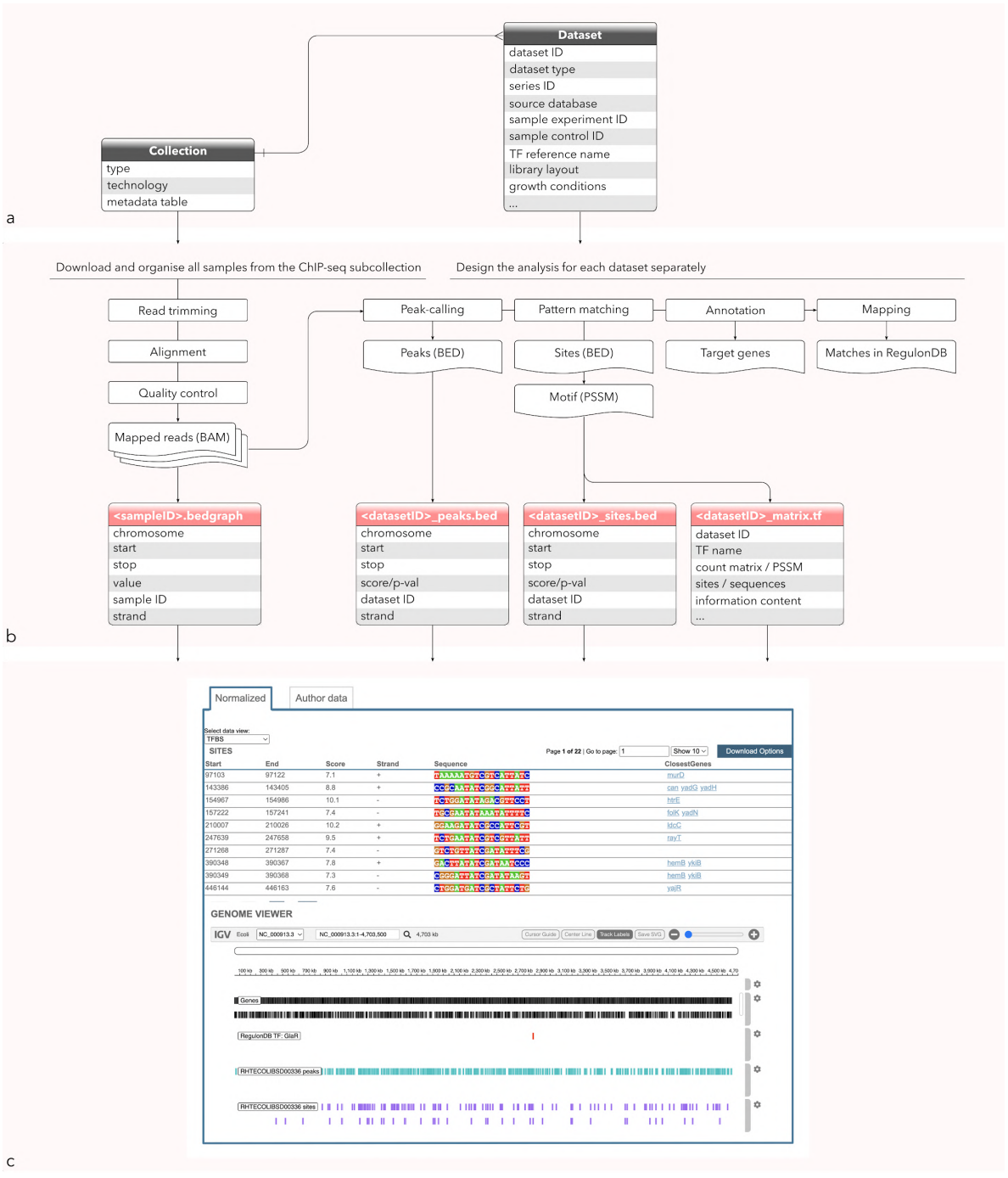


Figure 17. Integration of the ChIP-seq collection processing within the RegulonDB HT framework. **a.** Datasets are manually curated from literature and databases, and their attributes are gathered into the metadata table. **b.** All of the samples are automatically downloaded, and datasets analyses are designed and run using their specific properties (library layout, TF chipped) as well as common tools (trimming, mapping, peak-calling). **c.** The result files are integrated in the RegulonDB HT portal for browsing and downloading.

Dataset ID	Growth Conditions	TF	Reference
BSCS001	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:plasmid pT7-FLAG-4 (IPTG-induced CsiR) mutant |MEDIUM:LB medium |MEDIUM_SUPPLEMENTS:isopropyl beta-D-thiogalactopyranoside 1 mM |TEMPERATURE:37 °C	GlaR	Aquino et al., 2017
BSCS002	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM_SUPPLEMENTS:glucose 0.2% |AERATION:N2 95% and CO2 5% |TEMPERATURE:37 °C |OPTICAL_DENSITY:OD600 of 0.3 |GROWTH_PHASE:mid exponential phase	FNR	Myers et al., 2013
BSCS003	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM_SUPPLEMENTS:glucose 0.2%; iron(2+) sulfate (anhydrous) 10 µM |AERATION:70% N2, 5% CO2, and O2 25% |TEMPERATURE:37 °C |OPTICAL_DENSITY:OD600 from 0.3 to 0.35 |GROWTH_PHASE:mid exponential phase	Fur	Beauchene et al., 2015
BSCS004	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM_SUPPLEMENTS:glucose 0.2%; iron(2+) sulfate (anhydrous) 10 µM |AERATION:N2 95% and CO2 5% |TEMPERATURE:37 °C |OPTICAL_DENSITY:OD600 from 0.3 to 0.35 |GROWTH_PHASE:mid exponential phase	Fur	Beauchene et al., 2015
BSCS005	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:lacZ knockout mutant; tonB knockout mutant; feoA knockout mutant; zupT knockout mutant |MEDIUM:MOPS minimal medium |MEDIUM_SUPPLEMENTS:glucose 0.2%; iron(2+) sulfate (anhydrous) 1.0 µM |AERATION:N2 95% and CO2 5% |TEMPERATURE:37 °C |OPTICAL_DENSITY:OD600 from 0.3 to 0.35 |GROWTH_PHASE:mid exponential phase	Fur	Beauchene et al., 2015
BSCS006	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:hns-3xflag |MEDIUM:LB medium, Luria-NaCl 0.5% |AERATION:aerobic |TEMPERATURE:37 °C |GROWTH_PHASE:early exponential phase	H-NS	Kahramanoglou et al., 2011
BSCS007	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:hns-3xflag |MEDIUM:LB medium, Luria-NaCl 0.5% |AERATION:aerobic |TEMPERATURE:37 °C |GROWTH_PHASE:mid exponential phase	H-NS	Kahramanoglou et al., 2011
BSCS008	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:hns-3xflag |MEDIUM:LB medium, Luria-NaCl 0.5% |AERATION:aerobic |TEMPERATURE:37 °C |GROWTH_PHASE:stationary phase	H-NS	Kahramanoglou et al., 2011
BSCS009	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:hns-3xflag |MEDIUM:LB medium, Luria-NaCl 0.5% |AERATION:aerobic |TEMPERATURE:37 °C |GROWTH_PHASE:transition to stationary phase	H-NS	Kahramanoglou et al., 2011
BSCS010	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:fis-3xflag |MEDIUM:LB medium, Luria-NaCl 0.5% |AERATION:aerobic |TEMPERATURE:37 °C |GROWTH_PHASE:early exponential phase	Fis	Kahramanoglou et al., 2011
BSCS011	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:fis-3xflag |MEDIUM:LB medium, Luria-NaCl 0.5% |AERATION:aerobic |TEMPERATURE:37 °C |GROWTH_PHASE:mid exponential phase	Fis	Kahramanoglou et al., 2011
BSCS012	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:plasmid pT7-FLAG-4 (IPTG-induced nac) mutant|MEDIUM:LB medium |MEDIUM_SUPPLEMENTS:isopropyl beta-D-thiogalactopyranoside 1 mM |TEMPERATURE:37 °C	Nac	Aquino et al., 2017
BSCS013	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:plasmid pT7-FLAG-4 (IPTG-induced ntrC) mutant	NtrC	Aquino et al., 2017

Dataset ID	Growth Conditions	TF	Reference
	|MEDIUM:LB medium |MEDIUM__SUPPLEMENTS:isopropyl beta-D-thiogalactopyranoside 1 mM |TEMPERATURE:37 °C		
BSCS014	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:glnG knockout mutant; glnG-flag |MEDIUM:Gutnick minimal medium |MEDIUM__SUPPLEMENTS:Ho-LE trace elements; glucose 0.4%; ammonium chloride 3 mM |TEMPERATURE:37 °C |AGITATION__SPEED:200 rpms	NtrC	Brown et al., 2014
BSCS015	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:plasmid pT7-FLAG-4 (IPTG-induced ompR) mutant |MEDIUM:LB medium |MEDIUM__SUPPLEMENTS:isopropyl beta-D-thiogalactopyranoside 1 mM |TEMPERATURE:37 °C	OmpR	Aquino et al., 2017
BSCS016	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.2%; leucine 0.2%; isoleucine 0.2%; valine 0.2% |TEMPERATURE:37 °C |OPTICAL__DENSITY:OD600 from 0.15 to 0.25 |GROWTH__PHASE:exponential phase |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS017	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.2%; leucine 0.2%; isoleucine 0.2%; valine 0.2% |TEMPERATURE:37 °C |OPTICAL__DENSITY:OD600 from 1.8 to 2.2 |GROWTH__PHASE:transition point |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS018	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.2%; leucine 0.2%; isoleucine 0.2%; valine 0.2% |TEMPERATURE:37 °C |GROWTH__PHASE:stationary phase |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS019	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.4%; ACGU; EZ |TEMPERATURE:37 °C |OPTICAL__DENSITY:OD600 from 0.15 to 0.25 |GROWTH__PHASE:exponential phase |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS020	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.4%; ACGU; EZ |TEMPERATURE:37 °C |OPTICAL__DENSITY:OD600 from 2.3 to 2.7 |GROWTH__PHASE:transition point |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS021	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.4%; ACGU; EZ |TEMPERATURE:37 °C |GROWTH__PHASE:stationary phase |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS022	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.2% |TEMPERATURE:37 °C |OPTICAL__DENSITY:OD600 from 0.15 to 0.25 |GROWTH__PHASE:exponential phase |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS023	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.2% |TEMPERATURE:37 °C |OPTICAL__DENSITY:OD600 from 1.8 to 2.2 |GROWTH__PHASE:transition point |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS024	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC__BACKGROUND:WT |MEDIUM:MOPS minimal medium |MEDIUM__SUPPLEMENTS:glycerol 0.2% |TEMPERATURE:37 °C |GROWTH__PHASE:stationary phase |AGITATION__SPEED:200 rpms	Lrp	Kroner et al., 2019
BSCS025	ORGANISM:Escherichia coli str. K-12 substr. W3110 |GENETIC__BACKGROUND:WT |MEDIUM:LB medium |MEDIUM__SUPPLEMENTS:ZnSO4 500 µM |TEMPERATURE:37	ZraR	Rome et al., 2018

Dataset ID	Growth Conditions	TF	Reference
	|OPTICAL_DENSITY:OD600 of 0.6		
BSCS026	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:flhC-3xflag |MEDIUM:LB medium |AERATION:aerobic |TEMPERATURE:37 °C |OPTICAL_DENSITY:OD600 from 0.5 to 0.7	FlhDC	Fitzgerald et al., 2014
BSCS027	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:flhD-3xflag |MEDIUM:LB medium |AERATION:aerobic |TEMPERATURE:37 °C |OPTICAL_DENSITY:OD600 from 0.5 to 0.7	FlhDC	Fitzgerald et al., 2014
BSCS028	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:phoB-3xflag |MEDIUM:MOPS minimal medium |MEDIUM_SUPPLEMENTS:phosphate(3-) 0.2 mM	PhoB	Fitzgerald et al., not published
BSCS029	ORGANISM:Escherichia coli str. K-12 substr. MG1655 |GENETIC_BACKGROUND:phoB-3xflag |MEDIUM:MOPS minimal medium |MEDIUM_SUPPLEMENTS:phosphate(3-) 1.32 mM	PhoB	Fitzgerald et al., not published

Table 4. Summary of the ChIP-seq datasets, and their associated growth conditions and references.

Reference & availability

Data

The full collection of HT datasets can be consulted from the RegulonDB portal: http://regulondb.ccg.unam.mx

The RegulonDB-HT documentation is available at github: https://github.com/PGC-CCG/RegulonDB-HT

Publication

This work was published in the following article:

RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in Escherichia coli K-12 (2022)

My main personal contributions to this article include methodology and software development for the conception of the framework, the generation of standardized TSS, TU and TTS datasets, and the complete processing of the ChIP-seq collection. Additional contributions include the writing, editing and reviewing of the article manuscript, the production of three complete figures, the conception and/or formatting of three others, and the final submission.

Citation

Tierrafría, V. H#., **Rioualen, C.#**, Salgado, H., Lara, P., Gama-Castro, S., Lally, P., Gómez-Romero, L., Peña-Loredo, P., López-Almazo, A. G., Alarcón-Carranza, G., Betancourt-Figueroa, F., Alquicira-Hernández, S., Polanco-Morelos, J. E., García-Sotelo, J., Gaytan-Nuñez, E., Méndez-Cruz, C.-F., Muñiz, L. J., Bonavides-Martínez, C., Moreno-Hagelsieb, G., Galagan J. E., Wade J. T., Collado-Vides, J. (2022). RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in Escherichia coli K-12. Microbial Genomics, 8(5). https://doi.org/10.1099/mgen.0.000833

#these authors contributed equally

MICROBIAL GENOMICS

RESEARCH ARTICLE

Tierrafría *et al.*, *Microbial Genomics* 2022;8:000833

DOI 10.1099/mgen.0.000833

MICROBIOLOGY SOCIETY

OPEN DATA OPEN ACCESS

RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in *Escherichia coli* K-12

Víctor H. Tierrafría[1,2]†, Claire Rioualen[1]†, Heladia Salgado[1], Paloma Lara[1], Socorro Gama-Castro[1], Patrick Lally[2], Laura Gómez-Romero[3], Pablo Peña-Loredo[1], Andrés G. López-Almazo[1], Gabriel Alarcón-Carranza[1], Felipe Betancourt-Figueroa[1], Shirley Alquicira-Hernández[1], J. Enrique Polanco-Morelos[1], Jair García-Sotelo[4], Estefani Gaytan-Nuñez[1], Carlos-Francisco Méndez-Cruz[1], Luis J. Muñiz[1], César Bonavides-Martínez[1], Gabriel Moreno-Hagelsieb[5], James E. Galagan[2], Joseph T. Wade[6,7] and Julio Collado-Vides[1,2,8,*]

Abstract

Genomics has set the basis for a variety of methodologies that produce high-throughput datasets identifying the different players that define gene regulation, particularly regulation of transcription initiation and operon organization. These datasets are available in public repositories, such as the Gene Expression Omnibus, or ArrayExpress. However, accessing and navigating such a wealth of data is not straightforward. No resource currently exists that offers all available high and low-throughput data on transcriptional regulation in *Escherichia coli* K-12 to easily use both as whole datasets, or as individual interactions and regulatory elements. RegulonDB (https://regulondb.ccg.unam.mx) began gathering high-throughput dataset collections in 2009, starting with transcription start sites, then adding ChIP-seq and gSELEX in 2012, with up to 99 different experimental high-throughput datasets available in 2019. In this paper we present a radical upgrade to more than 2000 high-throughput datasets, processed to facilitate their comparison, introducing up-to-date collections of transcription termination sites, transcription units, as well as transcription factor binding interactions derived from ChIP-seq, ChIP-exo, gSELEX and DAP-seq experiments, besides expression profiles derived from RNA-seq experiments. For ChIP-seq experiments we offer both the data as presented by the authors, as well as data uniformly processed in-house, enhancing their comparability, as well as the traceability of the methods and reproducibility of the results. Furthermore, we have expanded the tools available for browsing and visualization across and within datasets. We include comparisons against previously existing knowledge in RegulonDB from classic experiments, a nucleotide-resolution genome viewer, and an interface that enables users to browse datasets by querying their metadata. A particular effort was made to automatically extract detailed experimental growth conditions by implementing an assisted curation strategy applying Natural language processing and machine learning. We provide summaries with the total number of interactions found in each experiment, as well as tools to identify common results among different experiments. This is a long-awaited resource to make use of such wealth of knowledge and advance our understanding of the biology of the model bacterium *E. coli* K-12.

Received 18 December 2021; Accepted 24 April 2022; Published 18 May 2022

Author affiliations: [1]Centro de Ciencias Genómicas, Universidad Nacional Autónoma de México, Avenida Universidad s/n, Cuernavaca 62210, Morelos, Mexico; [2]Department of Biomedical Engineering, Boston University, 44 Cummington Mall, Boston, MA 02215, USA; [3]Instituto Nacional de Medicina Genómica, INMEGEN, Periférico Sur 4809, Arenal Tepepan, Tlalpan 14610, CDMX, Mexico; [4]Laboratorio Internacional de Investigación sobre el Genoma Humano, Universidad Nacional Autónoma de México, Querétaro 76230, Querétaro, Mexico; [5]Department of Biology, Wilfrid Laurier University, 75 University Ave W, Waterloo, ON N2L 3C5, Canada; [6]Wadsworth Center, New York State Department of Health, Albany, NY, USA; [7]Department of Biomedical Sciences, University at Albany, SUNY, Albany, NY, USA; [8]Centre for Genomic Regulation (CRG), The Barcelona Institute of Science and Technology, Dr. Aiguader 88, Barcelona 08003, Universitat Pompeu Fabra(UPF), Barcelona, Spain.

***Correspondence:** Julio Collado-Vides, colladojulio@gmail.com

Keywords: ChIP-seq; ChIP-exo; RNA-seq; gSELEX; DAP-seq; Transcriptional Regulatory Network; High-Throughput Nucleotide Sequencing; *Escherichia coli* K-12.

Abbreviations: CRF, conditional random field; GC, growth condition; HT, high-throughput; LT, low-throughput; MCO, microbial conditions ontology; NLP, natural language processing; PWM, position weight matrix; TF, transcription factor; TFBS, transcription factor binding site; TFRS, transciption factor regulatory site; TSS, transcription start site; TTS, transcription termination site; TU, transcription unit.

†These authors contributed equally to this work

Data statement: All supporting data, code and protocols have been provided within the article or through supplementary data files. Nine supplementary tables are available with the online version of this article.

DATA SUMMARY

All the data are available on the RegulonDB portal (https://regulondb.ccg.unam.mx/). We also provide all the code and documentation associated with these new collections:

RegulonDB software project (https://github.com/regulondbunam/). Database, web services, and web interface.

RegulonDB-HT documentation (https://github.com/PGC-CCG/RegulonDB-HT). Programs used to generate uniform collections of HT objects, mapping them to low-throughput (LT) data, and a manual describing the associated processes and formats.

RegulonDB-HT dataset docker (https://doi.org/10.5281/zenodo.6376425). From Zenodo, the users can find a link to this docker container with the dataset collections in MongoDB, the web services in GraphQL, and the web interface in React.

ChIP-seq pipeline (https://github.com/PGC-CCG/SnakeChunks). A library based on the snakemake workflow management system, which was used to design a generalizable workflow to perform reproducible ChIP-seq analyses [1].

EcoliGenes library (https://github.com/PGC-CCG/EcoliGenes). This R-based library was developed to efficiently deal with frequent and all too-often fastidious tasks related to the programmatic manipulation and comparison of genes and TFs. This library was used in multiple scripts and pipelines mentioned in this article to identify the wide variety of names and IDs used to report genes and TFs in databases and literature, the existence of multiple synonyms, spellings, and outdated bnumbers, and to convert them all into the most up-to-date symbols and bnumbers. It also includes a variety of functions that allow to efficiently get additional information on genes (coordinates, length, product, etc.) or specific genome coordinates (type of region, closest gene) directly into R data.frames, and to convert genomic coordinates from *E. coli* K-12 genome version NC_000913.2 to NC_000913.3.

The authors confirm all supporting data, code, and protocols have been provided within the article or through supplementary data files.

INTRODUCTION

Genomics has enabled a variety of technologies for the genome-wide identification of different elements defining transcription initiation, gene regulation, and transcription unit organization in any organism, provided its genome has been sequenced. In bacteria, these elements include TFs, TF binding sites (TFBS) that show specific binding of TFs, out of which we distinguish TF regulatory sites (TFRS; defined as TFBSs that are involved in transcription regulation) [2]. Moreover, genes can be transcribed either individually, or in polycistronic units, defining transcription units (TUs), which are delimited by transcription start sites (TSSs) and transcription termination sites (TTSs). As reported recently, with the development of technologies and the extension of our knowledge of transcriptional regulation, several classic definitions had to be extended. For instance, both promoters and terminators can have multiple TSSs and TTSs, respectively [2]. These updated definitions have been timely incorporated in RegulonDB [3] and in EcoCyc [4], another major resource containing information on transcriptional regulation of *E. coli* K-12.

Genome-scale technologies allow for the identification of several types of elements, such as TFBSs, gene expression profiles, and genomic elements including TUs, promoters and terminators. Approaches for TFBS identification include *in vivo* chromatin immunoprecipitation sequencing (ChIP-seq) [5, 6], its higher-resolution variant ChIP-exo [7], in addition to *in vitro* approaches, such as biotin-DNA affinity purification sequencing (DAP-seq) [8] and genomic systematic evolution of ligands by exponential enrichment (gSELEX) [9]. Note that given the binding evidence, it is not certain that proteins considered as TFs in these HT binding experiments are *bona fide* TFs, since many of them lack evidence of change in gene expression. Gene expression profiles are obtained using RNA-seq. Higher-resolution variants of RNA-seq, protecting the 5′-end of transcripts, allow for TSS identification at single-nucleotide resolution [10–12], and more recently, for the determination of full-length transcripts, along with their TSSs and TTSs [13, 14].

Publications reporting these experiments frequently describe a subset of regulatory objects, either spread along the main text [15] or compiled in tables [16–19]. Authors also provide processed datasets as supplementary material [20, 21], whereas the raw data are deposited in public repositories, such as NCBI's Gene Expression Omnibus (https://www.ncbi.nlm.nih.gov/gds), the Sequence Read Archive (https://www.ncbi.nlm.nih.gov/sra), and the EMBL-EBI's ArrayExpress (https://www.ebi.ac.uk/arrayexpress/). Extracting and processing such datasets can be challenging. Gathering these types of data in a single resource, such as RegulonDB, saves a lot of work and accelerates research facilitating data comparison with the accumulated existing knowledge based on classic molecular biology experiments, as well as comparisons with future novel knowledge.

E. coli K-12 is the prokaryote with the largest number of regulatory systems studied by classic experimental methods of molecular biology. Our laboratory at UNAM has gathered this rich, classic low-throughput (LT) knowledge for more than two decades, feeding both RegulonDB and EcoCyc [3, 4]. With the publication of large collections resulting from HT sequencing methods, we were concerned by the potential dilution of the LT classic corpus, historically considered as the gold standard, with larger collections identified by novel approaches that involve a large number of processing steps in the final identification of regulatory objects. We therefore considered offering users HT results as separate collections, the way we were offering a few genome-wide

Impact Statement

RegulonDB has been the main resource for knowledge about transcriptional regulation and organization in *E. coli* K-12, and has been accessed intensively since its first publication in 1998 [52]. For instance, in the last 4 years, RegulonDB was accessed an average of ~16300 times per year, and citations to RegulonDB articles quickly count in the hundreds. This curated database started more than 20 years ago, before the advent of high-throughput (HT) experimentation, gathering data obtained by traditional methods, with some HT data added later on. Here we present a major undertaking in ensuring high coverage of the latest HT experimental data in RegulonDB, by incorporating more than 500 HT datasets for transcription factor (TF) DNA-binding, in addition to 1864 RNA-seq datasets generated under different growth conditions and/or genetic backgrounds. Another novelty in this BioResource is the curation effort to associate each dataset with its corresponding detailed metadata that is key for its utilization. The value of having the derived genomic features, or objects, from different kinds of experiments, available in a single repository, will add to the already acknowledged value of RegulonDB to the scientific community.

datasets of TSSs generated by our collaborators back in 2009 [22]. We thus gathered datasets of TFBSs obtained by ChIP-seq and gSELEX in versions 8.0 [10] and 9.0 of RegulonDB [23]. Detailed manual curation has been devoted to extract TFRSs from those publications, for which additional evidence showing a change in expression of a nearby target gene [24] supports a regulatory interaction. Those have been uploaded into EcoCyc and RegulonDB with a clear HT evidence type along with those identified by classic LT methods. In addition to COLOMBOS with expression data [25], the Transcription Profile of *Escherichia coli* (TEC) database [26], released in 2016, offers gSELEX data in *E. coli* and the PROkaryotic Chromatin ImmunoPrecipitation database (proChIPdb, [27]), recently released, offers ChIP-seq and ChIP-exo datasets. However, to our knowledge, there is no comprehensive resource facilitating access in a single place to the diverse wealth of data of different types of objects relevant to the regulation of gene expression in *E. coli* K-12.

In this article we present a radical upgrade of RegulonDB, offering up-to-date collections of TFBSs identified from ChIP-seq, ChIP-exo, gSELEX, and biotin-modified DAP-seq approaches, as well as TSSs, TTSs, TUs and a large collection of RNA-seq expression profiles. For most of them we offer the data published by the authors, extracted either from publications or from dedicated databases. We also processed some collections from available raw data using uniform pipelines reducing their methodological differences or batch effects.

Knowing the biological conditions and genetic background supporting a binding site, an expression profile, the mapping of transcription initiation, or a transcription unit, is crucial to compare them and locate them in the wider context of additional knowledge. We used the Microbial Conditions Ontology (MCO) [28] as our theoretical framework to organize this knowledge, and, as explained below, we also implemented an assisted curation strategy applying Natural language processing (NLP) and machine learning (jointly named: *NLP method*) to automatically extract this knowledge. This assisted curation strategy consists in curating the automatically extracted growth conditions instead of curating conditions from the sources of this knowledge, saving human effort. Additionally, we added search capabilities, besides reorganizing displays in a way that should considerably improve the browsing and visualization of the different datasets and collections.

METHODS

RegulonDB-HT data model and definitions

In this work, we offer facilitated access to HT *collections*. Each collection comprises the curated *datasets* resulting in a specific type of object (Fig. 1); and a *metadata* table containing the complete list of *datasets* and their curated properties. The specific collection of TF binding objects has several *subcollections* based on the type of technology. We conceive a *dataset* as a set of data from a given experiment and its growth conditions as detailed in the MCO (culture medium, medium supplements, aeration, temperature, pH, agitation, growth phase, optical density, genetic background). *Metadata* tables also include additional information such as the genome version, features associated with the publications (author list, year of publication, PMID), as well as reported database identifiers, and any additional pertinent information. Datasets contain data files provided in the original publications (referred to as 'author files'), data files with results from our in-house processing pipelines (referred to as 'uniformized files'), or both types of files.

A new repository was designed to store the different types of datasets. The classes representing the organization of information within RegulonDB-HT, and the types of datasets processed, include TFbindingPeak, TFbindingSite, TranscriptionUnit, TranscriptionStartSite, TranscriptionTerminationSite and GeneExpression. Each of these are accompanied by their metadata and growth conditions, and at least one author data file or uniformized data file (Fig. 1). Growth conditions in the GeneExpression collection were obtained using the NLP method explained below.

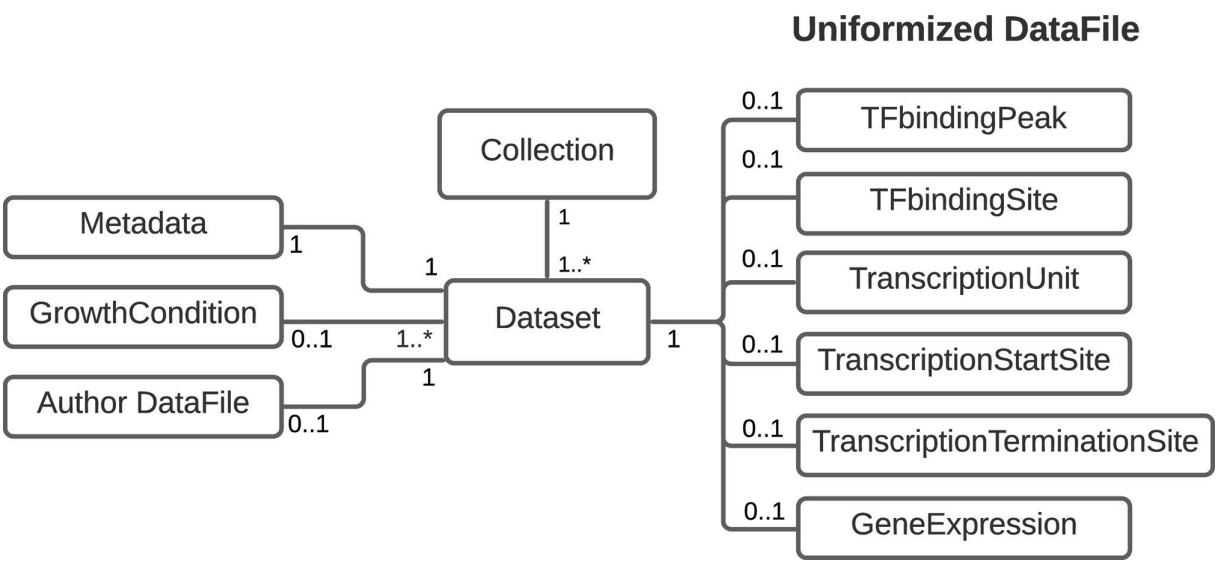


Fig. 1. Data model for HT dataset collections represented as a Unified Modelling Language (UML) class diagram. The links represent bidirectional associations between two classes, and the numbers 1, 0.*, 1.* represent the multiplicity value. For example, the class *Dataset* can have 0 or 1 *Author DataFile*. The components of datasets are the *Metadata*, defined as properties in the *Dataset* class, the *Growth Conditions*, curated manually or using the NLP method, and related data files, either gathered from authors or processed for uniformity.

The data repository was implemented in MongoDB v4.4.5 (https://www.mongodb.com/), a document-oriented database manager that provides the flexibility to deal with the variety of information of each type of dataset and collection. The package for processing the authors' and uniformized data files, and to extract, transform, and load data, was developed under python 3.9. The ChIP-seq workflows were implemented in snakemake 6.10.0 [29]. Access to data was implemented through web services that use Node v16.13.0 (https://nodejs.org/es/), the query language GraphQL v15.5.0 (https://graphql.org/), and Apollo Server Express v2.21.0. A component-based web interface was developed using React v17.0.2 (https://es.reactjs.org/). The tracks display uses igv.js, an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV) [30]. The software and applications related to the database are available at GitHub (https://github.com/regulondbunam/).

Gathering and processing of the HT data collections

To implement this new framework, we carefully coordinated the different steps involved: manual curation and annotation of literature, data uniformization, computational mapping and display of the HT collections (Fig. 2).

Data gathering

Original scientific papers about transcriptional regulation in *E. coli* K-12 are monthly searched in PubMed (https://pubmed.ncbi.nlm.nih.gov/). Then, articles are selected and curated as described previously [3]. For this work, databases associated with the publications were also explored, these include: Gene Expression Omnibus (GEO https://www.ncbi.nlm.nih.gov/gds) and the Sequence Read Archive (SRA https://www.ncbi.nlm.nih.gov/sra) from the NCBI, ArrayExpress (https://www.ebi.ac.uk/arrayexpress/) from EMBL-EBI, Digital Expression Explorer 2 (DEE2 http://dee2.io/), proChIPdb (https://prochipdb.org/), and TEC (https://shigen.nig.ac.jp/ecoli/tec/top/).

Curation and annotation

The information provided within the original publications was carefully collected and organized into custom metadata tables (one per collection, or one per subcollection in the case of TF-binding), with metadata and growth conditions for each dataset. The datasets constructed from authors sources were annotated and organized into the RegulonDB-HT repository.

Normalization and uniformization

To facilitate processing, display and analysis of these datasets, several strategies were used to uniformize and/or normalize certain datasets.

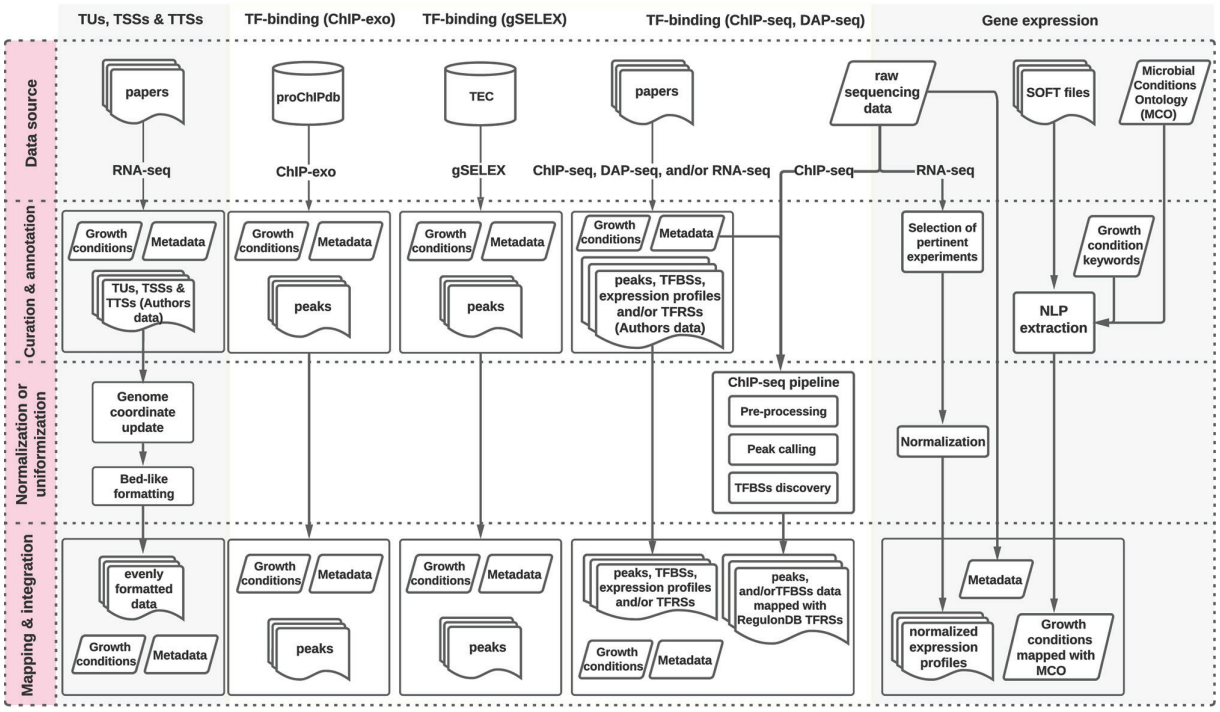


Fig. 2. Overview of the RegulonDB HT framework. This diagram summarizes the three types of dataset collections built in RegulonDB HT: i) genomic features (TUs, TSSs, and TTSs), ii) TF binding and iii) gene expression, displayed as grayscale background columns; and the steps implemented to generate them: i) data gathering, ii) curation, iii) normalization and iv) integration, displayed as horizontal lanes. Further details are described in Methods sections regarding datasets.

Mapping and integration

The resulting uniform HT objects were mapped to reference datasets from LT experiments as curated in RegulonDB. As already mentioned, growth conditions were mapped to the MCO terms and annotated, when available, according to the annotation framework reported in [28].

TF binding datasets

Data gathering

We are including binding data from four HT technologies: ChIP-seq, ChIP-exo, gSELEX and DAP-seq. The ChIP-seq datasets encompass two types of data contained in two different tables: data as reported by the authors, and data generated from our in-house processing of the raw data reported by the same authors. The TFBSs and/or peaks reported by authors were obtained mostly from supplementary material and the associated information described in the main text of their publications. ChIP-seq raw samples and metadata were downloaded systematically from the SRA. The ChIP-exo subcollection was retrieved from the recently published proChIPdb [27]. This subcollection includes datasets tagged in proChIPdb as 'curated', as well as TF binding information for OxyR, SoxR, SoxS, and UvrY, from [31, 32].

The gSELEX datasets were extracted from the TEC database [26]. Each TF was searched in the Tab 'Gene/TF search' with a selected cut-off (indicated in the metadata). The data were obtained by copy and paste since it was not possible to download it otherwise. The datasets contain the TF name, peak center coordinates, target gene, peak location relative to the target, and binding intensity (%) relative to the highest peak intensity in the experiment. We built 63 datasets for 41 TFs using a defined threshold either indicated in the corresponding references, or, in their absence, inferred by us to include all targets indicated in the publications. Ninety-four gSELEX datasets (corresponding to 74 different TFs) were not analysed by the authors, they were only listed in one publication [26]; for these we took the forty targets with the top binding intensities, and the lowest binding intensity was registered in the metadata as the cut-off for each dataset. To allow comparisons with data derived from other methodologies, we offer complete datasets from gSELEX, i.e. with no cut-off, for the nucleoid-associated proteins H-NS, Fis, IHF and HU, as well as Dps and Dan which have also been proposed as nucleoid-associated proteins [26, 33–36]. Overall, a total of 164 TFBS datasets (corresponding to 121 different TFs) derived from gSELEX were generated.

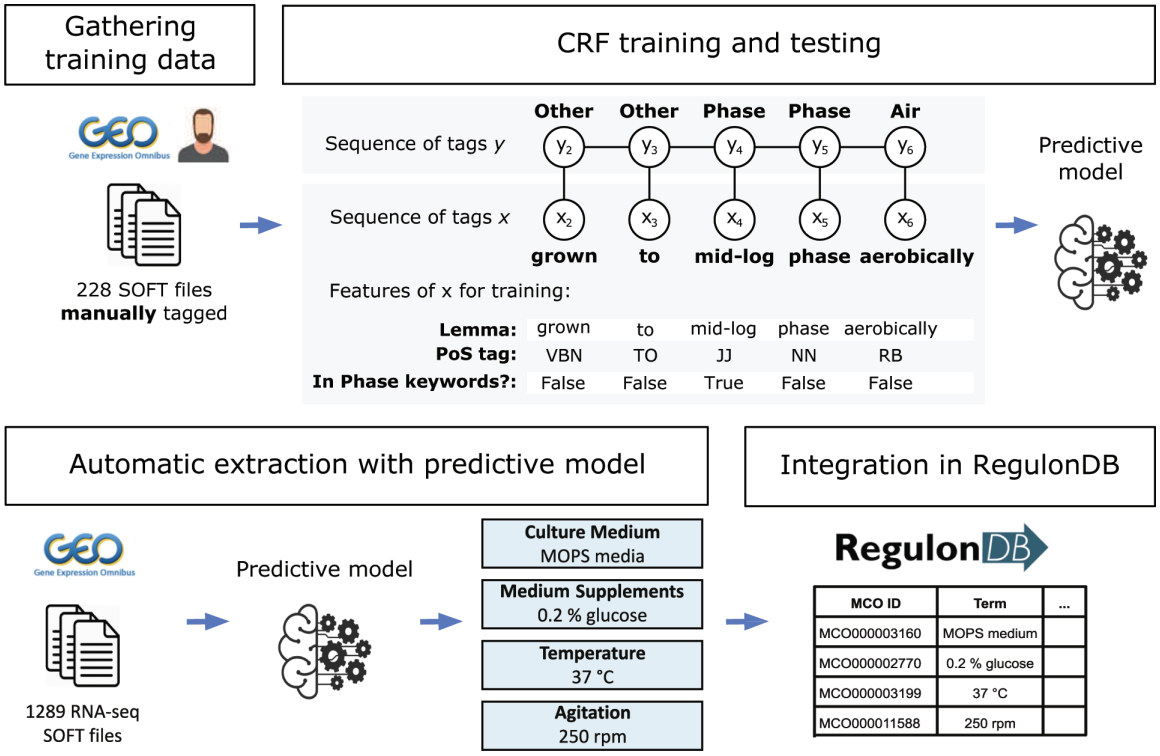


Fig. 3. Steps for growth conditions extraction using our NLP method.

Finally, we obtained the collection of experiments and metadata for 215 TFs in *E. coli* using biotin DAP-seq from the supplementary material available in [37].

Curation and annotation

To build the dataset component with data as reported by authors (Fig. 2, Curation and Annotation lane), we retrieved the following features when available: TF name, peak and TFBS features, such as start- and end genomic coordinates, genomic sequence, statistical values from peak calling or motif prediction, experimental or computational evidence, and the closest gene, considered as the target gene. The associated metadata, including growth conditions, were also extracted from the publications and databases mentioned above. Finally, when ChIP-seq experiments were linked to gene expression in the same publication, we flagged target genes which showed changes in expression and a significant *p-value* for differential expression, annotating the resulting TF function as either activator or repressor. These TFRSs support regulatory interactions which are in the process of being uploaded into EcoCyc and RegulonDB.

Uniformization

We gathered a total of 185 raw data files from 28 ChIP-seq datasets associated with 11 TFs. We processed them in a uniform and reproducible way using the SnakeChunks library of workflows for HT analysis [1, 29]. This framework ensures the consistency of analyses, keeps track of the tools and versions used, while also allowing parameter customization. Adapter and quality trimming were performed using cutadapt with a quality and length threshold of 20 [38]. Read alignment was performed using Bowtie 2 [39] in local alignment mode against the *E. coli* K-12 MG1655 genome (version NC_000913.3). Overall sample quality was checked using FastQC [40] (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) and multiQC [41]. Peak calling was performed using the latest version of Macs 3 [42], with a *q-val* threshold of 1.10–3 and the following options: *--nomodel --shift 0 --extsize 200*. Then, TFBSs were identified from the peak sequences via pattern-matching using RSAT matrix-scan [43] and the reference TF motifs built from RegulonDB 10.5 [3], and motif-specific thresholds defined by RSAT matrix-quality [44]. Two exceptions were made with GlaR and Nac, where a putative binding motif was obtained through *de novo* motif search using RSAT peak-motifs with a significance threshold of 0 [45], in order to detect binding sites. A new motif was generated for each individual dataset, using TFBS sequences and the RSAT tool convert-matrix [46]. For the other types of binding datasets, we retrieved the data as reported by the authors, in particular: start- and end positions, intensity, and the closest gene to each peak.

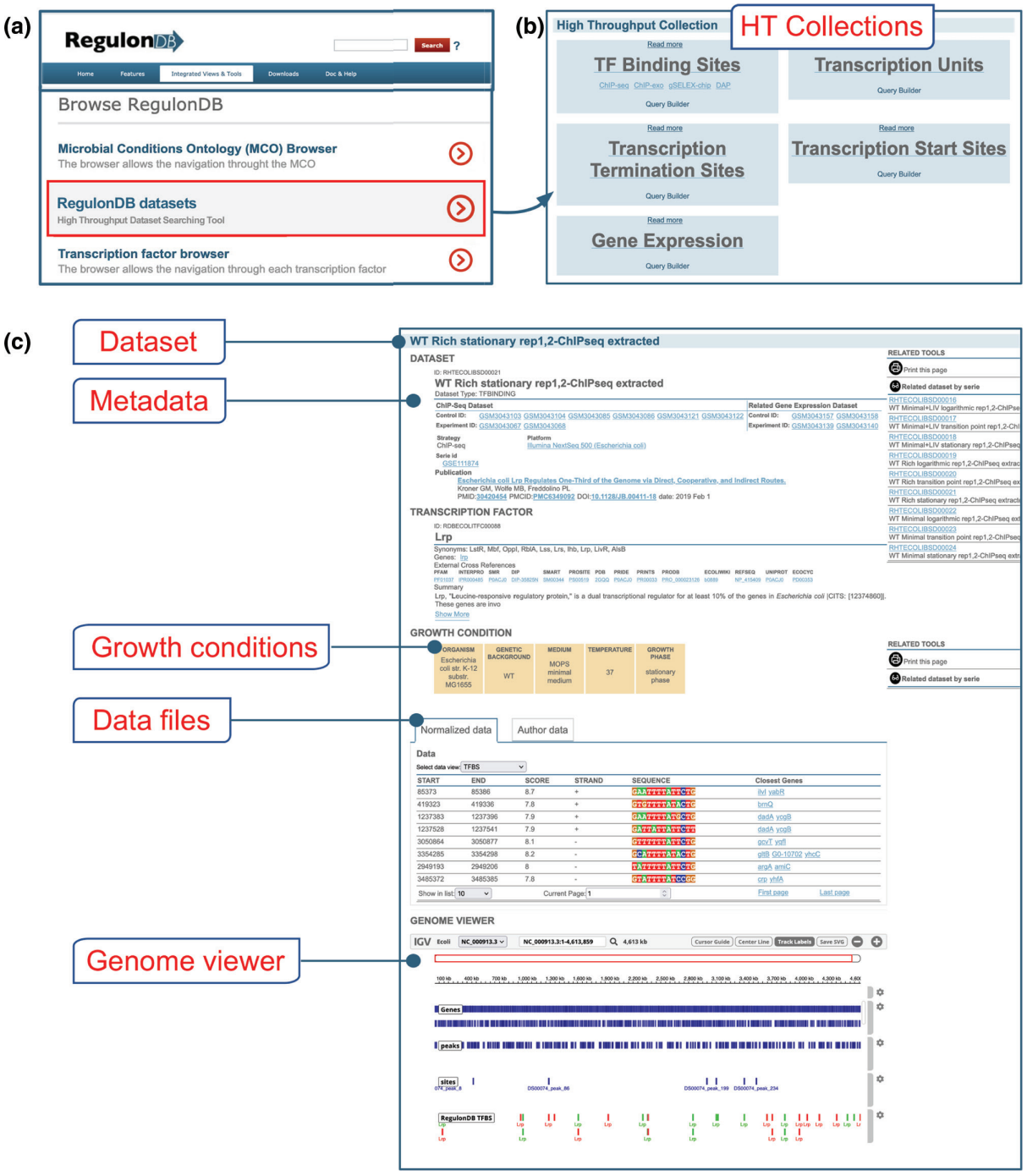


Fig. 4. RegulonDB-HT search tool. This tool gives access to all types of HT datasets retrieved so far, but an example of access to a TF binding HT dataset is shown. (a) RegulonDB portal. (b) RegulonDB HT collections. (c) Content of a TF binding dataset, from the ChIP-seq subcollection.

Mapping and integration

With the aim of comparing the TF binding data derived from HT technologies with the knowledge derived from LT studies, we performed the mapping of TFBS datasets to the RegulonDB subset of TFRSs with classical evidence. We mapped our in-house processed ChIP-seq datasets at the level of peaks and sites: a peak is considered a match when a known binding site falls within its coordinates, and a site matches when its centre position is at most 30 bp away from a known site (in average, motifs are 20 bp long, and a 10-bp distance may be close enough for protein interaction). Mapping datasets from authors proved to be more difficult since not all of them were generated using the same version of the genome, and the precise location of peaks or motifs is

Table 1. Number and content of RegulonDB HT datasets

Object	Strategy	No. of datasets	No. of objects		Additional information
			Curated from papers	Identified from raw data	
EXPRESSION PROFILES					
Gene expression	RNA-seq	1864[a]	ND	4618[b]	
TF BINDING					
TF Binding	ChIP-seq	29[c]	6585 peaks	13167 peaks 5108 sites	Table S2
	ChIP-exo	94	23170 peaks	ND	Table S3
	gSELEX	164	35022 peaks	ND	Table S4
	DAP-seq	215	19540 peaks	ND	Table S5
TUs, TSSs and TTSs					
TUs	RNA-seq	5	12347[d]	ND	Table S6
TSSs	RNA-seq	16	68049[d]	ND	Table S7
TTSs	RNA-seq	5	5326[d]	ND	Table S8

a, The total of SRRs retrieved, which include 575 only in DEE2, 914 (820 GSMs) only in GEO, and 375 (337 GSMs) in both DEE2 and GEO
b, Average number of genes per dataset.
c, Including 27 processed by authors and 28 processed in house.
d, The number of these objects may be higher from the original publications as they were calculated per *dataset*, after our uniformization process. ND. Object identification not determined by the RegulonDB Team.

not always available in publications. Thus, the datasets processed by authors were mapped at the level of the TF-gene interactions. For each TF binding dataset, target genes were compared against the known regulatory interactions from RegulonDB, taking into account the evidence they are associated with (Table S1). Positive mapping results display the type of evidence (classical strong or weak, or computational prediction) of the corresponding interaction in RegulonDB.

TU, TSS, and TTS datasets

Data gathering

Datasets of TUs, TSSs and TTSs came from different sources, though their growth conditions were not always consistently documented. TSS datasets generated by the group of Enrique Morett [10, 22], as well as those from the laboratory of Gisela Storz [47], were already available in RegulonDB [23]. Four collections are from Cho, B. K., *et al.* [48], with additional collections obtained from publications that implemented the identification of TUs using different approaches, which concomitantly identified TSSs and TTSs as TU boundaries [11, 13, 14]. A dataset not-yet-published of more than 5000 TSSs was kindly provided by Joseph T. Wade.

Curation and annotation

Given that transcriptional regulation involves a machinery that deals with different growth conditions, we gathered the precise growing conditions under which these different elements were identified, directly requesting authors for the information when it was not detailed in the publications. Key growth conditions obtained through personal communication include culture medium, either minimal or rich, and growth phase, either exponential or stationary phase.

Uniformization

When necessary, we updated object coordinates to the current genome version NC_000913.3. While the original datasets came in a variety of formats, we extracted the most relevant features for each type of collection, and generated uniform bed files for each dataset to allow their visualization in our genome browser. Objects that shared the same start, stop and strand information were considered duplicates and merged as single objects. Finally, when objects provided in a single file by the author were associated with distinct growth conditions, they were separated in distinct datasets (see *dataset* definition in the Methods section).

Mapping and integration

The uniform TSS datasets were mapped against RegulonDB promoters, and were considered a match when they fell within a 5-bp distance of a known TSS. TUs and TTSs will also be mapped in the near future. Those three uniformized collections were

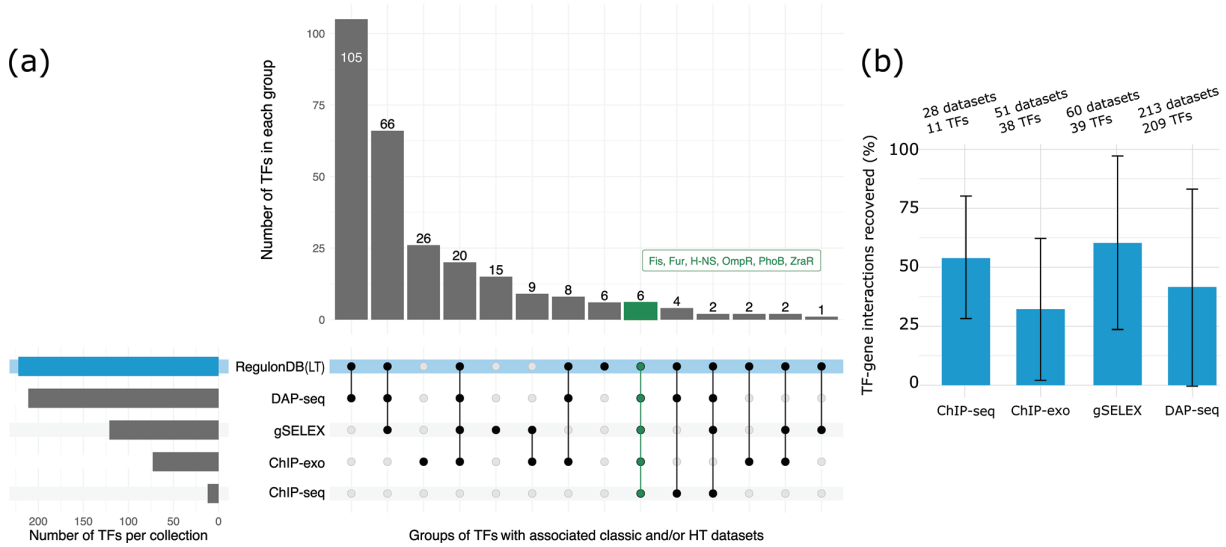


Fig. 5. TFs with binding identified by ChIP-exo, ChIP-seq, DAP-seq and/or gSELEX. (a) Comparison of TFs studied with LT approaches available in RegulonDB, with TFs examined with HT technologies. In RegulonDB, 222 TFs have been confirmed by classical LT evidence with at least one regulatory interaction (displayed as a horizontal blue bar). Each vertical bar represents a group of TFs associated with LT and/or HT experiments, as displayed by the black dots in the bottom rows. (b) Average percentage of TF-gene interactions with classical evidence in RegulonDB, identified in data processed by authors.

integrated into our genome viewer. The original author datasets were not mapped nor integrated into the genome viewer, since they come in a variety of formats and genome versions.

Gene expression datasets

Data gathering

We collected RNA-seq experiments from two different sources, GEO and DEE2. A total of 1429 experiments were retrieved from GEO using 'RNA-seq' and *E. coli*'s taxon id (txid562) as a query. We also obtained 1255 experiments from DEE2 that were not found in our initial GEO query.

Curation and annotation

We filtered these datasets based on the type of experiment and sequencing format used, retaining only RNA-seq experiments, and removing those performed with SOLiD sequencing, as our pipeline is tailored towards Illumina. We also filtered out the datasets that were associated with strains other than K-12. Of the 2684 total samples, we uploaded into RegulonDB the 1864 that could be processed by our pipeline (see Normalization subsection below). This collection is up-to-date as of the end of October 2021. The metadata were also retrieved from the corresponding database. We used the NLP method to extract growth conditions from the metadata files provided by the authors to complement the datasets obtained from GEO. For experiments only found in the SRA (retrieved from DEE2), we used NCBI's Entrez tool, along with custom software, to gather the metadata. In particular, when the metadata were missing or scarce, we used the python package Beautiful Soup four to perform web-scraping.

To gather training data for our NLP method, we selected GEO SOFT files containing metadata of studies performed with different technologies such as RNA-seq, ChIP-seq, and ChIP-exo, available in previous versions of RegulonDB. In total, the SOFT files of 228 GEO samples from 27 GEO series were gathered (Fig. 3). We automatized SOFT files download using the R package GEOquery. We manually curated and tagged the following features describing growth conditions: organism, genetic background, culture medium, medium supplements, growth phase, OD, temperature, pH, aeration, agitation, and genome version.

Manually tagged contexts from 228 SOFT files were used to train and test a linear chain Conditional Random Field (CRF): 70% for training and cross-validation, and 30% for testing. In addition, we manually obtained lists of keywords related to some types of growth conditions. A CRF is a probabilistic framework for tagging and segmenting sequence data based on the conditional probability $P(y|x)$ of a sequence of tags $y = y_1 \ldots y_n$ given a sequence of observations $x = x_1 \ldots x_n$ [49]. In this case, x is the sequence of words of contexts from the SOFT files, and y is the sequence either of tagged growth conditions ('Air', 'Phase', etc.), or the label 'Other' in other cases. The CRF probabilities are based on feature functions which may consider any feature of x_i (e.g. the part-of-speech tag, the lemma, if it contains the symbol '°', if it appears in a list of keywords) and the transition $y_{i-1} \rightarrow y_i$ (e.g. 'Phase' before 'Air'). For the final output, the consecutive words with the same label were collapsed

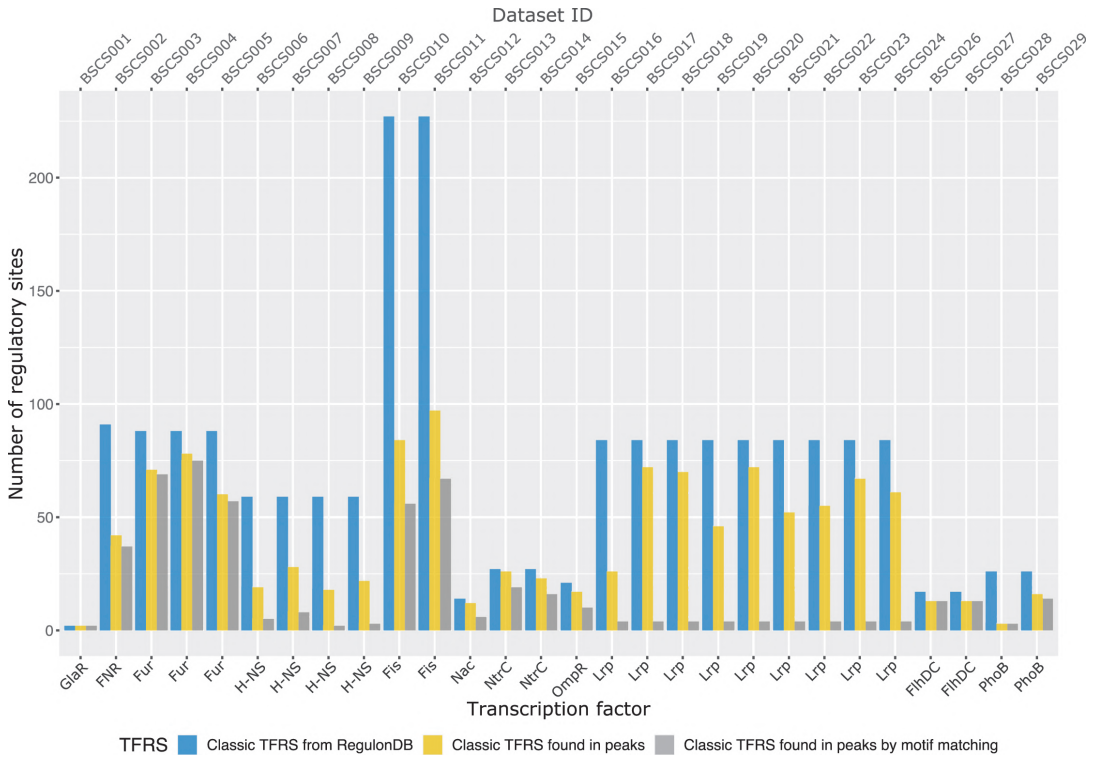


Fig. 6. Number of TFRSs from classic RegulonDB (blue bars), those found in in-house processed ChIP-seq peaks (yellow bars), and those identified in peaks through pattern-matching, using RegulonDB TF motifs (grey bars).

into a fragment of text, while the probabilities were summarized as the mean. This approach has been successfully applied previously for information extraction and it does not require a lot of training data [50].

Normalization

We downloaded the fastq files from the SRA for all datasets to be homogeneously processed by our sequence analysis pipeline. We aligned all samples to the *E. coli* reference genome NC_000913.3 using HISAT2. Our alignments are always run as unpaired; and when the metadata allow determination of the library preparation kit used, we provide the appropriate strandedness parameters, which indicate whether reads are to be expected on the same, or opposite strand of the mRNA transcript. We performed DEseq-normalization to facilitate comparisons across different datasets. Shortly, we created a 'pseudo-reference' sample, where we obtained the geometric mean of each gene's expression, measured in counts, FPKM/RPKM (depending if the experiment is paired-end or single-end, respectively), and TPM. Each gene in a given sample was divided by its pseudo-reference value, and a scaling factor for each sample was obtained by taking the median of these values. The final DEseq-normalized values were obtained by dividing each sample's expression by the sample scaling factor. In total, 1864 samples were processed without errors by our pipeline.

Mapping and integration

We took two approaches for mapping the automatically extracted growth conditions to MCO identifiers comparing the extracted term with the MCO term: (i) exact term matching and (ii) string similarity. String similarity was implemented using the python library fuzzywuzzy v0.18.0 (https://pypi.org/project/fuzzywuzzy/) taking into account string length differences calculated as Levenshtein distances, i.e. the minimum number of edits of one character (insertions, deletions or substitutions) required to change one word into the other. String similarity allowed us to match, for example, the extracted term 'W2 minimal medium' with the MCO term 'W2 minimal media' (ID: MCO000003317).

RESULTS

General overview of HT datasets and objects

As mentioned above, we report several collections of HT datasets that hold distinct types of objects (genomic features, TF binding sites, gene expression profiles) from distinct types of HT experiments (RNA-seq, ChIP-seq, gSELEX, DAP-seq, ChIP-exo). Some

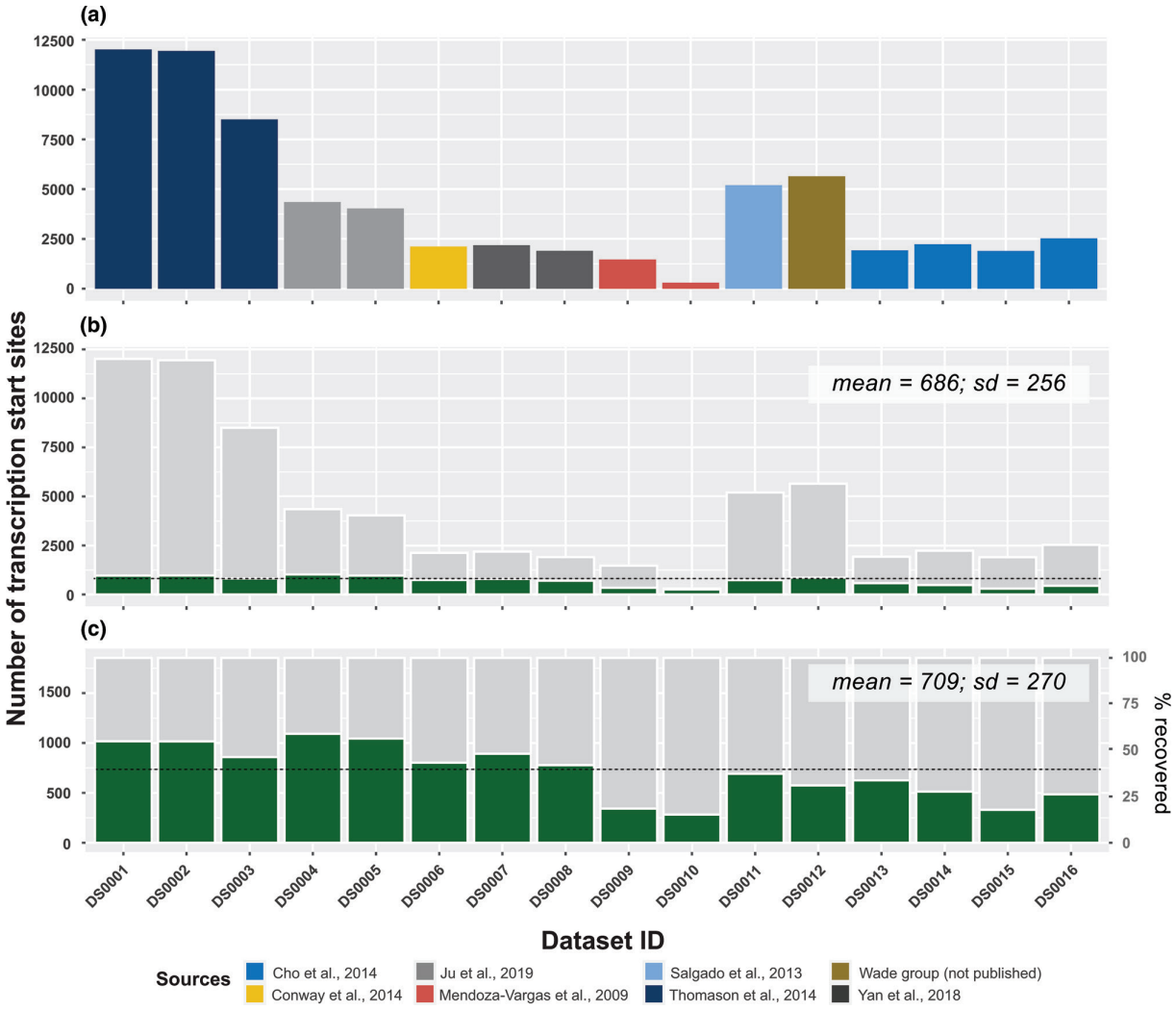


Fig. 7. High-throughput TSS datasets collected and mapped to RegulonDB classic TSSs. (a) Number of TSSs per HT dataset. (b) Number of HT TSSs that match with at least one classic TSS. (c) Number of classic TSSs that match with at least one HT TSS, for each HT dataset.

collections contain two dataset tables: data as reported by the authors, and data uniformized and/or normalized in-house. Data as reported by authors were obtained from publications curated by us, or from the authors' databases, such as TEC and proChIPdb, generated by the Ishihama and Palsson groups respectively (Fig. 2, lane 1). Data processed by other authors frequently vary in reference genome used and/or format, so we processed the author datasets to map TUs, TSSs, and TTSs with the latest reference genome and to display them in the same format (Fig. 2, lane 3). Finally, we integrated (i) data files, (ii) metadata, and (iii) growth conditions to build the RegulonDB HT datasets (Fig. 2, lane 4 and Fig. 4).

We generated three classes of RegulonDB HT datasets, roughly grouped by type of objects (described in more detail in the following sections). For example, *gene expression* datasets comprise the largest collection of datasets and objects, as expected, but are associated with only one object type and strategy, i.e. RNA-seq. In contrast, *TF binding* datasets were produced using several strategies, i.e., ChIP-seq, ChIP-exo, gSELEX, and DAP-seq. Lastly, TU, TSS, and TTS datasets include different objects identified using variations of one strategy, i.e. RNA-seq (Table 1).

Browsing the data

All the curated and annotated information, as well as the standardized data, can be found in the RegulonDB portal (https://regulondb.ccg.unam.mx/). From the menu 'Integrated Views and Tools', in the 'Browse RegulonDB' section, the option 'RegulonDB-HT datasets' is available (Fig. 4a).

Table 2. F1-score in testing for types of growth condition

Growth condition	Precision	Recall	F1-score	Support*
Optical density	1.00	1.00	1.00	21
pH	1.00	1.00	1.00	10
Technique	1.00	1.00	1.00	33
Culture medium	1.00	0.80	0.89	56
Temperature	0.86	0.80	0.83	15
Agitation	1.00	0.29	0.44	7
Growth phase	0.94	0.76	0.84	21
Aeration	0.63	0.59	0.61	88
Genetic background	0.89	0.86	0.88	78
Medium supplements	0.88	0.84	0.86	136
Genome version	1	0.5	0.667	6

**Support* stands for the number of growth conditions available in testing data for evaluation.

An initial page allows the user to select from all types of RegulonDB collections (Fig. 4b). The search builder, which is the subsequently displayed page, allows users to choose search filters associated with the RegulonDB collections' metadata. Any dataset that meets the search criteria will be displayed in a list ordered according to the number of terms found in it. The user will be able to select the desired RegulonDB dataset by clicking its link in the results list. The content of the selected dataset looks as shown in Fig. 4c), and is composed of three main components: (i) metadata, (ii) growth conditions, and iii) related data files. In the Data Files section, users can navigate through two tabs, one to access data as reported by authors, and the other one to access the standardized data produced by the RegulonDB Team.

When uniformized data are available, it is possible to visualize them in the IGV Tool, where the genes, peaks, TFBSs found in peaks, and TFRSs of the TF already stored in RegulonDB (RegulonDB TFRSs) are displayed as tracks. In the RegulonDB TFRSs track, the colour of sites is associated with the function of the TF in line with the current EcoCyc and RegulonDB TFRSs colour code, i.e. green for activators and red for repressors (Fig. 4c).

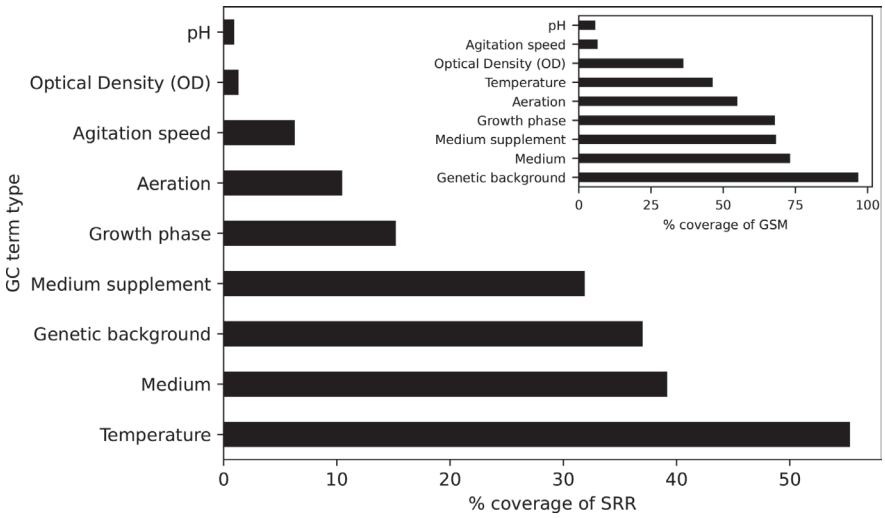


Fig. 8. Foreground bar plot: fraction of SRRs for each type of growth condition. GC term types retrieved for RNA-seq datasets from GEO (1289 SRRs, 1157 GSMs, 95 GSEs), 3224 extracted GC terms: 2680 were mapped and 544 non-mapped with MCO entities. Background bar plot: fraction of GSMs for each type of growth condition in the training data (228 GSMs from 27 GSEs).

HT data content details

In every uniformized RegulonDB HT dataset (Table 1 and Fig. 2, bottom lane), we provide the precise genomic coordinates of objects together with additionally processed information, such as the closest downstream gene(s) in the case of TFBSs and TSSs, and the gene content, in the case of TUs. Another column indicates the list of objects that match previously known objects identified by LT methods as indicated in the evidence type in RegulonDB. This pre-processed column should be highly valuable for users performing comparative analyses. In a future version we will pre-process the comparisons across the multiple HT collections.

In datasets with information provided by the authors, the confidence may vary. For instance, of the TUs identified by Yan, B. *et al.* using SMRT-Cappable-seq, some have a well-identified terminator either by sequence structure or because a significant fraction of transcripts that start at a given TSS terminate at a well defined TTS position. These TUs have a higher confidence level than the other TUs, defined by the end of one or very few long transcripts [13]. In the case of TFBSs, users can identify sites matching previously known sites stored in RegulonDB LT and/or additional evidence supporting change in expression of downstream genes.

TF binding datasets

The ChIP-seq subcollection is conformed by 29 datasets corresponding to 12 different TFs, of which 28 were processed using our dedicated pipeline (one dataset does not come with raw data), and 27 are associated with author files (two datasets are not associated with a publication). Overall, besides those exceptions, 26 datasets associated with ten TFs are provided with two tables: one with data processed by authors and built from the publications, and one with data uniformly processed in-house from raw data (Table S2).

The ChIP-exo subcollection consists of 94 datasets built with data processed by authors, which include 87 datasets corresponding to 73 different TFs assayed independently, and seven datasets derived from assays of a mixture of various TFs (Table S3).

The gSELEX subcollection consists of 164 datasets built with data processed by authors and extracted from the TEC database, corresponding to the binding of 121 different TFs assayed *in vitro* in presence or absence of effector molecules (Table S4). However, as mentioned in methods, this is a heterogeneous collection given the limitations in their extraction: 63 datasets for 41 TFs had thresholds defined by the authors, for 94 datasets of 74 TFs we arbitrarily included the top 40 sites, and for seven datasets we included all interactions with no threshold (see Methods section).

Finally, the DAP-seq subcollection comprises 215 datasets of data processed by authors and built from the supplementary material of a single publication [37], which corresponds to the binding of 211 different TFs assayed *in vitro*. Some datasets correspond to the same TF because their different subunits were assayed independently (Table S5). Some TFs have been studied by more than one of these methodologies. For example, H-NS, Fur, Fis, OmpR, ZraR and PhoB are represented in all four subcollections. Moreover, some TFs without classical evidence of regulatory interactions have been studied exclusively by one of these four HT strategies, this is the case for 26 and 15 TFs from ChIP-exo and gSELEX, respectively. Six TFs with at least one regulatory interaction with classical evidence have no data in any HT binding dataset. Fig. 5a shows the total number of TFs present in the different subcollections and their comparison with classic data from RegulonDB.

We estimated the proportion of TF-gene classic interactions present in RegulonDB that were recovered in the datasets that we constructed from author data. This percentage for every dataset is shown in Tables S2–S5 (available in the online version of this article), Fig. 5b displays the average of such percentages for all datasets within each methodology. However, these numbers have to be taken with a grain of salt, first because the TFs shared by the different methodologies are quite variable, as shown in Fig. 5a, second the recovery is quite variable for different datasets provoking a large standard deviation. Furthermore, this was done only for 63 datasets from 41 TFs of the gSELEX collection since only those have a cut-off defined by the authors. The recovery of known sites is an index frequently reported in HT publications. Note that in spite of the fact that classic evidence is mostly *in vitro* binding, there is not a clear cut tendency of HT *in vitro* methods to recover more classic interactions than the *in vivo* methods.

As mentioned already, for 28 ChIP-seq experiments we also used a uniform bioinformatics pipeline to identify TF binding sites from raw data. In such cases we provide in the same dataset two tables, one with the data as extracted from authors, and one with the results of our in-house pipeline. We generated position weight matrices (PWMs) based on the in-house obtained sites for each dataset in addition to those existing in RegulonDB, and provide the distribution of sites in relation to the start of genes or promoters. Fig. 6 shows the number of classic TFRSs in RegulonDB that are found in the peak sequences as well as those found in peaks by motif matching. The results are quite variable depending on the TF studied. In particular, the Lrp and H-NS datasets show a low rate of recovery, which can be explained by the poor specificity of their PWMs in RegulonDB.

TSS, TU and TTS datasets

We gathered a collection of 16 TSS datasets from seven articles and one unpublished dataset (see Methods section), for a total of 68049 objects (Fig. 7). The TU and TTS collections each comprise five datasets from three articles, for a total of 12347 and 5326 objects respectively (see Tables S6–S8). The original data processed by the authors as well as our uniform datasets were compared with the RegulonDB classic collection. HT TSSs were mapped to classic TSSs when located within five bases on the

same strand. It is interesting to note that even though the total number of TSSs varies from 12000 to slightly less than 300 in the different datasets (Fig. 7a), the number of HT TSSs that match with LT TSSs is much less variable (Fig. 7b), just like the numbers of classic TSSs that match with HT datasets (Fig. 7c). It should be noted that those matches, although similar in number, are not symmetrical, as a result of the window-based mapping.

Gene expression datasets

To ensure high-quality comparisons of expression data, we assessed RNA-seq samples based on sequence read alignment metrics. We tagged as 'PASS' those samples with more than five million raw reads, more than 90% of their reads aligned to the *E. coli* reference genome, and more than 90% of genes with non-zero coverage. Out of 1864 total experiments, 648 were tagged as PASS. This collection offers processed expression values at the gene level. The expression values (counts, RPKM/FPKM, and TPM) from all 1864 experiments were normalized using the DEseq method described above, allowing users to make comparisons among any desired combination of experiments, whether or not they are tagged as 'PASS'.

The growth conditions for the GEO collection were extracted by our NLP method as mentioned in the Methods section. The trained predictive model (CRF) was used to automatically extract the growth conditions from the SOFT files associated with RNA-seq data (Fig. 3). The F1-score (the harmonic mean of precision and recall) of our predictive model was 0.81 in a five-fold cross-validation, and 0.83 in testing. Precision, also known as positive predictive value, was the proportion of true positive growth conditions among all conditions classified as positive by the model. Recall, also known as sensitivity, was the proportion of known positive growth conditions classified as positive by the model. Most growth conditions attained F1-scores above 0.80 (Table 2).

Following our assisted curation strategy, the most accurately predicted NLP-extracted growth conditions terms (probability >0.7) were manually reviewed. Only the correctly predicted terms were uploaded to the searching tool for RNA-seq datasets. These correct terms of growth conditions were mapped to MCO IDs before uploading to RegulonDB.

Our NLP method was applied to 1289 SOFT RNA-seq files, associated with 95 GSEs, 1289 SRRs (SRA accession IDs) for a total of 1157 GSMs or samples. We mapped to MCO IDs ~83% of terms (15% by exact matching, and ~68% by string similarity). The unmapped terms were also included in the RNA-seq searching tool of RegulonDB.

In summary, our NLP method provided 3224 terms supporting queries for 84 GSEs, 1131 SRRs for a total of 1001 GSMs. The percentage of SRRs (coverage) with any type of growth condition was different for each type (foreground bar plot in Fig. 8). For instance, temperature, medium and genetic background are reported in more than 35% of the 1001 (100%) SRRs. In spite of our good F-scores, we know from the training set that a large fraction of data is simply missing (background bar plot in Fig. 8). A lack of data for pH, agitation speed and optical density in the training set is shown, as in the NLP-extracted data. This is a pity since it limits the comparability and usability of the data, a well-known problem in database efforts in genomics [51].

On the other hand, we gathered metadata for 575 SRRs that were not found in GEO and had no available SOFT RNA-seq files. Using NCBI's Entrez tool we were able to retrieve at least one attribute for 520 SRRs. Genetic background and medium supplements were often recovered (520 and 506 SRRs, respectively). Culture medium and growth phase were recovered for only 91 and 80 SRRs, respectively (Table S9). Thus, we have metadata that allow datasets to be searched for 928 out of 1157 RNA-seq datasets from GEO, and for 520 out of 575, SRA experiments that could not be found in GEO.

All expression data is linked to a specific SRR ID. One GEO sample (GSM ID) could include more than one SRR ID and some SRRs are not found in GEO. We processed 1289 SRRs (1157 GEO samples) by the NLP method described earlier and the remaining 575 by the NCBI's Entrez tool strategy. We were able to retrieve at least one metadata attribute for 1131 (out of 1289) and 520 SRRs (out of 575), respectively. This implies that we do not have any metadata associated with 416 SRRs. All these experiments can only be searched based on their SRR ID in RegulonDB HT.

DISCUSSION

As mentioned before, gathering all publicly available HT data from *E. coli* K-12 in a single place would be of great benefit to advance research. In this work we present RegulonDB version 11.0, a major upgrade that offers the largest variety of publicly available HT data relevant to transcriptional regulation of *E. coli* K-12. We did not however update our ChIP-chip nor microarray datasets, and we did not include any Hi-C data.

Most HT data are deposited in repositories like GEO and ArrayExpress. Although GEO requires users to complete major fields to upload genomic datasets in a uniform way, there is a lack of guidelines, or final supervision, to guarantee standardized annotations. The lack of essential information allowing the reproducibility of experiments in the literature about transcriptional regulation became evident when we curated 600 papers in high detail to build the MCO, and found none that described the growth rate, and less than 100 provided the pH, among other properties [28] This represents a major known bottleneck for proper identification and use of HT datasets in downstream analyses [51], requiring manual curation of metadata prior to choosing a final collection to work with. Our application of a method combining Natural language processing and machine learning for the automatic

extraction of growth conditions from GEO files may greatly facilitate re-analysis of these datasets. We are working on improving the predictive model for growth condition extraction.

Another recurrent issue with HT datasets is that there is no standard way of processing the raw data, and a wide variety of tools and approaches can be used, depending on the original publications. Curation has been historically limited to reflect, as precisely as possible, what authors publish and report. A major novelty in RegulonDB 11.0 is the addition of in-house processed collections. The normalized RNA-seq collection standardizes analyses across individual datasets, in principle setting the basis for future tools that would allow users to select their 'control' and 'experimental' RNA-seq datasets and obtain the relative expression of novel comparisons. The uniformized ChIP-seq subcollection was generated using our publicly-available pipeline (see Methods and Data summary sections for details). This ensures its reproducibility, which is a frequent concern when analysing published datasets from numerous sources [1, 29]. Finally, we also offer uniformized TSS, TTS and TU collections. These data were all updated to the current annotated version of the *E. coli* K-12 genome. As updates occur, traceability will be supported by the corresponding versions in GitHub, keeping all details of the tools, parameters and thresholds at hand. The diversity of information and formats provided by authors makes it difficult to compare in a comprehensive way the results of our in-house processing with those provided originally, so we leave users with the liberty of choosing which dataset to use. In a future version we will add comparisons between them.

In the current version we are offering comparisons of some HT datasets with classic LT data from RegulonDB, considered as a gold standard. This way, users can easily evaluate how each HT dataset reproduces known data from classical experiments, which is the first question to arise when applying HT strategies. In the future we will compare as many HT datasets as possible with their corresponding classic corpus, and we plan also to provide comparisons across HT datasets. This information should be highly valuable for users to compare results from different sources and technologies.

Finally, we designed a new integrated web interface, including a genome viewer and increased search capabilities. Previous RegulonDB searching capabilities were limited to TF and object type. We now allow searching for many other fields like, author, PMID, TF, growth conditions, and many more. These metadata are valuable for search and re-analysis of more than two thousand HT datasets gathered in this version.

Besides the technical aspects of the management of HT datasets we described above, we have been revisiting fundamental biological concepts. An important conceptual distinction that HT methods require for their precise description is the one between the ability to bind to specific DNA operator sites, and the capacity to alter the activity of a given promoter. Current HT publications frequently combine a binding experiment like ChIP-seq for instance, with a global expression experiment (i.e. RNA-seq) performed in the same experimental conditions. In this way it is possible to identify those sites that bind, defining TFBSs, and those that bind and modify the expression of a downstream gene, defining TFRSs. The distinction between TFBSs and TFRSs was proposed in the recent update of concepts of gene regulation [2], motivated in fact by the type of data generated with novel post-genomic technologies. By the same token, there are many potential TFs that have been assayed for instance with DAP-seq and gSELEX but have no evidence yet of any concomitant change of expression for a target gene, and therefore, as mentioned before, they do not satisfy the requirements to be fully identified yet as TFs. Lastly, we formally distinguished promoters from TSSs and terminators from TTSs, terms that are frequently used interchangeably in publications.

The version 11.0 of RegulonDB, presented here, represents an important quantitative and qualitative upgrade, offering novel features that make our repository the most comprehensive resource to utilize the wealth of HT data available, together with knowledge accumulated through decades of research with classic molecular biology approaches. We expect this unique resource will help advance research in *E. coli* K-12.

Funding information
We acknowledge funding from Universidad Nacional Autónoma de México (UNAM), as well as funding by NIGMS-NIH grant number 5RO1GM131643, and by UNAM-PAPIIT IA203420. GM-H is funded by The Natural Sciences and Engineering Council of Canada (NSERC). CR is a doctoral student from the Programa de Doctorado en Ciencias Biomédicas, UNAM, and has received fellowship 929687 from CONACyT. PL acknowledges a postdoctoral fellowship from DGAPA-UNAM. EGN thanks DGAPA-UNAM for the scholarships 181821, 369220.

Acknowledgements
We acknowledge Peter L. Freddolino, Laurence Ettwiller, Bo Yan, Xiangwu Ju and Akira Ishihama for fruitful discussion on proper interpretation of their datasets. We acknowledge the observations by anonymous referees, and also thank IT support by Víctor Del Moral.

Author contributions
V.H.T.: Data curation, conceptualization, methodology, validation, writing, review and editing. C.R.: Conceptualization, software, formal analysis, validation, visualization, writing, review and editing. H.S.: Conceptualization, validation, visualization, software, writing, review and editing. P.L.: Data curation, conceptualization, formal analysis, writing, review and editing. L.G.R.: Software, writing. P.P.L.: Software. A.G.L.A.: Software. G.A.C.: Software. F.B.F.: Software. S.A.H.: Resources. J.E.P.M.: Software. J.G.S.: Software. S.G.C.: Data curation, formal analysis. E.G.N.: Formal analysis, software, writing, review. C.F.M.C.: Formal analysis, methodology, supervision, writing, review. C.B.M.: Software, visualization. L.J.M.: Software. G.M.H.: Formal analysis, writing, review and editing. J.E.G.: Conceptualization, formal analysis, funding acquisition. J.T.W.: Conceptualization,

funding acquisition, data contributor, writing, review and editing, J.C.V.: Conceptualization, supervision, funding acquisition, writing, review and editing.

Conflicts of interest
The authors declare that there are no conflicts of interest.

References

1. Rioualen C, Charbonnier-Khamvongsa L, Collado-Vides J, van Helden J. Integrating bacterial ChIP-seq and RNA-seq data with snakechunks. *Curr Protoc Bioinformatics* 2019;66:e72.
2. Mejía-Almonte C, Busby SJW, Wade JT, van Helden J, Arkin AP, *et al*. Redefining fundamental concepts of transcription initiation in bacteria. *Nat Rev Genet* 2020;21:699–714.
3. Santos-Zavaleta A, Salgado H, Gama-Castro S, Sánchez-Pérez M, Gómez-Romero L, *et al*. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in *E. coli* K-12. *Nucleic Acids Res* 2019;47:D212–D220.
4. Keseler IM, Gama-Castro S, Mackie A, Billington R, Bonavides-Martínez C, *et al*. The EcoCyc Database in 2021. *Front Microbiol* 2021;12:711077.
5. Johnson DS, Mortazavi A, Myers RM, Wold B. Genome-wide mapping of *in vivo* protein-DNA interactions. *Science* 2007;316:1497–1502.
6. Robertson G, Hirst M, Bainbridge M, Bilenky M, Zhao Y, *et al*. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nat Methods* 2007;4:651–657.
7. Seo SW, Kim D, Latif H, O'Brien EJ, Szubin R, *et al*. Deciphering Fur transcriptional regulatory network highlights its complex role beyond iron metabolism in *Escherichia coli*. *Nat Commun* 2014;5:4910.
8. O'Malley RC, Huang S-SC, Song L, Lewsey MG, Bartlett A, *et al*. Cistrome and epicistrome features shape the regulatory DNA landscape. *Cell* 2016;165:1280–1292.
9. Shimada T, Ogasawara H, Ishihama A. Genomic SELEX screening of regulatory targets of *Escherichia coli* transcription factors. *Methods Mol Biol* 2018;1837:49–69.
10. Salgado H, Peralta-Gil M, Gama-Castro S, Santos-Zavaleta A, Muñiz-Rascado L, *et al*. RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic Acids Res* 2013;41:D203-13.
11. Conway T, Creecy JP, Maddox SM, Grissom JE, Conkle TL, *et al*. Unprecedented high-resolution view of bacterial operon architecture revealed by RNA sequencing. *mBio* 2014;5:e01442-14.
12. Ettwiller L, Buswell J, Yigit E, Schildkraut I. A novel enrichment strategy reveals unprecedented number of novel transcription start sites at single base resolution in a model prokaryote and the gut microbiome. *BMC Genomics* 2016;17:199.
13. Yan B, Boitano M, Clark TA, Ettwiller L. SMRT-Cappable-seq reveals complex operon variants in bacteria. *Nat Commun* 2018;9:3676.
14. Ju X, Li D, Liu S. Full-length RNA profiling reveals pervasive bidirectional transcription terminators in bacteria. *Nat Microbiol* 2019;4:1907–1918.
15. Kurata T, Katayama A, Hiramatsu M, Kiguchi Y, Takeuchi M, *et al*. Identification of the set of genes, including nonannotated morA, under the direct control of ModE in *Escherichia coli*. *J Bacteriol* 2013;195:4496–4505.
16. Shimada T, Kori A, Ishihama A. Involvement of the ribose operon repressor RbsR in regulation of purine nucleotide synthesis in *Escherichia coli* . *FEMS Microbiol Lett* 2013;344:159–165.
17. Shimada T, Katayama Y, Kawakita S, Ogasawara H, Nakano M, *et al*. A novel regulator RcdA of the csgD gene encoding the master regulator of biofilm formation in *Escherichia coli*. *Microbiologyopen* 2012;1:381–394.
18. Aquino P, Honda B, Jaini S, Lyubetskaya A, Hosur K, *et al*. Coordinated regulation of acid resistance in *Escherichia coli*. *BMC Syst Biol* 2017;11:1.
19. Fitzgerald DM, Bonocora RP, Wade JT, Søgaard-Andersen L. Comprehensive mapping of the *Escherichia coli* flagellar regulatory network. *PLoS Genet* 2014;10:e1004649.
20. Gao Y, Lim HG, Verkler H, Szubin R, Quach D, *et al*. Unraveling the functions of uncharacterized transcription factors in *Escherichia coli* using ChIP-exo . *Nucleic Acids Res* 2021;49:9696–9710.
21. Seo SW, Kim D, O'Brien EJ, Szubin R, Palsson BO. Decoding genome-wide GadEWX-transcriptional regulatory networks reveals multifaceted cellular responses to acid stress in *Escherichia coli*. *Nat Commun* 2015;6:7970.
22. Mendoza-Vargas A, Olvera L, Olvera M, Grande R, Vega-Alvarado L, *et al*. Genome-wide identification of transcription start sites, promoters and transcription factor binding sites in *E. coli*. *PLoS One* 2009;4:10.
23. Gama-Castro S, Salgado H, Santos-Zavaleta A, Ledezma-Tejeida D, Muñiz-Rascado L, *et al*. RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Res* 2016;44:D133-43.
24. Santos-Zavaleta A, Sánchez-Pérez M, Salgado H, Velázquez-Ramírez DA, Gama-Castro S, *et al*. A unified resource for transcriptional regulation in *Escherichia coli* K-12 incorporating high-throughput-generated binding data into RegulonDB version 10.0. *BMC Biol* 2018;16:91.
25. Moretto M, Sonego P, Dierckxsens N, Brilli M, Bianco L, *et al*. COLOMBOS v3.0: leveraging gene expression compendia for cross-species analyses. *Nucleic Acids Res* 2016;44:D620-3.
26. Ishihama A, Shimada T, Yamazaki Y. Transcription profile of *Escherichia coli*: genomic SELEX search for regulatory targets of transcription factors. *Nucleic Acids Res* 2016;44:2058–2074.
27. Decker KT, Gao Y, Rychel K, Al Bulushi T, Chauhan SM, *et al*. proChIPdb: a chromatin immunoprecipitation database for prokaryotic organisms. *Nucleic Acids Res* 2022;50:D1077–D1084.
28. Tierrafría VH, Mejía-Almonte C, Camacho-Zaragoza JM, Salgado H, Alquicira K, *et al*. MCO: towards an ontology and unified vocabulary for a framework-based annotation of microbial growth conditions. *Bioinformatics* 2019;35:856–864.
29. Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, *et al*. Sustainable data analysis with Snakemake. *F1000Res* 2021;10:33.
30. Robinson JT, Thorvaldsdóttir H, Turner D, Mesirov JP. IGV.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV). *Bioinformatics* 2020.
31. Seo SW, Kim D, Szubin R, Palsson BO. Genome-wide reconstruction of OxyR and SoxRS transcriptional regulatory networks under oxidative stress in *Escherichia coli* K-12 MG1655. *Cell Rep* 2015;12:1289–1299.
32. Zere TR, Vakulskas CA, Leng Y, Pannuri A, Potts AH, *et al*. Genomic targets and features of Bara-uvry (-sira). *Signal Transduction Systems PLoS One* 2015;10:12.
33. Ueguchi C, Mizuno T. The *Escherichia coli* nucleoid protein H-NS functions directly as a transcriptional repressor. *EMBO J* 1993;12:1039–1046.
34. Antipov SS, Tutukina MN, Preobrazhenskaya EV, Kondrashov FA, Patrushev MV, *et al*. The nucleoid protein Dps binds genomic DNA of *Escherichia coli* in a non-random manner. *PLoS One* 2017;12:e0182800.
35. Prieto AI, Kahramanoglou C, Ali RM, Fraser GM, Seshasayee ASN, *et al*. Genomic analysis of DNA binding and gene regulation by homologous nucleoid-associated proteins IHF and HU in *Escherichia coli* K12. *Nucleic Acids Res* 2012;40:3524–3537.
36. Lim CJ, Lee SY, Teramoto J, Ishihama A, Yan J. The nucleoid-associated protein Dan organizes chromosomal DNA through rigid nucleoprotein filament formation in *E. coli* during anoxia. *Nucleic Acids Res* 2013;41:746–753.

37. Baumgart LA, Lee JE, Salamov A, Dilworth DJ, Na H, *et al*. Persistence and plasticity in bacterial gene regulation. *Nat Methods* 2021;18:1499–1505.
38. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet j* 2011;17:10.
39. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. *Nat Methods* 2012;9:357–359.
40. Wingett SW, Andrews S. FastQ Screen: A tool for multi-genome mapping and quality control. *F1000Res* 2018;7:1338.
41. Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* 2016;32:3047–3048.
42. Feng J, Liu T, Zhang Y. Using MACS to identify peaks from ChIP-Seq data. *Curr Protoc Bioinformatics* 2011;Chapter 2:Unit.
43. Turatsinze J-V, Thomas-Chollier M, Defrance M, van Helden J. Using RSAT to scan genome sequences for transcription factor binding sites and cis-regulatory modules. *Nat Protoc* 2008;3:1578–1588.
44. Medina-Rivera A, Abreu-Goodger C, Thomas-Chollier M, Salgado H, Collado-Vides J, *et al*. Theoretical and empirical quality assessment of transcription factor-binding motifs. *Nucleic Acids Res* 2011;39:808–824.
45. Thomas-Chollier M, Darbo E, Herrmann C, Defrance M, Thieffry D, *et al*. A complete workflow for the analysis of full-size ChIP-seq (and similar) data sets using peak-motifs. *Nat Protoc* 2012;7:1551–1568.
46. Nguyen NTT, Contreras-Moreira B, Castro-Mondragon JA, Santana-Garcia W, Ossio R, *et al*. RSAT 2018: regulatory sequence analysis tools 20th anniversary. *Nucleic Acids Res* 2018;46:W209–W214.
47. Thomason MK, Bischler T, Eisenbart SK, Förstner KU, Zhang A, *et al*. Global transcriptional start site mapping using differential RNA sequencing reveals novel antisense RNAs in *Escherichia coli*. *J Bacteriol* 2015;197:18–28.
48. Cho BK, Kim D, Knight EM, Zengler K, Palsson BO. Genome-scale reconstruction of the sigma factor network in *Escherichia coli*: topology and functional states. *BMC Biol* 2014;12:4.
49. Lafferty JD, McCallum A, Pereira FCN. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the Eighteenth International Conference on Machine Learning. 2001; Morgan Kaufmann Publishers Inc.: 282–9.
50. Peng F, McCallum A. Information extraction from research papers using conditional random fields. *Information Processing & Management* 2006;42:963–979.
51. Bernstein MN, Doan A, Dewey CN. MetaSRA: normalized human sample-specific metadata for the Sequence Read Archive. *Bioinformatics* 2017;33:2914–2923.
52. Huerta AM, Salgado H, Thieffry D, Collado-Vides J. RegulonDB: a database on transcriptional regulation in *Escherichia coli*. *Nucleic Acids Res* 1998;26:55–59.

Chapter 5.

An alternative collection of binding motifs

Problematic

As mentioned before, transcription factors can bind DNA via specific sequences, called transcription factor binding sites. Each TF has its own specific binding motif based on the DNA patterns its binding domain has affinity with. However, most of these motifs are still unknown: about 70% of *E. coli* TFs still have unknown binding patterns, and some 30% have no binding sites identified at all. Moreover, some of the available motifs are somewhat fuzzy, for they're built on few DNA sequences.

In this chapter, I explain how I used RegulonDB's carefully curated binding sites (Tierrafría, Rioualen et al., 2022), together with a pattern-discovery strategy, in order to propose an alternative collection of TF binding matrices for *E. coli* K-12. Additionally, I produced matrices using public ChIP-seq datasets processed using my own framework (Chapter 3, Chapter 4).

Motifs and matrices of *E. coli* K-12

All transcription factors comprise a DNA-binding domain, and therefore the ability to bind to specific locations of the DNA. By aligning the set of DNA binding sequences of a given TF and computing nucleotide frequencies, one can represent its binding motif in the form of a degenerated consensus sequence, position-specific scoring matrix (Stormo et al., 1982), or a logo image (Figure 18). A motif should allow to distinguish a binding site from the background genomic sequence, and illustrate the specificity of the binding TF. Its discriminative power can be measured using the information

content (IC), which denotes how much a given matrix differs from the background nucleotidic composition.

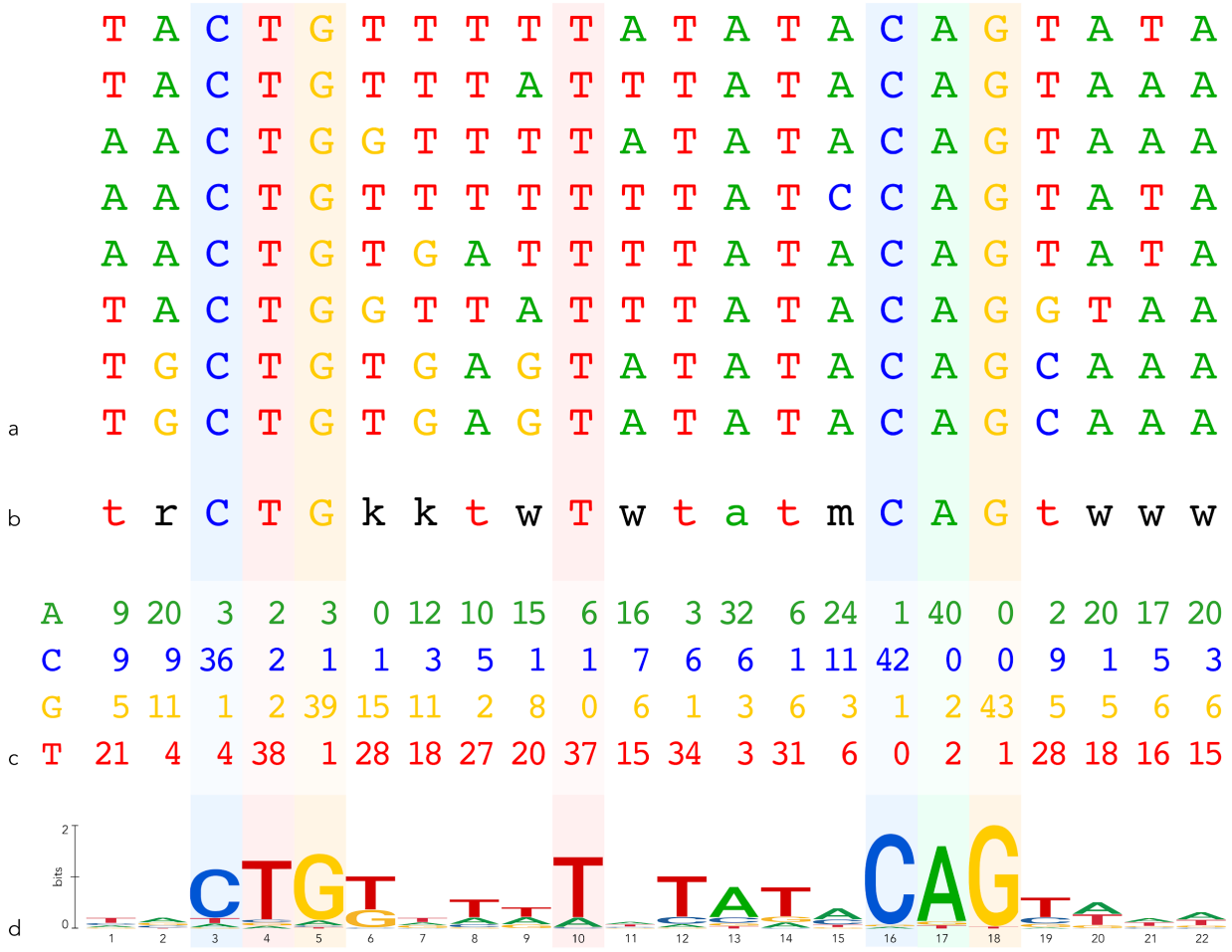


Figure 18. Example: LexA binding sites and motif. a. A subset of the 44 LexA TFBSs available in RegulonDB are aligned **b.** The corresponding consensus sequence based on the IUPAC code. **c.** The complete position-specific scoring matrix built from 44 TFBSs. **d.** The logo representation. Adapted from Medina-Rivera et al., 2011.

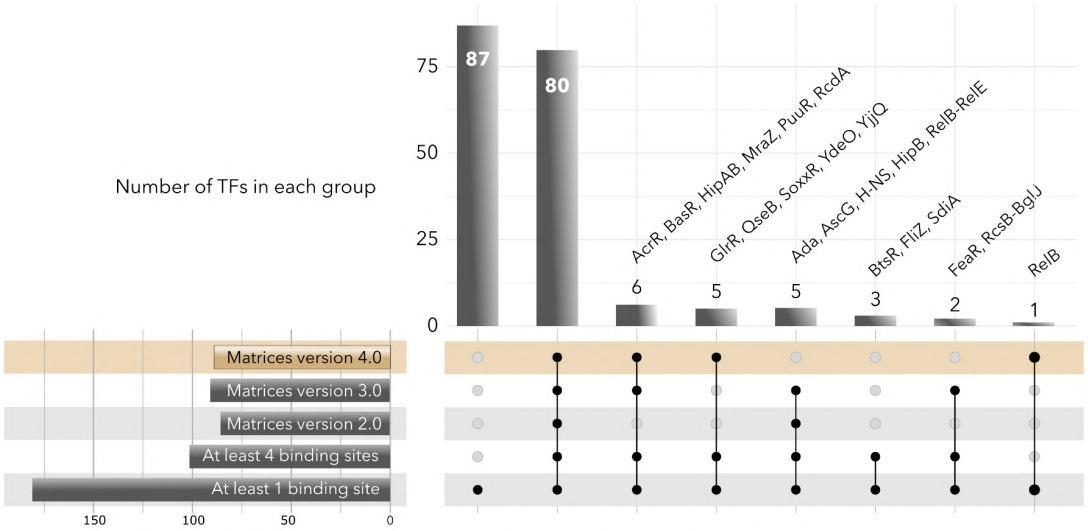
As of today, it is considered that *Escherichia coli* K-12 has about 300 transcription factors, although not all of them are formally identified as such (Chapter 1). In RegulonDB, a TF is considered "confirmed" when it has at least one *strong* regulatory interaction characterized based on experimental evidence. As of RegulonDB version 10.10 (released in Feb. 2022), a total of 2,549 binding sites are associated with 189 TFs, and 92 TFs have a matrix built from a minimum of 4 distinct binding sites sequences,

using the program MEME (Bailey et al., 2015) to build matrices from sequences, and RSAT matrix-quality to select the best matrix for each TF (Medina-Rivera et al., 2011). Most of this knowledge comes from low-throughput *in vitro* experiments, and has been manually curated from the literature into the RegulonDB database. While it is frequently actualized with new binding sites, the collection of matrices has not been updated on a regular basis, mostly due to the difficulty of automatizing its construction and validation. Overall, a comparison of the last three versions of the collection shows little evolution (Figure 19). While the number of binding sites has increased, as well as the total information content of the matrices, the length of the matrices has also increased and the binding information was watered-down, which ultimately results in a decreased information content per column, or average information content per position (Figure 19c).

a

RegulonDB matrix collections				
Version 4.0	92 TFs	2,113 TFBSs	RegulonDB 10.6	2019, July
Version 3.0	93 TFs	2,042 TFBSs	RegulonDB 8.8	2015, May
Version 2.0	85 TFs	1,913 TFBSs	RegulonDB 7.4	2012, March

b



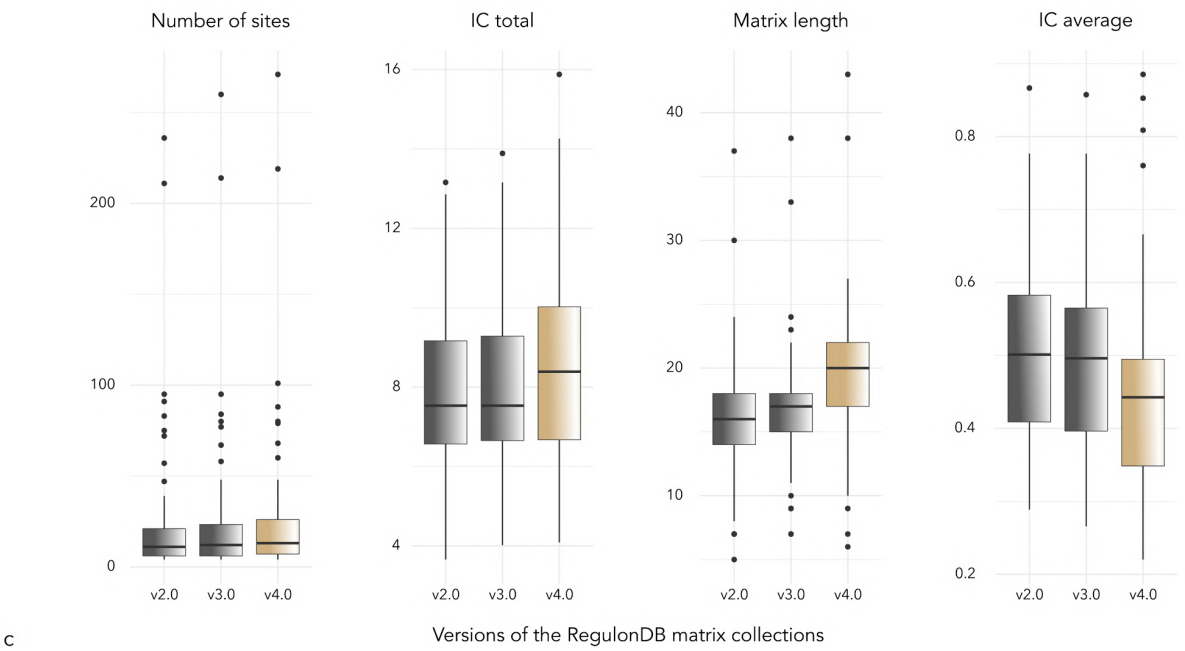


Figure 19. Statistics from the RegulonDB motif collection and its past versions. a. Summary of the last three versions of the collection. **b.** Comparison of the TFs included in each version, and the TFs that currently have at least 1 or 4 binding sites in the database. **c.** Evolution of the distribution of several matrix parameters over time: number of sites used to build each matrix, total information content, matrix length and information content per column. Note: from version 4.0 the TF RelB-RelE is annotated as RelB. Both are considered as the same TF in subsequent analyses.

A visual inspection of RegulonDB's updated matrices shows that a handful of TFs indeed seem to have weak motifs. On one hand, matrices made of too few site sequences might not reach a high resolution (Figure 20a), however, on the other hand a high number of sequences might actually dilute the core motif (Figure 20b). Some motifs have a poor nucleotidic complexity (Figure 20c), and subsequently have a low discriminative power. Due to the fact that many TFs form multimers in their active form, and some of them have various binding sites in the same vicinity, there might also be larger, spurious motifs as a result (Figure 20d). Lastly, most TFs don't have a matrix at all, for their binding sites are mostly or completely unknown.

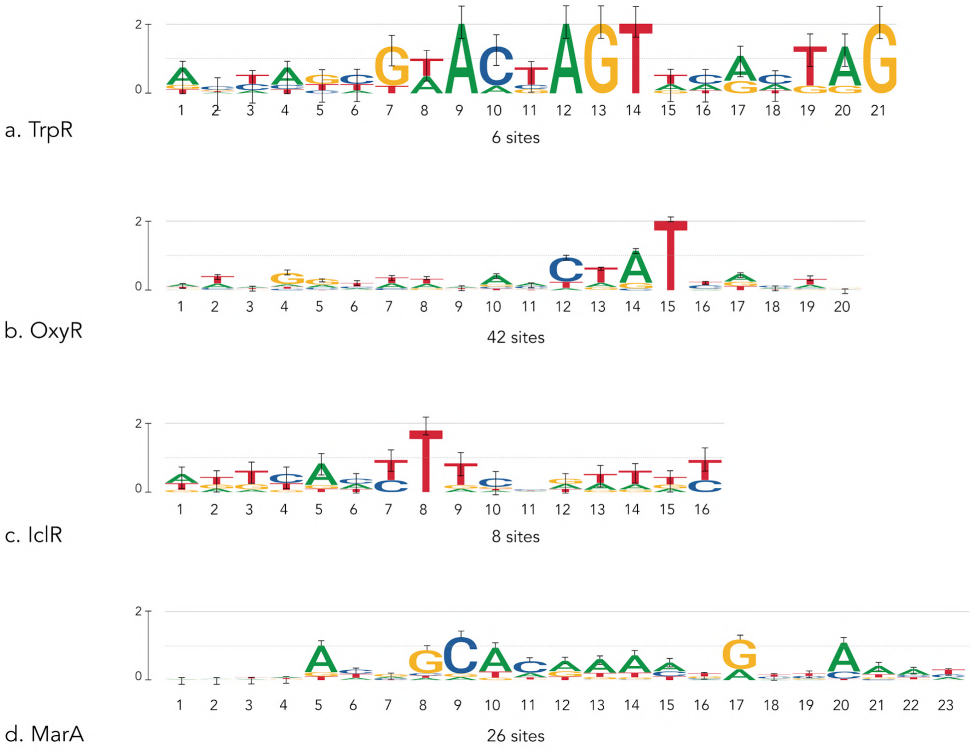


Figure 20. Examples of motifs from the RegulonDB collection version 4.0. **a.** TrpR has a rather well-defined motif, but shows low confidence due to a limited number of sequences available. **b.** OxyR has 42 site sequences available, but very little information shows in the logo. **c.** IclR has an AT-rich motif that shows poor specificity. **d.** MarA has a very large motif, but most of its positions hold little to no information.

Extraction of motifs through pattern discovery

Obtaining robust motifs is not a straightforward process. A matrix is basically a representation of the binding information contained in a collection of TFBS, however, it can fulfill several purposes. It should accurately represent the specificity of a given TF, allowing it to identify and distinguish said TF binding sites from the genomic background, as well as from other TFs' binding sites. A collection of motifs allows one to classify TFs according to their binding profiles similarity. A good motif can also be used to quantify the affinity of a given site, to observe variations between several binding sequences, or even predict novel binding sites from larger genomic sequences. Furthermore, at a multi-species level, the analysis of motifs and binding sites'

conservation can give further insights into distinct TF conformations and regulatory mechanisms (Oliver et al., 2016).

In bacteria, most transcription factors are known to possess HTH motifs in their DBD, and to be active in a dimeric form (and at times, in tetrameric or hexameric conformations). Consequently, they tend to have dyadic motifs: a pair of short sequences (3-5bp) separated by a less conserved sequence, which length is variable and depends on each TF. These sequences are generally reverse palindromes, and in fewer cases, direct repeats or distinct words (in the case of heterodimeric TFs), separated by non-specific, AT-rich segments that provide DNA flexibility. In order to build an alternative collection of motifs, I used pattern discovery algorithms that take advantage of those properties.

The algorithm *dyad-analysis* (van Helden et al., 2000) from the Regulatory Sequence Analysis Tools suite (Santana-Garcia et al., 2022) was specifically developed to identify dyadic motifs. It assumes that such motifs can be modeled as follows:

$D = w_1 \cdot n_s \cdot w_2$

Where:

- D = sequence of a dyad
- w_1 and w_2 = first and second words of the dyad
- n_s = any sequence of s unspecified nucleotides

Although most TFs bind to some kind of dyadic pattern, these are not always fully conserved, and in some cases one of two words is hardly detectable, which is why I also used the algorithm *oligo-analysis* (van Helden et al., 1998), that identifies significantly over-represented oligonucleotides given a background model. Indeed, it has been shown that less-conserved motifs can be just as biologically relevant as conserved motifs (Oliver et al., 2016). Furthermore, some 20% of bacterial transcription factors are believed to present non-canonical binding domains (Flores-Bautista et al., 2020), thus their binding sites could present distinct patterns.

I ran both algorithms for 101 transcription factors that had at least 4 distinct binding sites currently indexed in RegulonDB (Table 5). A manual selection of motifs was made based on those results. Some motifs were found by both algorithms, in which case the

version found with dyad-analysis was kept, some were found only by one algorithm, and in a few cases, no algorithm could find a significant pattern.

Overall, a collection of 101 motifs was built: 9 motifs were produced for TFs that didn't have one yet, 85 motifs were updated, and 7 TF motifs were kept in their original version, since neither algorithm detected significant patterns among their binding sequences (Figure 21).

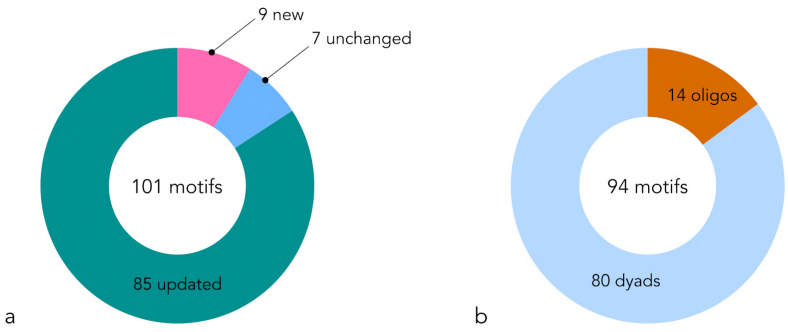


Figure 21. Overview of the alternative collection of motifs. a. Number of motifs newly created, updated, or unchanged. **b.** Type of algorithm used to produce the new and updated motifs.

The alternative motif collection

The pattern discovery strategy I designed has two significant differences compared to the original strategy used to build the RegulonDB collection: it doesn't necessarily use all of the input site sequences to build the matrix, and it can use a given sequence several times, should there be a duplicated pattern. It is also worth noting that the RegulonDB v4.0 collection was built on RegulonDB 10.6 (July 2019; total TFBS = 2,113), while I built the alternative collection on RegulonDB 10.10 (February 2022; TFBS = 2,549). Still, overall the total number of sequences used to build the new matrices is lower, despite the total information content being significantly higher. Since the pattern-discovery strategy allows to leave out poorly-conserved sequences and trim out the low-information positions from both ends of the patterns identified, the alternative matrices have a smaller length, and the information content per column is much higher (Figure 22).

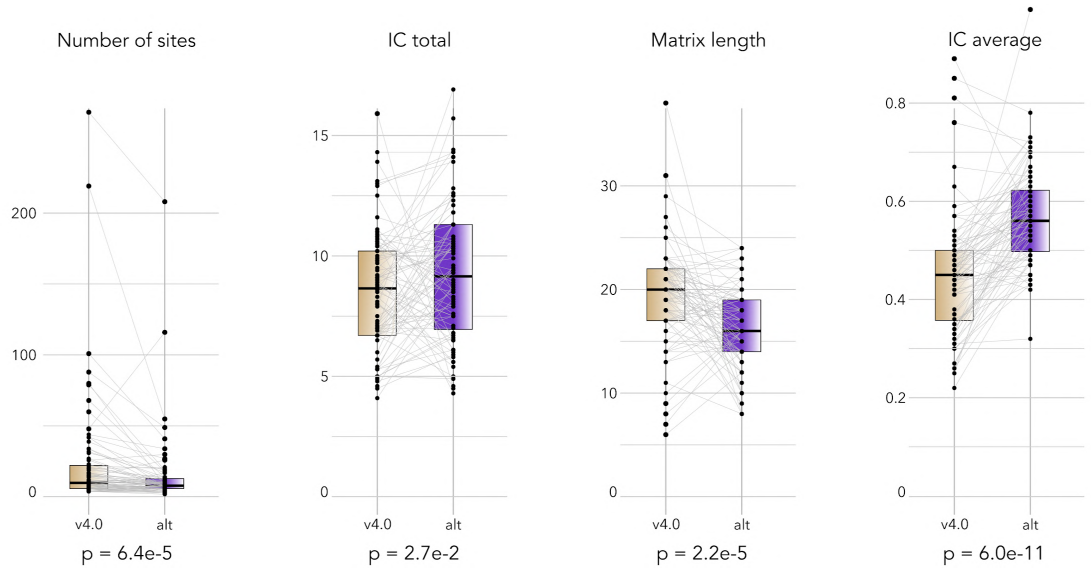


Figure 22. Comparison of the RegulonDB collection and the alternative collection. Beige boxes represent the current RegulonDB collection (version 4.0), while purple boxes represent the alternative collection. The statistical significance of the comparisons was computed for 92 TFs using paired Wilcoxon tests, leaving out the 9 TFs that have a newly-generated motif.

I produced exhaustive graphical reports allowing to visualize and quantify the changes between the current motif collection from RegulonDB, and the new alternative collection. Although the new matrices are built with fewer sites, the pattern discovery strategy produced motifs that show a better resolution, as well as a clear symmetry. While these motifs may not represent the complete set of binding sites underneath, in particular those that are less conserved, they offer a clear visualization of the core dyadic motifs (Figure 23).

As for the newly-built matrices, although some are rather weak given the few sequences used as an input, most do give an idea of the possible pattern behind (Figure 24). The full list of matrices and their parameters are summarized in Table 5.

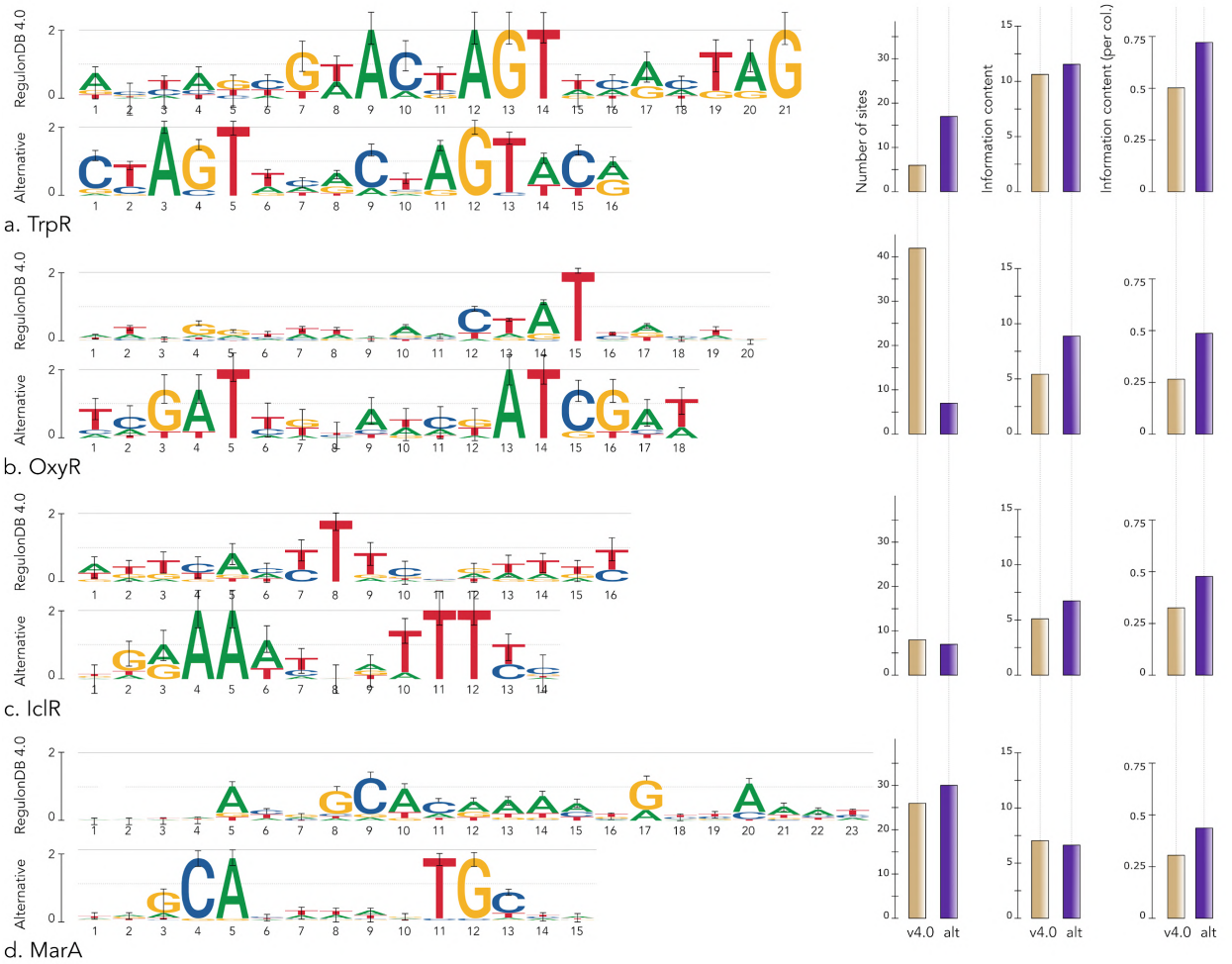


Figure 23. Updated motifs and their basic statistics. **a.** The TrpR motif gained in resolution, while shortening in length. **b.** The OxyR alternative motif was built using just a subset of its known sites, but its pattern is much clearer. **c.** A symmetric pattern was identified in most of Iclr binding sites, which is a lot more specific. **d.** The MarA motif was trimmed to ⅔ of its size, but shows a well-defined dyadic pattern.

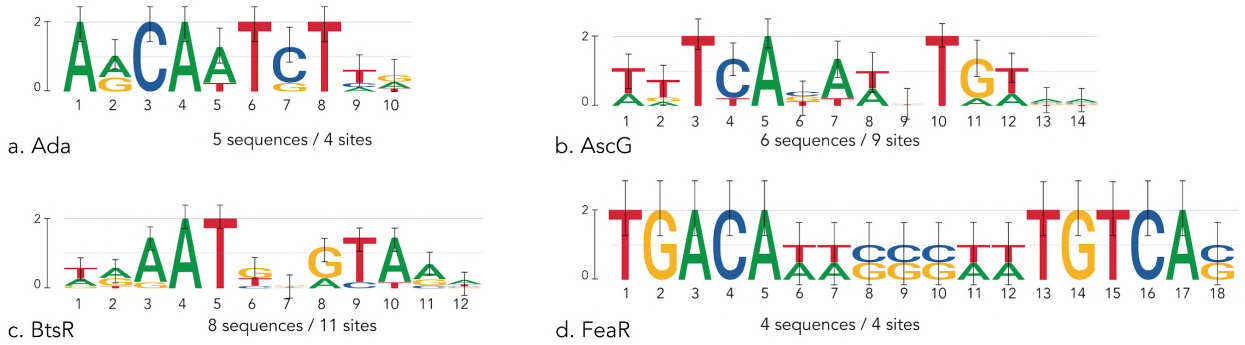


Figure 24. Some of the newly-generated matrices.

		RegulonDB matrices (v4.0)			Alternative matrices		
	TF name	IC total	IC average	Num TFBS	IC total	IC average	Num seq
new	Ada				6	0.6	5
	AscG				6.2	0.44	6
	BtsR				5.6	0.47	8
	FeaR				10.9	0.61	4
	FliZ				5	0.56	5
	H-NS				8.9	0.59	9
	HipB				14.2	0.57	8
	RcsB-BglJ				9.1	0.65	15
	SdiA				9.3	0.55	3
unchanged	ArgP	5.4	0.28	19			
	CsgD	4.6	0.33	24			
	CysB	12.7	0.3	14			
	QseB	7.3	0.46	4			
	RcsB	7.5	0.44	26			
	RhaS	7.6	0.38	8			
	Rob	8	0.35	14			
updated	AcrR	7	0.5	5	7	0.5	7
	AgaR	11.6	0.51	11	10.6	0.56	13
	AraC	7	0.35	15	7.9	0.49	7
	ArcA	5.3	0.31	79	11.3	0.59	19
	ArgR	8.9	0.43	31	10.3	0.6	19
	AsnC	8	0.47	4	6	0.54	8
	BaeR	12.5	0.54	4	6.5	0.65	4
	BasR	10.8	0.47	8	12.3	0.56	7
	CaiF	10.9	0.57	4	10.5	0.58	4
	CpxR	5.4	0.27	60	10.6	0.56	14
	Cra	9.5	0.45	42	12.1	0.61	27
	CRP	7	0.35	271	9.5	0.5	208
	CytR	5.7	0.25	17	11.3	0.7	10
	Dan	5	0.45	5	4.9	0.61	4
	DcuR	9.2	0.44	6	7.9	0.53	8
	DeoR	9.1	0.48	7	11.8	0.59	7
	DnaA	6.7	0.67	23	10.4	0.65	8
	EvgA	13	0.59	8	13.9	0.63	10
	ExuR	10.4	0.52	11	8.9	0.64	12
	FadR	9.7	0.48	20	7.6	0.54	12
	FhlA	7.5	0.44	7	9.3	0.58	6
	Fis	4.5	0.26	219	5.9	0.65	55
	FlhDC	8.8	0.46	16	7.5	0.62	8
	FNR	6.7	0.48	88	11.3	0.54	49
	Fur	9.8	0.44	48	10.8	0.57	116
	GadE	8.1	0.37	10	4.6	0.58	10

	RegulonDB matrices (v4.0)			Alternative matrices		
TF name	IC total	IC average	Num TFBS	IC total	IC average	Num seq
GadW	7.1	0.34	17	7.5	0.58	4
GadX	4.8	0.22	34	5.8	0.48	12
GalR	9.1	0.51	12	10.3	0.73	10
GalS	10	0.53	12	10.3	0.73	10
GcvA	8	0.44	6	5.4	0.45	3
GlpR	9.1	0.41	17	10.2	0.49	9
GlrR	8.6	0.38	6	12.8	0.53	6
GntR	10.2	0.46	9	12.5	0.7	12
HipAB	13.1	0.45	5	14.1	0.59	8
IclR	5	0.32	8	6.6	0.47	7
IHF	4.9	0.35	101	8.2	0.52	41
IscR	9	0.33	11	5.6	0.56	9
LeuO	8.3	0.36	10	5.6	0.43	11
LexA	10.2	0.46	44	14.4	0.63	34
Lrp	4.1	0.31	80	8.8	0.52	8
MalT	7.3	0.81	15	11.3	0.7	10
MarA	6.9	0.3	26	6.5	0.43	30
MelR	10.6	0.5	5	9	0.45	4
MetJ	7.9	0.42	15	16.9	0.99	13
MetR	8.5	0.47	5	7.1	0.42	8
Mlc	13.9	0.53	7	8.5	0.43	8
MlrA	15.9	0.42	4	4.5	0.45	2
MntR	14.3	0.57	6	14.3	0.65	10
ModE	11	0.41	7	6.1	0.61	8
MqsA	9.2	0.44	6	8.5	0.57	6
MraZ	5	0.63	6	6.7	0.48	3
Nac	5.3	0.31	14	6.1	0.32	26
NagC	11	0.44	20	6.1	0.47	21
NanR	6	0.85	9	12.6	0.78	6
NarL	4.6	0.27	68	9.6	0.53	18
NarP	7	0.44	12	15.7	0.71	8
NhaR	8.7	0.41	6	7.5	0.44	6
NrdR	9.4	0.5	6	9.9	0.58	5
NsrR	7	0.47	39	9.4	0.67	17
NtrC	9.8	0.49	27	14.1	0.62	8
OmpR	6.7	0.3	21	8.5	0.45	13
OxyR	5.3	0.26	42	8.7	0.48	7
PdhR	10.2	0.54	9	12.5	0.6	7
PhoB	6.7	0.33	26	12.5	0.69	7
PhoP	6.7	0.37	34	8	0.54	7
PurR	12.9	0.76	22	14.3	0.72	20
PutA	5.3	0.89	5	6.8	0.57	6
PuuR	11.1	0.5	4	8.1	0.48	3

	RegulonDB matrices (v4.0)			Alternative matrices		
TF name	IC total	IC average	Num TFBS	IC total	IC average	Num seq
RcdA	7.1	0.51	8	10.4	0.58	4
RcsAB	8.8	0.49	6	6.8	0.57	6
RelB-RelE	6.4	0.43	4	7.8	0.46	10
RstA	10.5	0.52	4	8.3	0.55	5
RutR	9.9	0.49	5	10.5	0.66	6
SlyA	6.7	0.44	14	9.3	0.47	7
SoxR	6.7	0.34	6	5	0.56	4
SoxS	7.2	0.33	32	10.7	0.56	14
TorR	6.5	0.5	10	7.5	0.5	5
TrpR	10.4	0.49	6	11.3	0.71	17
TyrR	9	0.43	19	11.8	0.49	14
UlaR	10.8	0.47	4	8.6	0.48	6
UxuR	10.7	0.51	8	10.1	0.67	11
XylR	8.6	0.41	8	6.5	0.5	10
YdeO	10.6	0.34	6	4.3	0.54	6
YjjQ	10.2	0.57	7	9.4	0.52	10

Table 5. The complete list of matrices and associated parameters for 101 TFs.

Evaluation of motifs quality

As mentioned above, a motif should represent the specificity of a TF binding pattern, and allow to distinguish potential binding sites from the genomic background. Although there is no single metric that could formally quantify the quality of a motif, several criteria can be explored that show an overall tendency. For one, the visual inspection of logos gives a quick impression of the precision and specificity of a motif, as well as its "shape" in the particular case of dyadic motifs. The specificity can also be quantified by calculating the information content (IC) of the matrix. Since it is highly dependent on the length of the matrix, the average IC per position shall also be taken into account.

However, all of these criteria are somewhat imperfect, thus the RSAT tool *matrix-quality* was developed in order to assess the quality of matrices (Medina-Rivera et al., 2011). It combines theoretical and empirical score distributions for sets of genomic sequences given a PSSM in order to estimate its predictive capacity. It is based on the RSAT program *matrix-scan* (Turatsinze et al., 2008). Matrix-scan was developed

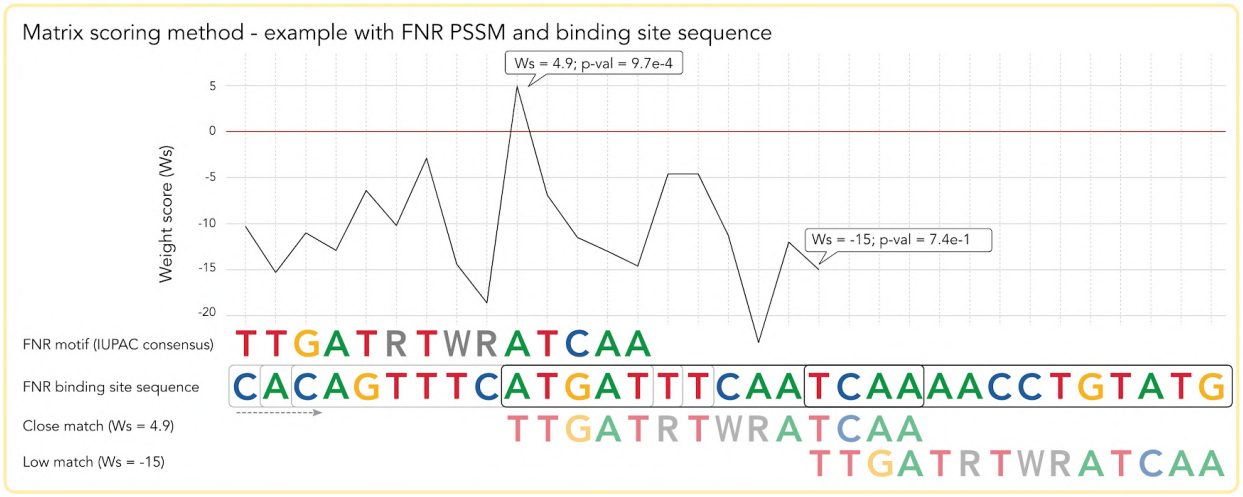
to scan genome sequences and detect potential transcription factor binding sites and cis-regulatory modules, by computing weight score distributions at each position of the input sequences (Figure 25, yellow box). It uses a background model B and a reference matrix M, and calculates for each sequence segment S if it's likely to be an instance of the motif rather than an instance of the background:

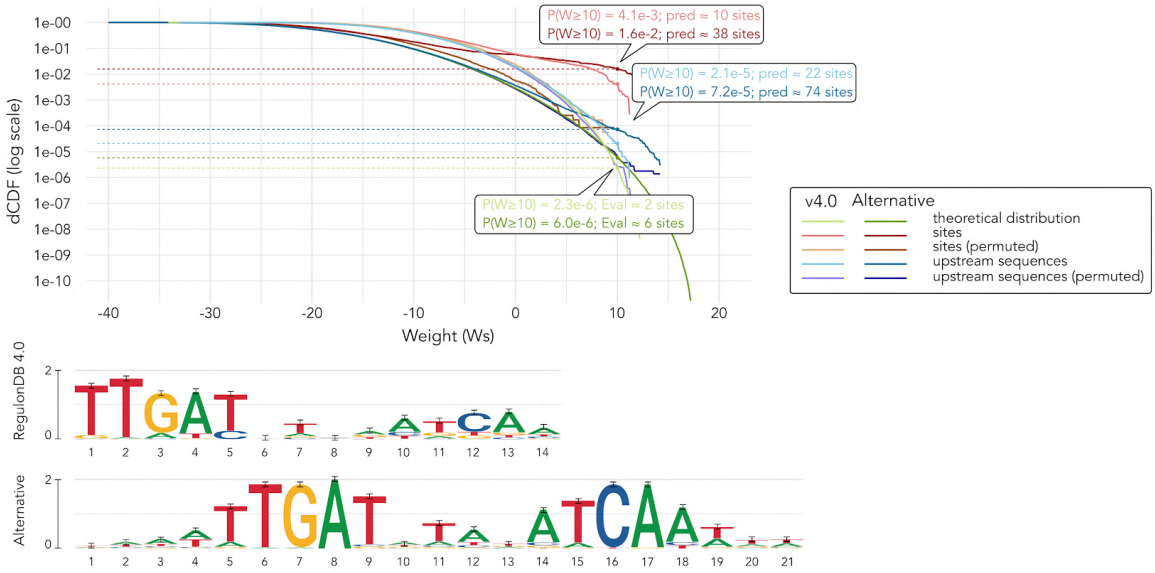
$$W_S = log\left(\frac{P(S/M)}{P(S/B)}\right)$$

Where:

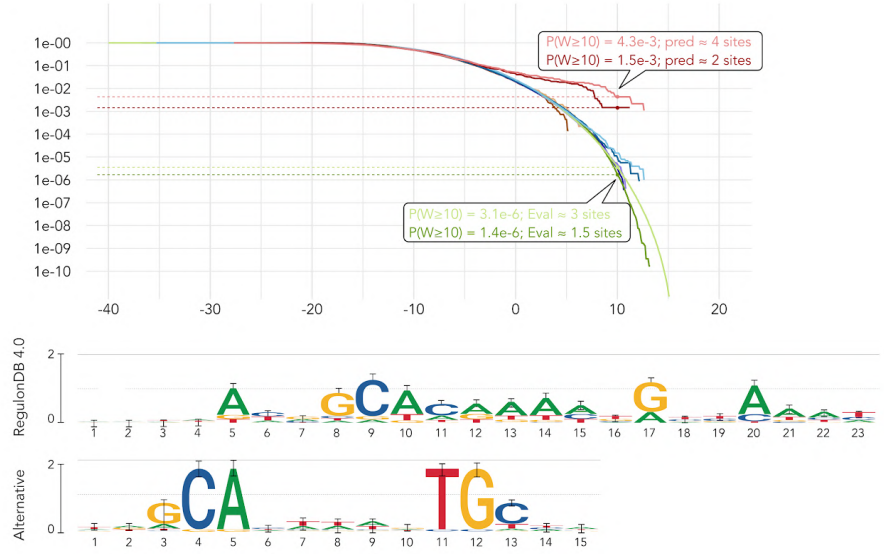
- W_S = weight score of sequence segment S
- $P(S/M)$ = probability of sequence S according to motif M
- $P(S/B)$ = probability of sequence S according to background B

The weight score W_S is also prone to inaccuracy, for it depends on the matrix length and IC, which is why matrix-quality combines it with theoretical and empirical score distributions. Theoretical distributions of scores generated given a specific PSSM allow estimating the p-value associated with a given sequence and its weight score (as computed as shown in the above formula), taking into account the genomic background. Empirical distributions of scores computed from collections of sequences such as TFBSs should significantly diverge from the theoretical distributions if the PSSM used is specific enough of the TF considered. This can be visualized using a decreasing cumulative distribution function (dCDF), which depicts the probability (ordinate) to obtain randomly a weight score W_S higher than, or equal to a given W_S value (abscissa) (Figure 25).





a. FNR - total sites = 91



b. MarA - total sites = 28

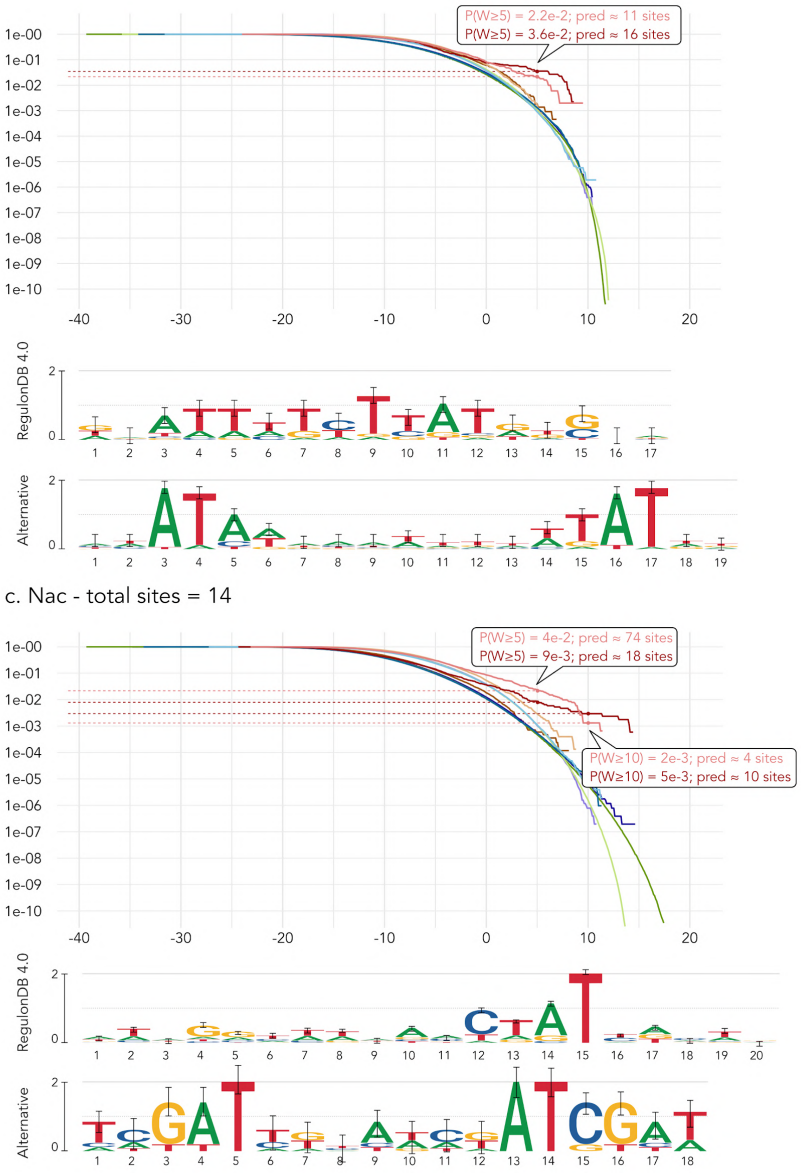


Figure 25. Theoretical and empirical probability distributions using matrix-quality. Distributions were generated using FNR, MarA, Nac and OxyR PSSMs from RegulonDB 4.0 and the alternative collection. **a. FNR.** Green curves represent the probabilities of observing by chance a sequence of score equal to or higher that *Ws* given the RegulonDB PSSM (light green) and the alternative PSSM (dark green), computed for all non-coding upstream regions of *E. coli* K-12. For instance, observing by chance a sequence scoring higher than 10 has a low probability of 2.3e-6 considering the RegulonDB PSSM, however, taking into account the multiple testing of all possible positions from the upstream regions, it is associated with an e-value of 2 false positives (and in the case of the alternative PSSM, 6 false positives). Overall, the distribution from the alternative PSSM includes a larger range of possible scores, consistent with the fact that it has a higher IC (Table 5), but a lower FDR for scores lower than 10. Blue curves represent the observed distribution of scores in all non-coding upstream sequences. It shows that 22 sites scoring

higher than 10 can be predicted with RegulonDB PSSM, and 74 sites using the alternative PSSM. The difference observed compared to the theoretical distribution as well as a permuted set of upstream sequences (purple curves) demonstrate the relevance of the PSSM in predicting potential TFBSs. Finally, the red curves show the distribution of scores observed in FNR TFBS sequences from RegulonDB. They are both well-above the theoretical distributions and the upstream sequences distributions, as well as the binding sites permuted sequences (orange curves), denoting the specificity of the PSSM for FNR binding sites. **b. MarA.** The alternative PSSM predicts fewer sites of score higher than 10 compared to the v4.0 PSSM, consistent with the theoretical distributions and the fact that the new matrix is smaller, and has a lower total IC. **c. Nac.** The alternative matrix predicts more sites of score higher than 5 than the RegulonDB matrix, while having a very similar background distribution (p-val = 1.1e-3; e-val = 0.5 sites). Both PSSMs have a rather low IC and fail at predicting sites of scores higher than 10. **d. OxyR.** The RegulonDB PSSM predicts more sites than the alternative one considering a score threshold at 5, but the tendency is reverted around scores of 10 and above. The predictive capacity of the RegulonDB matrix (pale red) drops significatively, while that of the alternative matrix is maintained.

Overall, the results produced by matrix-quality for the 92 TF compared show disparities. Some of the alternative matrices predict more high-scoring sites, congruent with the fact that they generally have a higher information content. They may be associated with higher FDR around high scores, but generally have a lower FDR than the RegulonDB collection when considering lower scores. High-scoring matrices are usually better at predicting high-scoring sites, but may filter out more false negative sites of lower scores. Yet, those binding sites of low sequence conservation can be just as relevant to regulation, and can even be necessary for some regulatory mechanisms that rely on TF cooperation (Oliver et al., 2016). For this reason, it may be relevant to have TFs associated with several alternative PSSMs allowing one to fulfill distinct purposes, from visualization and TFBS prediction to TFs classification.

Classification of transcription factor motifs

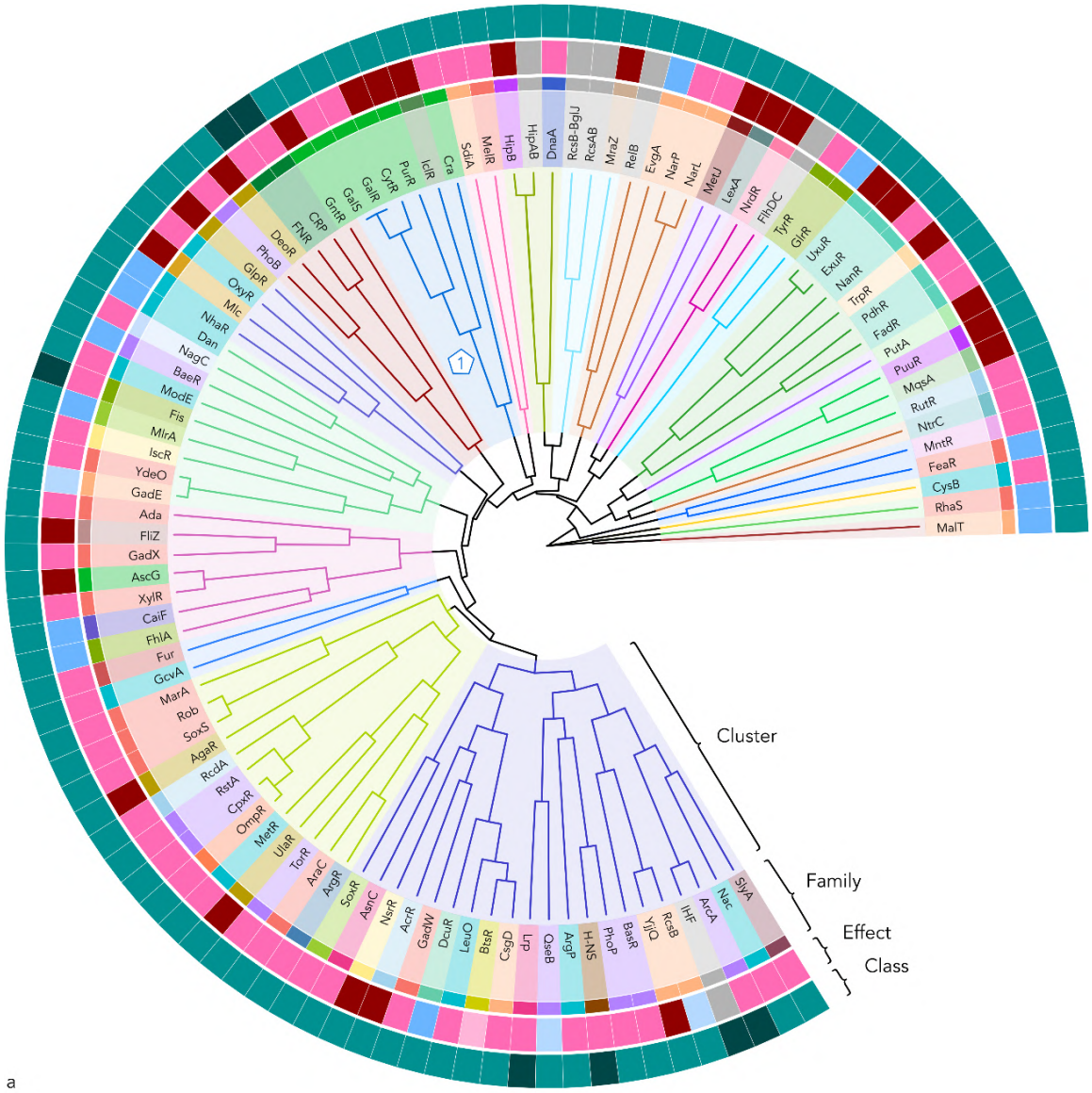
Matrices can be used to classify TFs based on their motif similarity. Matrix-clustering (Castro-Mondragón et al., 2017) is a tool that clusters similar transcription factor binding motifs by computing a matrix of similarity between all pairs of input matrices, and performing hierarchical clustering to build a motif tree. The tree can then be partitioned into clusters, using a variety of similarity metrics.

I performed the clustering of 101 PSSM from the alternative collection using the normalized correlation coefficient Ncor in order to compute pairwise similarity between all PSSMs, with the average-linkage method. Since some matrices lack precision, I used a low threshold for the tree-partitioning step (Ncor = 0.3). A total of

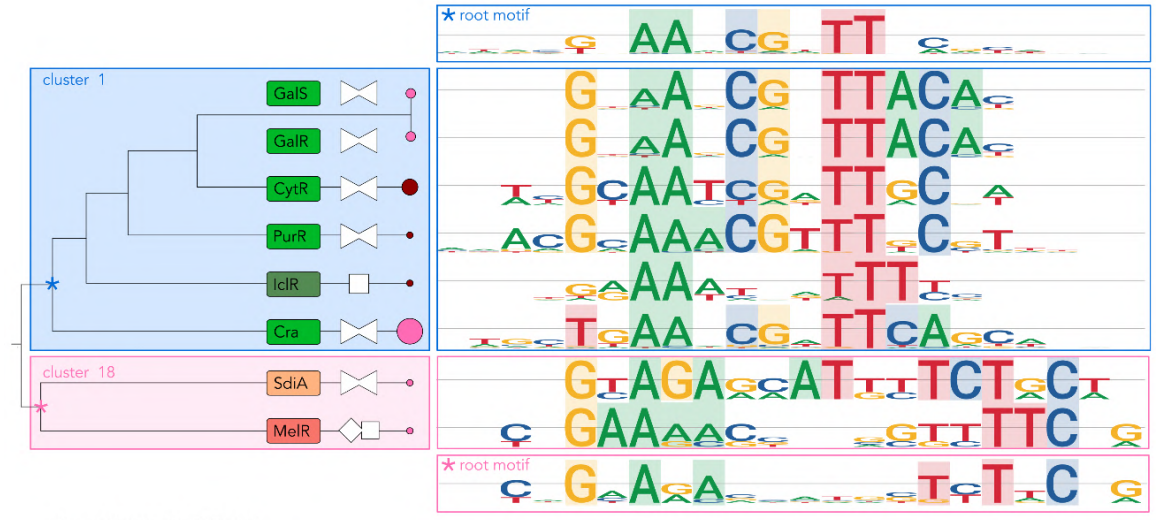
23 clusters were generated (Table 6). Their sizes range from 1, meaning a given PSSM did not cluster at all (clusters 19 to 23: CysB, MalT, NtrC, PutA, RhaS) to 21 matrices (cluster 7).

The complete tree is presented in Figure 26a, alongside TF family information, their known effects, and their reported class (local or global regulator). Interestingly, some of the largest families seem to be clustered together while others are more scattered. Of the 101 PSSMs in the collection, 10 are associated with TFs from the prominent LysR family, involved in amino acid synthesis and evolutionary related (Pérez-Rueda et al., 2015). However, those 10 motifs are part of 6 different clusters. Another major family described in *E. coli* is AraC/XylS, involved in carbon source assimilation, which accounts for 12 matrices in this collection. These are scattered between 7 different clusters, but 4 of them are grouped in the same cluster (cluster 4). In particular, three TFs share a close motif similarity: MarA, Rob and SoxS. Those TFs are known to form part of a regulon involved in antibiotics and superoxide resistance (Pérez-Rueda et al., 2015). In the same cluster, another 3 TFs are also closely related together: OmpR, CpxR and RstA, from the OmpR family, involved in particular in biofilm formation and response to acidic stress (Ogasawara et al., 2010; Aquino et al., 2017). Among the families that show a more consistent clustering, we can cite GntR, and GalR/LacI. All 5 PSSMs from TFs that are part of the GntR family were grouped together in cluster 3 with only one outsider, TrpR. As for the GalR/LacI family, 5 PSSMs out of 7 are clustered together in cluster 1, of which a detailed view is shown in Figure 26b. The only TF in the cluster that is not part of the GalR/LacI family, specialized in sugar metabolism, is a repressor of the glyoxylate bypass operon, Iclr.

Overall, TFs that are part of the same evolutionary families do not significantly cluster together despite a low clustering threshold. Still, many TFs remain poorly characterized and their motifs lack precision, which explains why their clustering remains difficult. Several global regulators are found in the same cluster (ArcA, H-NS, IHF, Lrp). A similar tendency was reported in a previous study where TFs were clustered based on corregulation (Pérez-Rueda et al., 2015), but the opposite behavior was observed when studying topological modules of the *E. coli* TRN (Resendis-Antonio et al., 2005). However, it is difficult to draw conclusions, as global regulators tend to have rather degenerated motifs despite having numerous known binding sites.



a



b

Reported symmetry of TF motif

Inverted repeat

Direct repeat

Asymmetric

Unspecified

Number of reported targets of TF

50

10

1

Horizontal grey lines accross the motifs represent the 1-bit ordinate

Nucleotides that account for at least 1 bit are emphasized with colored background

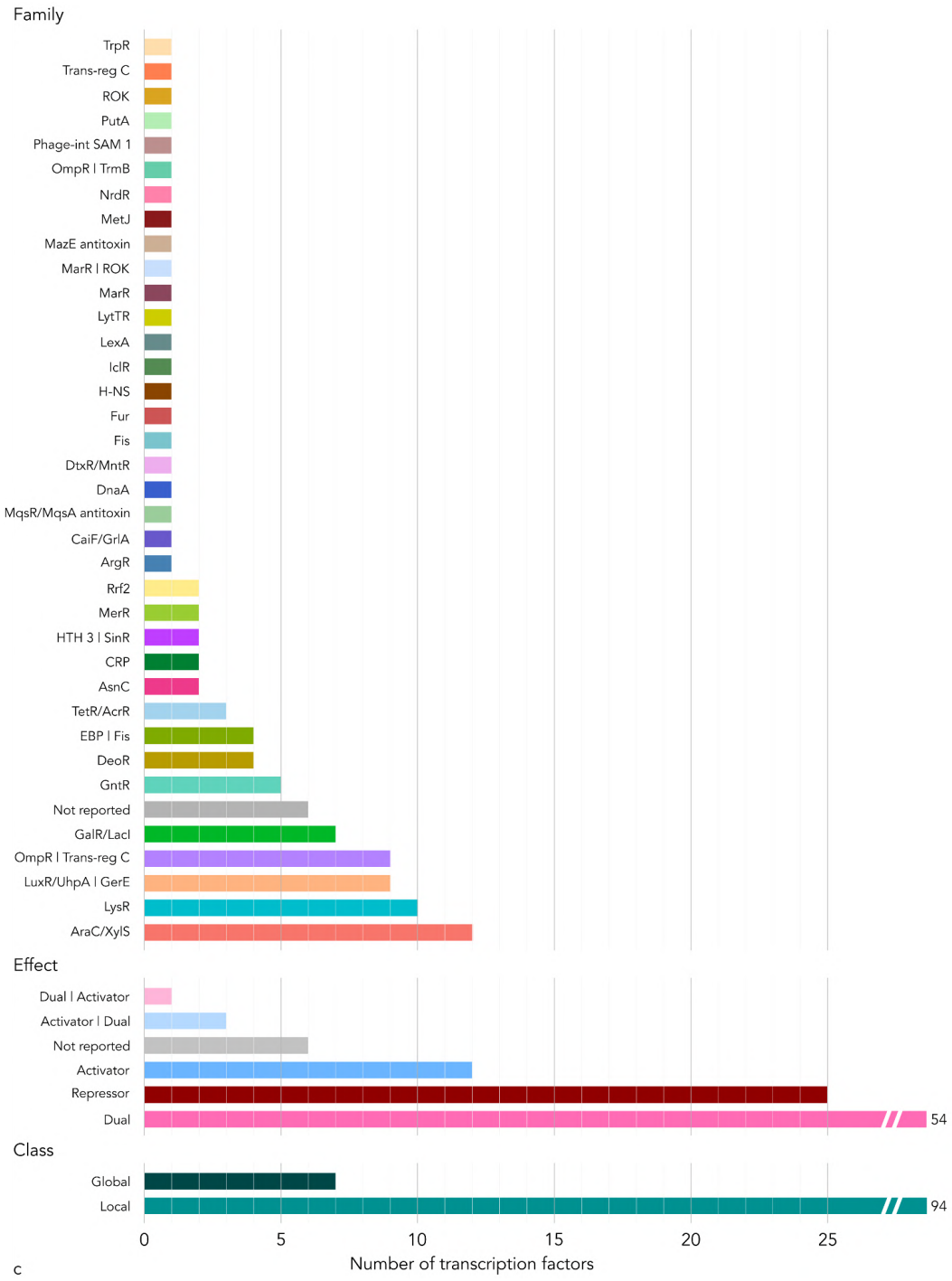


Figure 26. Clustering of the 101 PSSM from the alternative collection. a. The complete tree of clustered matrices and associated parameters for 101 TFs. Proportionate tree branches were manually twitched to enhance readability. **b.** Detailed subtree for cluster 1 and its closest relative. **c.** Color legend and associated TF numbers. TF family and effect annotations were retrieved from RegulonDB and in some cases, completed with recent annotations from Flores-Bautista et al., 2020 (annotations separated by a pipe "|").

Cluster	TF name	Cluster size
1	Cra,CytR,GalR,GalS,IclR,PurR	6
2	DnaA,HipAB,HipB	3
3	ExuR,FadR,NanR,PdhR,TrpR,UxuR	6
4	AgaR,AraC,ArgR,CpxR,MarA,MetR,OmpR,RcdA,Rob,RstA,SoxS,TorR,UlaR	13
5	EvgA,NarL,NarP,RelB-RelE	4
6	BaeR,Dan,Fis,GadE,IscR,MlrA,ModE,NagC,YdeO	9
7	AcrR,ArcA,ArgP,AsnC,BasR,BtsR,CsgD,DcuR,GadW,H-NS,IHF,LeuO,Lrp,Nac, NsrR,PhoP,QseB,RcsB,SlyA,SoxR,YjjQ	21
8	CRP,DeoR,FNR,GntR,PhoB	5
9	Ada,AscG,CaiF,FhlA,FliZ,GadX,XylR	7
10	MqsA,PuuR,RutR	3
11	MraZ,RcsAB,RcsB-BglJ	3
12	GlpR,Mlc,NhaR,OxyR	4
13	GlrR,TyrR	2
14	FlhDC,NrdR	2
15	LexA,MetJ	2
16	Fur,GcvA	2
17	FeaR,MntR	2
18	MelR,SdiA	2
19	PutA	1
20	NtrC	1
21	MalT	1
22	RhaS	1
23	CysB	1

Table 6. The complete list of 23 clusters for 101 PSSMs.

ChIP-seq based motifs

As mentioned, the main issue for PSSM construction is a lack of binding data. Binding datasets from genome-wide experiments are very helpful in that regard. Using 28 ChIP-seq datasets (Chapter 4), I built matrices targeting 11 TFs (Table 7). The motifs generally show a better accuracy than the alternative collection, but many fail at detecting dyads, particularly for TFs that have a lot of binding sites (Figure 27). Indeed, many TFBSs are poorly conserved that end up "diluting" the pattern, although their regulatory role is relevant under specific conditions (Oliver et al., 2016). It is also reasonable to assume that the set of binding sites previously identified and based on low-throughput experiments actually represent a very small proportion of the actual binding sites in *E. coli*, and could be biased towards better conserved sequences.

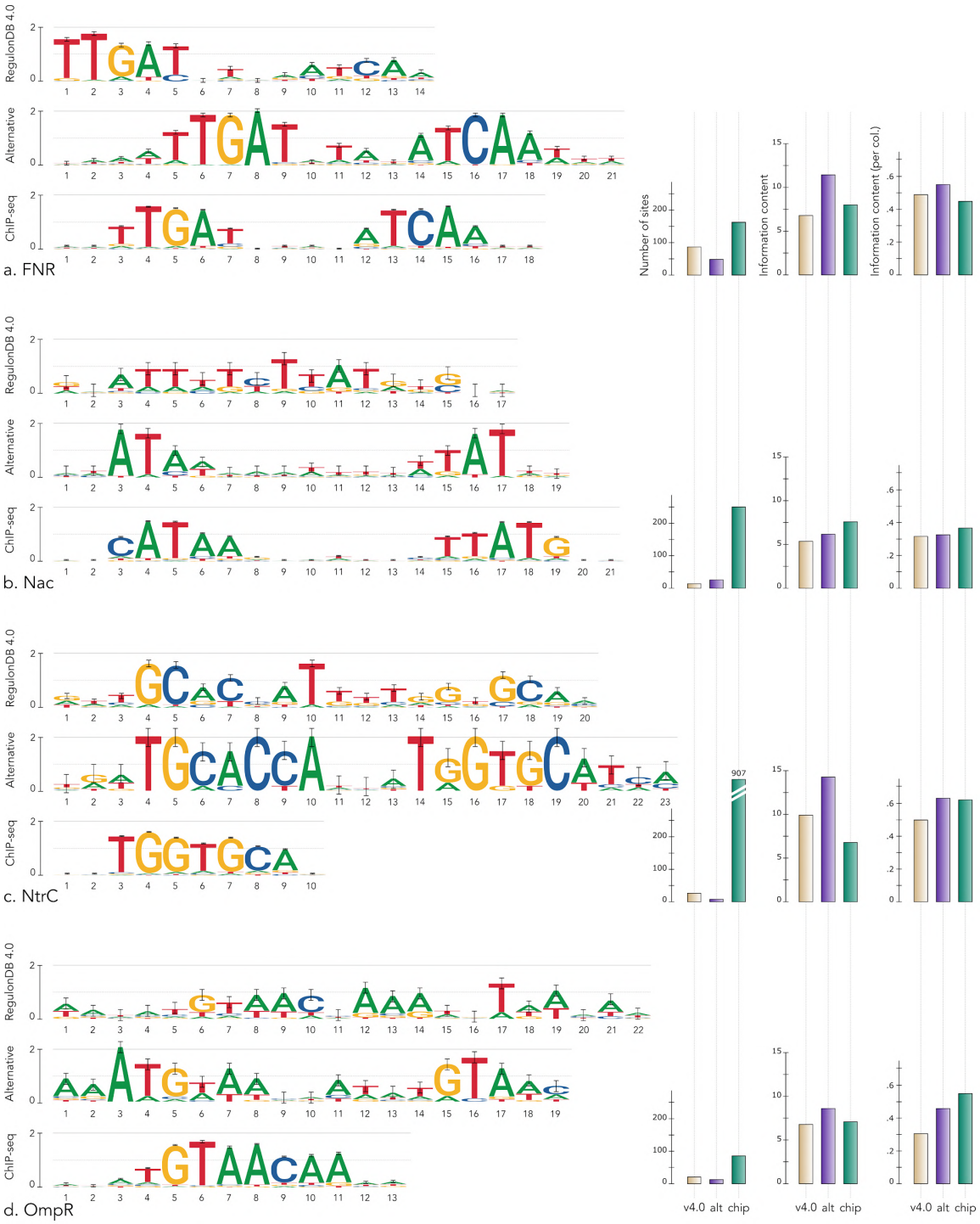


Figure 27. ChIP-seq motifs. **a. FNR.** The ChIP-seq motif is shorter and shows a very conserved dyadic pattern. **b. Nac.** The ChIP-seq dataset provides a 10-fold pool of binding sequences to build on, however, the final motif is very similar to the alternative motif, with the addition of 2 significant nucleotides giving it a higher specificity. **c. NtrC.** Two datasets were combined, totaling a number of 907 sequences. The motif produced has a high resolution, but only a half dyad is conserved. **d. OmpR.** Similar to NtrC, the ChIP-seq based motif for OmpR is well-defined but only includes half of its known motif.

	RegulonDB v4.0			ChIP-seq matrices			
TF	**Sites**	**IC (avg)**	**Consensus**	**Peaks**	**Sites**	**IC (avg)**	**Consensus**
Fis	219	4.5 (0.26)	GbyyrwtttttvasCra	1301	322	8.5 (0.57)	bstTGCTGGCGatsk
FlhDC	16	8.8 (0.46)	aAwsGsskGAwtwrGsGsc	47			
FNR	88	6.7 (0.48)	TTGAtrtwratCaa	157	165	7.9 (0.44)	wwtTGAtstasaTCAaww
Fur	48	9.8 (0.44)	tRAtAAtsaTtmtCAtTwbcaw	1746	1268	6.4 (0.53)	grATGATAAtsa
GlaR				599	322	7.5 (0.63)	raAATGGCGAyr
H-NS				2549			
Lrp	80	4.1 (0.31)	kmwtwttwtyCtK	5167	1489	7.2 (0.72)	awTATTCTgc
Nac	14	5.3 (0.31)	krattykyTyatrkssr	499	254	7.5 (0.36)	kmCATAagmawtkcttATGkm
NtrC	27	9.8 (0.49)	rwtGCaCsaTkktgGkGCam	845	907	6.7 (0.61)	gsTGGTGCAss
OmpR	21	6.7 (0.3)	wayatGtaaCcaarwgtwwmaw	136	87	6.9 (0.53)	ytTTGTTACatrt
PhoB	26	6.7 (0.33)	wtrtkaCAkhttTrtgwcAg	121	107	7.4 (0.62)	mytTGTCATatk

Table 7. Summary of the matrices built from ChIP-seq datasets and their original version in RegulonDB.

The analysis of ChIP-seq data shows that, as expected, many more binding sites are discovered than what is currently described in the literature. Most of this curated knowledge stems from *in vitro*, low-throughput experiments, while high-throughput experiments like ChIP-seq, allowing a genome-wide characterization, has barely been applied to *Escherichia coli* K-12, despite being a widely-studied model organism.

As observed in the alternative collection of matrices, the ChIP-seq-based matrix collection demonstrates the variability observed among transcription factor binding sites. Although TFBS are generally evolutionary conserved, they show a diversity of profiles that can be equally relevant to transcriptional regulation. Besides, some binding sites have been observed that displayed a spacing distinct from the expected one, although it is supposed to be a conserved, TF-specific characteristic. This further supports the idea that TFs could be associated with a set of alternative matrices rather than a single one.

Discussion

Results

During this PhD, I worked towards the goal of exhaustively characterizing *Escherichia coli* K-12's regulatory networks. I started by tackling several facets of this challenge separately.

Upon manipulating a variety of data from different sources, I quickly noticed how a lack of congruence in such basic information as gene names and coordinates was going to be a recurrent bottleneck. This triggered the EcoliGenes project, a software library which I then used in most of my subsequent works, and kept developing and updating to fit the needs of my goals.

Concomitantly, I built an exhaustive set of *E. coli* genomic features by combining long-established data from the literature and numerous datasets generated through next-generation sequencing technologies and published in recent years. I processed the data so as to homogenize their respective formats, and formally defined different types of objects to fit a common framework.

In order to integrate binding and expression data I developed SnakeChunks, a library of workflows and rules based on snakemake. These workflows allow automated analyses of ChIP-seq and RNA-seq data, from raw samples to final results such as transcription factor binding sites, motifs, and differentially-expressed genes. This work culminated in the publication of a protocol (Rioualen et al., 2019).

I used these founding elements in order to pursue my main goal, the characterization of the transcriptional regulatory network of *Escherichia coli* K-12. Together with the team from the Program of Computational Genomics and our collaborators from Boston University and the Wadsworth Center at SUNY Albany, we conceived a new framework to integrate thousands of high-throughput datasets in RegulonDB, by articulating

together three facets of the project and the respective fields of expertise of our team: (i) the gathering and curation of relevant datasets, led by biocurators; (ii) the standardization and/or processing of the data, led by bioinformaticians; and (iii) the integration and visual display of the results via the RegulonDB HT portal, led and realized by the computational team.

Finally, I investigated an alternative strategy to generate a collection of transcription factor binding matrices by using pattern discovery approaches. While it produced high-resolution motifs, it also raised a thought as to the relevance of keeping several alternative matrices for certain transcription factors that display a variety of binding profiles.

Conclusion

Escherichia coli K-12, despite being the single best-characterized organism on Earth, still offers mysteries to solve. While its genome is relatively small and its genes count "only" in the thousands, it has very complex and ramified regulatory networks, from signaling pathways to metabolic reactions. The transcriptional regulatory network is key to articulating and coordinating cellular responses to environmental stimuli. Numerous promoters and terminators offer endless possibilities of transcription initiation and termination, finely regulated through external signals triggered by growth conditions, and subsequent activation or repression of gene expression by transcription factors.

Biological paradigms are permanently challenged by the ever increasing amount of knowledge acquired, and an *exhaustive* characterization of the regulatory networks of *E. coli* K-12 may never actually be achieved. However, through this PhD project I was able to contribute to this ambitious perspective in a significant way, by gathering and formatting numerous high-throughput datasets, developing tools and workflows for their reproducible analysis and integration with classic knowledge, and generating TF binding motifs with a higher resolution.

Perspectives

The transcriptional regulatory network is a key component of *Escherichia coli*'s regulatory circuits, for it coordinates metabolic responses in the cell upon sensing

intra- and extracellular signals, offering an extremely high adaptability to environmental changes. Expanding the known TRN from *E. coli* opens a gate to better understanding its biology, but also that of other species. Being a model organism, *E. coli* has consistently been used to investigate and describe fundamental biological mechanisms that could be applied to other organisms later on, as well as identify genes, proteins and other features from related bacteria by homology.

In particular, it can help greatly to uncover the transcriptional regulatory network of one of its close relatives: *Salmonella enterica*. Both species have very similar genomes and lifestyles, but the latter is pathogenic, while *E. coli* is mostly a commensal bacteria. *S. enterica* is commonly studied by scientists, but its TRN is much less known than that of *E. coli*.

During my PhD, I had the opportunity of taking part in a project which aims at characterizing the *S. enterica* regulatory network by taking advantage of the knowledge acquired of the *E. coli* network, and combining it with computational approaches. This strategy offers the perspective of gathering significant amounts of regulatory data for *Salmonella* as well as other bacteria, in a much more efficient way than before. By combining once again the expertise of biocurators and computational scientists from the PGC, we are hoping to expand RegulonDB to cover multiple organisms, and make the decades of manual curation of a single organism performed in the past become years of combined approaches to characterize multiple organisms.

References

Aquino, P., Honda, B., Jaini, S., Lyubetskaya, A., Hosur, K., Chiu, J. G., Ekladious, I., Hu, D., Jin, L., Sayeg, M. K., Stettner, A. I., Wang, J., Wong, B. G., Wong, W. S., Alexander, S. L., Ba, C., Bensussen, S. I., Bernstein, D. B., Braff, D., … Galagan, J. E. (2017). Coordinated regulation of acid resistance in Escherichia coli. BMC Systems Biology, 11(1), 1. https://doi.org/10.1186/s12918-016-0376-y

Bailey, T. L., Johnson, J., Grant, C. E., & Noble, W. S. (2015). The MEME Suite. Nucleic Acids Research, 43(W1), W39–W49. https://doi.org/10.1093/nar/gkv416

Blattner, F. R., Plunkett, G., Bloch, C. A., Perna, N. T., Burland, V., Riley, M., Collado-Vides, J., Glasner, J. D., Rode, C. K., Mayhew, G. F., Gregor, J., Davis, N. W., Kirkpatrick, H. A., Goeden, M. A., Rose, D. J., Mau, B., & Shao, Y. (1997). The Complete Genome Sequence of Escherichia coli K-12. Science, 277(5331), 1453–1462. https://doi.org/10.1126/science.277.5331.1453

Buck, M. J., & Lieb, J. D. (2004). ChIP-chip: Considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. Genomics, 83(3), 349–360. https://doi.org/10.1016/j.ygeno.2003.11.004

Castro-Mondragon, J. A., Jaeger, S., Thieffry, D., Thomas-Chollier, M., & van Helden, J. (2017). RSAT matrix-clustering: Dynamic exploration and redundancy reduction of transcription factor binding motif collections. Nucleic Acids Research, 45(13), e119. https://doi.org/10.1093/nar/gkx314

Cho, B.-K., Kim, D., Knight, E. M., Zengler, K., & Palsson, B. O. (2014). Genome-scale reconstruction of the sigma factor network in Escherichia coli: Topology and functional states. BMC Biology, 12, 4. https://doi.org/10.1186/1741-7007-12-4

Cho, B.-K., Zengler, K., Qiu, Y., Park, Y. S., Knight, E. M., Barrett, C. L., Gao, Y., & Palsson, B. Ø. (2009). The transcription unit architecture of the Escherichia coli genome. Nature Biotechnology, 27(11), 1043–1049. https://doi.org/10.1038/nbt.1582

Choudhary, S. (2019). pysradb: A Python package to query next-generation sequencing metadata and data from NCBI Sequence Read Archive (8:532). F1000Research. https://doi.org/10.12688/f1000research.18676.1

Conway, T., Creecy, J. P., Maddox, S. M., Grissom, J. E., Conkle, T. L., Shadid, T. M., Teramoto, J., San Miguel, P., Shimada, T., Ishihama, A., Mori, H., & Wanner, B. L. (2014). Unprecedented high-resolution view of bacterial operon architecture revealed by RNA sequencing. MBio, 5(4), e01442-01414. https://doi.org/10.1128/mBio.01442-14

Desvillechabrol, D., Legendre, R., Rioualen, C., Bouchier, C., van Helden, J., Kennedy, S., & Cokelaer, T. (2018). Sequanix: A dynamic graphical interface for Snakemake workflows. Bioinformatics (Oxford, England), 34(11), 1934–1936. https://doi.org/10.1093/bioinformatics/bty034

Ewels, P., Magnusson, M., Lundin, S., & Käller, M. (2016). MultiQC: Summarize analysis results for multiple tools and samples in a single report. Bioinformatics (Oxford, England), 32(19), 3047–3048. https://doi.org/10.1093/bioinformatics/btw354

Feng, J., Liu, T., & Zhang, Y. (2011). Using MACS to identify peaks from ChIP-Seq data. Current Protocols in Bioinformatics, Chapter 2, Unit 2.14. https://doi.org/10.1002/0471250953.bi0214s34

Flores-Bautista, E., Cronick, C. L., Fersaca, A. R., Martinez-Nuñez, M. A., & Pérez-Rueda, E. (2018). Functional Prediction of Hypothetical Transcription Factors of Escherichia coli K-12 Based on Expression Data. Computational and Structural Biotechnology Journal, 16, 157–166. https://doi.org/10.1016/j.csbj.2018.03.003

Flores-Bautista, E., Hernandez-Guerrero, R., Huerta-Saquero, A., Tenorio-Salgado, S., Rivera-Gomez, N., Romero, A., Ibarra, J. A., & Pérez-Rueda, E. (2020). Deciphering the functional diversity of DNA-binding transcription factors in Bacteria and Archaea organisms. PloS One, 15(8), e0237135. https://doi.org/10.1371/journal.pone.0237135

Galas, D. J., & Schmitz, A. (1978). DNAse footprinting: A simple method for the detection of protein-DNA binding specificity. Nucleic Acids Research, 5(9), 3157–3170. https://doi.org/10.1093/nar/5.9.3157

Gálvez-Merchán, Á., Min, K. H. (Joseph), Pachter, L., & Booeshaghi, A. S. (2022). Metadata retrieval from sequence databases with ffq (p. 2022.05.18.492548). bioRxiv. https://doi.org/10.1101/2022.05.18.492548

Gao, Y., Yurkovich, J. T., Seo, S. W., Kabimoldayev, I., Dräger, A., Chen, K., Sastry, A. V., Fang, X., Mih, N., Yang, L., Eichner, J., Cho, B.-K., Kim, D., & Palsson, B. O. (2018). Systematic discovery of uncharacterized transcription factors in Escherichia coli K-12 MG1655. Nucleic Acids Research, 46(20), 10682–10696. https://doi.org/10.1093/nar/gky752

Garner, M. M., & Revzin, A. (1981). A gel electrophoresis method for quantifying the binding of proteins to specific DNA regions: Application to components of the Escherichia coli lactose operon regulatory system. Nucleic Acids Research, 9(13), 3047–3060.

Ishihama, A., Shimada, T., & Yamazaki, Y. (2016). Transcription profile of Escherichia coli: Genomic SELEX search for regulatory targets of transcription factors. Nucleic Acids Research, 44(5), 2058–2074. https://doi.org/10.1093/nar/gkw051

Jacob, F., & Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. Journal of Molecular Biology, 3(3), 318–356. https://doi.org/10.1016/S0022-2836(61)80072-7

Johnson, D. S., Mortazavi, A., Myers, R. M., & Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. Science (New York, N.Y.), 316(5830), 1497–1502. https://doi.org/10.1126/science.1141319

Ju, X., Li, D., & Liu, S. (2019). Full-length RNA profiling reveals pervasive bidirectional transcription terminators in bacteria. Nature Microbiology, 4(11), 1907–1918. https://doi.org/10.1038/s41564-019-0500-z

Keseler, I. M., Gama-Castro, S., Mackie, A., Billington, R., Bonavides-Martínez, C., Caspi, R., Kothari, A., Krummenacker, M., Midford, P. E., Muñiz-Rascado, L., Ong, W.

K., Paley, S., Santos-Zavaleta, A., Subhraveti, P., Tierrafría, V. H., Wolfe, A. J., Collado-Vides, J., Paulsen, I. T., & Karp, P. D. (2021). The EcoCyc Database in 2021. Frontiers in Microbiology, 12, 711077. https://doi.org/10.3389/fmicb.2021.711077

Kim, G. B., Gao, Y., Palsson, B. O., & Lee, S. Y. (2021). DeepTFactor: A deep learning-based tool for the prediction of transcription factors. Proceedings of the National Academy of Sciences, 118(2), e2021171118. https://doi.org/10.1073/pnas.2021171118

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. Nature Methods, 9(4), 357–359. https://doi.org/10.1038/nmeth.1923

Lederberg, J., & Tatum, E. L. (1946). Gene Recombination in Escherichia Coli. Nature, 158(4016), 558–558. https://doi.org/10.1038/158558a0

Ledezma-Tejeida, D., Ishida, C., & Collado-Vides, J. (2017). Genome-Wide Mapping of Transcriptional Regulation and Metabolism Describes Information-Processing Units in Escherichia coli. Frontiers in Microbiology, 8. https://doi.org/10.3389/fmicb.2017.01466

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.Journal, 17(1), 10–12. https://doi.org/10.14806/ej.17.1.200

Medina-Rivera, A., Abreu-Goodger, C., Thomas-Chollier, M., Salgado, H., Collado-Vides, J., & van Helden, J. (2011). Theoretical and empirical quality assessment of transcription factor-binding motifs. Nucleic Acids Research, 39(3), 808–824. https://doi.org/10.1093/nar/gkq710

Mejía-Almonte, C., Busby, S. J. W., Wade, J. T., van Helden, J., Arkin, A. P., Stormo, G. D., Eilbeck, K., Palsson, B. O., Galagan, J. E., & Collado-Vides, J. (2020). Redefining fundamental concepts of transcription initiation in bacteria. Nature Reviews. Genetics, 21(11), 699–714. https://doi.org/10.1038/s41576-020-0254-8

Mendoza-Vargas, A., Olvera, L., Olvera, M., Grande, R., Vega-Alvarado, L., Taboada, B., Jimenez-Jacinto, V., Salgado, H., Juárez, K., Contreras-Moreira, B., Huerta, A. M., Collado-Vides, J., & Morett, E. (2009). Genome-Wide Identification of Transcription

Start Sites, Promoters and Transcription Factor Binding Sites in E. coli. PLOS ONE, 4(10), e7526. https://doi.org/10.1371/journal.pone.0007526

Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., & Köster, J. (2021). Sustainable data analysis with Snakemake. F1000Research, 10, 33. https://doi.org/10.12688/f1000research.29032.2

Moreno-Hagelsieb, G., & Collado-Vides, J. (2002). A powerful non-homology method for the prediction of operons in prokaryotes. Bioinformatics, 18(suppl_1), S329–S336. https://doi.org/10.1093/bioinformatics/18.suppl_1.S329

Moretto, M., Sonego, P., Dierckxsens, N., Brilli, M., Bianco, L., Ledezma-Tejeida, D., Gama-Castro, S., Galardini, M., Romualdi, C., Laukens, K., Collado-Vides, J., Meysman, P., & Engelen, K. (2016). COLOMBOS v3.0: Leveraging gene expression compendia for cross-species analyses. Nucleic Acids Research, 44(Database issue), D620–D623. https://doi.org/10.1093/nar/gkv1251

Ogasawara, H., Yamada, K., Kori, A., Yamamoto, K., & Ishihama, A. 2010. (n.d.). Regulation of the Escherichia coli csgD promoter: Interplay between five transcription factors. Microbiology, 156(8), 2470–2483. https://doi.org/10.1099/mic.0.039131-0

Oliver, P., Peralta-Gil, M., Tabche, M.-L., & Merino, E. (2016). Molecular and structural considerations of TF-DNA binding for the generation of biologically meaningful and accurate phylogenetic footprinting analysis: The LysR-type transcriptional regulator family as a study model. BMC Genomics, 17, 686. https://doi.org/10.1186/s12864-016-3025-3

O'Malley, R. C., Huang, S.-S. C., Song, L., Lewsey, M. G., Bartlett, A., Nery, J. R., Galli, M., Gallavotti, A., & Ecker, J. R. (2016). Cistrome and Epicistrome Features Shape the Regulatory DNA Landscape. Cell, 165(5), 1280–1292. https://doi.org/10.1016/j.cell.2016.04.038

Pérez-Rueda, E., & Collado-Vides, J. (2000). The repertoire of DNA-binding transcriptional regulators in Escherichia coli K-12. Nucleic Acids Research, 28(8), 1838–1847. https://doi.org/10.1093/nar/28.8.1838

Pérez-Rueda, E., Collado-Vides, J., & Segovia, L. (2004). Phylogenetic distribution of DNA-binding transcription factors in bacteria and archaea. Computational Biology and Chemistry, 28(5), 341–350. https://doi.org/10.1016/j.compbiolchem.2004.09.004

Pérez-Rueda, E., Hernandez-Guerrero, R., Martinez-Nuñez, M. A., Armenta-Medina, D., Sanchez, I., & Ibarra, J. A. (2018). Abundance, diversity and domain architecture variability in prokaryotic DNA-binding transcription factors. PloS One, 13(4), e0195332. https://doi.org/10.1371/journal.pone.0195332

Pérez-Rueda, E., Tenorio-Salgado, S., Huerta-Saquero, A., Balderas-Martínez, Y. I., & Moreno-Hagelsieb, G. (2015). The functional landscape bound to the transcription factors of Escherichia coli K-12. Computational Biology and Chemistry, 58, 93–103. https://doi.org/10.1016/j.compbiolchem.2015.06.002

Resendis-Antonio, O., Freyre-González, J. A., Menchaca-Méndez, R., Gutiérrez-Ríos, R. M., Martínez-Antonio, A., Ávila-Sánchez, C., & Collado-Vides, J. (2005). Modular analysis of the transcriptional regulatory network of E. coli. Trends in Genetics, 21(1), 16–20. https://doi.org/10.1016/j.tig.2004.11.010

Rhee, H. S., & Pugh, B. F. (2011). Comprehensive genome-wide protein-DNA interactions detected at single-nucleotide resolution. Cell, 147(6), 1408–1419. https://doi.org/10.1016/j.cell.2011.11.013

Rioualen, C., Charbonnier-Khamvongsa, L., Collado-Vides, J., & van Helden, J. (2019). Integrating Bacterial ChIP-seq and RNA-seq Data With SnakeChunks. Current Protocols in Bioinformatics, 66(1), e72. https://doi.org/10.1002/cpbi.72

Salgado, H., Moreno-Hagelsieb, G., Smith, T. F., & Collado-Vides, J. (2000). Operons in Escherichia coli: Genomic analyses and predictions. Proceedings of the National Academy of Sciences of the United States of America, 97(12), 6652–6657. https://doi.org/10.1073/pnas.110147297

Salgado, H., Peralta-Gil, M., Gama-Castro, S., Santos-Zavaleta, A., Muñiz-Rascado, L., García-Sotelo, J. S., Weiss, V., Solano-Lira, H., Martínez-Flores, I., Medina-Rivera, A., Salgado-Osorio, G., Alquicira-Hernández, S., Alquicira-Hernández, K., López-Fuentes, A., Porrón-Sotelo, L., Huerta, A. M., Bonavides-Martínez, C., Balderas-Martínez, Y. I., Pannier, L., … Collado-Vides, J. (2013). RegulonDB v8.0: Omics data sets, evolutionary

conservation, regulatory phrases, cross-validated gold standards and more. Nucleic Acids Research, 41(Database issue), D203-213. https://doi.org/10.1093/nar/gks1201

Santana-Garcia, W., Castro-Mondragon, J. A., Padilla-Gálvez, M., Nguyen, N. T. T., Elizondo-Salas, A., Ksouri, N., Gerbes, F., Thieffry, D., Vincens, P., Contreras-Moreira, B., van Helden, J., Thomas-Chollier, M., & Medina-Rivera, A. (2022). RSAT 2022: Regulatory sequence analysis tools. Nucleic Acids Research, gkac312. https://doi.org/10.1093/nar/gkac312

Shimada, T., Ogasawara, H., & Ishihama, A. (2018). Genomic SELEX Screening of Regulatory Targets of Escherichia coli Transcription Factors. Methods in Molecular Biology (Clifton, N.J.), 1837, 49–69. https://doi.org/10.1007/978-1-4939-8675-0_4

Stormo, G. D., Schneider, T. D., Gold, L., & Ehrenfeucht, A. (1982). Use of the "Perceptron" algorithm to distinguish translational initiation sites in E. coli. Nucleic Acids Research, 10(9), 2997–3011.

Thomason, M. K., Bischler, T., Eisenbart, S. K., Förstner, K. U., Zhang, A., Herbig, A., Nieselt, K., Sharma, C. M., & Storz, G. (2015). Global transcriptional start site mapping using differential RNA sequencing reveals novel antisense RNAs in Escherichia coli. Journal of Bacteriology, 197(1), 18–28. https://doi.org/10.1128/JB.02096-14

Tierrafría, V. H., Rioualen, C., Salgado, H., Lara, P., Gama-Castro, S., Lally, P., Gómez-Romero, L., Peña-Loredo, P., López-Almazo, A. G., Alarcón-Carranza, G., Betancourt-Figueroa, F., Alquicira-Hernández, S., Polanco-Morelos, J. E., García-Sotelo, J., Gaytan-Nuñez, E., Méndez-Cruz, C.-F., Muñiz, L. J., Bonavides-Martínez, C., Moreno-Hagelsieb, G., … Collado-Vides, J. (2022). RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in Escherichia coli K-12. Microbial Genomics, 8(5). https://doi.org/10.1099/mgen.0.000833

Tierrafría, V. H., Mejía-Almonte, C., Camacho-Zaragoza, J. M., Salgado, H., Alquicira, K., Ishida, C., Gama-Castro, S., & Collado-Vides, J. (2019). MCO: Towards an ontology and unified vocabulary for a framework-based annotation of microbial growth conditions. Bioinformatics (Oxford, England), 35(5), 856–864. https://doi.org/10.1093/bioinformatics/bty689

Tsagmo Ngoune, J. M., Njiokou, F., Loriod, B., Kame-Ngasse, G., Fernandez-Nunez, N., Rioualen, C., van Helden, J., & Geiger, A. (2017). Transcriptional Profiling of Midguts Prepared from Trypanosoma/T. congolense-Positive Glossina palpalis palpalis Collected from Two Distinct Cameroonian Foci: Coordinated Signatures of the Midguts' Remodeling As T. congolense-Supportive Niches. Frontiers in Immunology, 8, 876. https://doi.org/10.3389/fimmu.2017.00876

Turatsinze, J.-V., Thomas-Chollier, M., Defrance, M., & van Helden, J. (2008). Using RSAT to scan genome sequences for transcription factor binding sites and cis-regulatory modules. Nature Protocols, 3(10), 1578–1588. https://doi.org/10.1038/nprot.2008.97

van Helden, J., André, B., & Collado-Vides, J. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. Journal of Molecular Biology, 281(5), 827–842. https://doi.org/10.1006/jmbi.1998.1947

van Helden, J., Rios, Alma. F., & Collado-Vides, J. (2000). Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. Nucleic Acids Research, 28(8), 1808–1818. https://doi.org/10.1093/nar/28.8.1808

Weiss, V., Medina-Rivera, A., Huerta, A. M., Santos-Zavaleta, A., Salgado, H., Morett, E., & Collado-Vides, J. (2013). Evidence classification of high-throughput protocols and confidence integration in RegulonDB. Database: The Journal of Biological Databases and Curation, 2013, bas059. https://doi.org/10.1093/database/bas059

Yan, B., Boitano, M., Clark, T. A., & Ettwiller, L. (2018). SMRT-Cappable-seq reveals complex operon variants in bacteria. Nature Communications, 9(1), 3676. https://doi.org/10.1038/s41467-018-05997-6

Zhou, D., & Yang, R. (2006). Global analysis of gene transcription regulation in prokaryotes. Cellular and Molecular Life Sciences CMLS, 63(19), 2260–2290. https://doi.org/10.1007/s00018-006-6184-6