Algorithms for the global mapping of RNA-RNA interactomes

DISSERTATION

Von der Fakultät Energie-, Verfahrens- und Biotechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Doctor rerum naturalium, Dr. rer. nat) genehmigte Abhandlung

Vorgelegt von

Richard A. Schäfer

aus Essen

Hauptberichter: Prof. Dr. Björn Voß

Mitberichter: Prof. Dr. Peter Stadler

Mündliche Prüfung: 26.09.2023

University of Stuttgart
Germany

Institute für Bioverfahrenstechnik
der Universität Stuttgart

2024

German title:

Algorithmen zur globalen Abbildung von RNA-RNA Interaktomen

to iris, my mother,
and my sisters

"Research means that you don't know,
but are willing to find out"
- Charles F. Kettering

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Ich erkläre hiermit, dass der Inhalt dieser Dissertation, außer in den Fällen, in denen ausdrücklich auf die Arbeit anderer verwiesen wird, eine Originalarbeit ist und nicht weder ganz noch teilweise für einen anderen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Universität eingereicht wurde. Diese Dissertation ist meine eigene Arbeit und enthält nichts, was das Ergebnis einer Zusammenarbeit mit anderen ist, es sei denn, dies ist im Text und in den Danksagungen angegeben.

Richard A. Schäfer
July 2024

Acknowledgements

This work would have been hardly possible without the support of numerous people from inside and outside the scientific community. First and foremost I would like to express my deepest gratitude to Prof. Dr. Björn Voß, who leads the group of *RNA Biology & Bioinformatics* at the Institute of Biochemical Engineering (IBMG) offering advice and encouragement with a perfect blend of insight and humor. His expert knowledge in the field of non-coding RNA and bioinformatics directed the progression in this field. Thank you, Björn, for offering me the position to participate in the exciting projects inteRNAct and RNAProNet, and enabled me to present the work at numerous conferences.

Also, I direct my thankfulness to the Co-examiner Prof. Dr. Peter Stadler and examination chair Prof. Dr. Stefan Legewie as well as Prof. Dr. Jörn Lausen, Prof. Dr. Christina Wege and Prof. Dr. Jürgen Pleiss for their participation during the circulation procedure.

No research can be adequately successful without the fruitful collaboration. Therefore, I apprecite the cooperation with Dr. Steffen C. Lott, Dr. Jens Georg and Prof. Dr. Wolfgang C. Heß from the group of Genetics & Exerimental Bioinformatics (cyanolab) at the University of Freiburg.

I want to thank all my wonderful colleagues who became good friends to me. I would like to thank Christoph Schaal, my longtime project partner who was never tired of answering all my questions about biology - these were nearly exhaustible. Thank you for the nice atmosphere and overall great time we had during all these years. I want to thank Dr. Brigitte Schönberger, who although she came late to the party made a significant impact of my work with her great insights and overall positive attitude. I also want to thank my office colleague Adrian Eilingsfeld, who participated in all the stupid things initiated. Special thanks go to Andreas Ankenbauer for a great collaboration, which not only led to the publication of our work but also enriched our friendship. Another heartfelt appreciation goes to Fikrat Talibli, whose engaging conversations and shared computational insights greatly enriched our interactions both during breaks and in our ongoing discussions.

Additional acknowledgments extend to individuals I had the pleasure of meeting Maria Ankenbauer, Christopher Sarkizi, Andreas Schwendtner, Felix Thoma, Michaela Graf, Lisa Junghans, Thorsten Haas, Marius Braakman, Claudia Hartung and everyone I forgot to mention.

This list would not be complete without mentioning Silke Reu, the good soul of the institute, who was never shy to answer all my administrational questions from which I even picked up some limited swabian along the way.

I would like to extend my heartfelt gratitude to everyone who enthusiastically participated in various sports activities after work, including football, volleyball, the campus run, or winter sports during our annual Söllerhaus retreat.

I want to thank everyone that participated in "cakefriday" and enjoys cake as much as I do. I miss those days.

Lastly, I can’t end this acknowledgements without including my family as they are worthy of limitless praises. I appreciate my wife Iris for the constant support, the bad moods and deprivations you had to endure. This would not have been possible without you. I would like to thank my mother for always beeing there, the constant love and support. I also want to thank my sisters who backed me up with the entire timeframe of the work.

List of publications

The workflows and analyses that are presented in this thesis were conducted under leadership and supervision of Prof. Dr. Björn Voß at Institute of Biochemical Engineering (University of Stuttgart, Germany). It has been supported by grants from the German Ministry of Education and Research, RNAProNet [031L0164A to B.V] and inteRNAct [031A310 to B.V.].

Parts of this work have been published in peer-reviewed jounals.

Schäfer, RA., and Voß, B. (2021). RNANUE: efficient data analysis for RNA-RNA interactomics. *Nucleic Acids Research*, 49:10

Schäfer, RA., Lott, SC., Georg, J., Grüning, BA., Hess, WR., Voß, B. (2020). GLASSgo in Galaxy: high-throughput, reproducible and easy-to-integrate prediction of sRNA homologs, *Bioinformatics*, 36:15

Lott, SC., **Schäfer, RA.,** Mann, M., Backofen, R., Hess, WR., Voß, B., and Georg, J. (2018) GLASSgo – Automated and Reliable Detection of sRNA Homologs From a Single Input Sequence. *Frontiers in Genetics*, 7

Schönberger, B., Schaal, C., **Schäfer, RA.,** and Voß, B. (2018). RNA interactomics: recent advances and remaining challenges. *F1000Research*, 7:1824

Schäfer, RA., and Voß,B. (2016). VISUALGRAPHX: interactive graph visualization within GALAXY, *Bioinformatics*, 32:22

Excerpts of this research have additionally been presented at international conferences as posters(∗) or talks(∘)

∘ **Schäfer, RA.,** and Voß, B. (2018) Analysis of RNA-RNA interaction data *German Conference on Bioinformatics 2018. Vienna, Austria*

∗ **Schäfer, RA.,** and Voß, B. (2016) - Analysis of chimeric reads for the detection of RNA-RNA Interactions. *Intelligent Systems for Molecular Biology / European Conference on Bioinformatics 2017, Prague, Czech Republic*

∗ **Schäfer, RA.,** and Voß, B. (2016) - Interactive Graph Visualization. *German Conference on Bioinformatics 2016, Berlin, Germany*

∗ **Schäfer, RA.,** and Voß, B. (2015) - Interactive Graph Visualization. $[BC]^2$ - *Basel Computational Conference 2015, Basel, Switzerland*

Summary

RNA-RNA intra- and intermolecular interactions are fundamental for numerous biological processes. While there are reasonable approaches to map RNA secondary structures genome-wide, understanding how different RNAs interact to carry out their regulatory functions requires mapping of intermolecular base pairs. RNA-RNA interaction prediction algorithms alone are not capable to consider all biological factors, thus they suffer from low accuracy. Recently, different strategies to detect RNA-RNA duplexes in living cells, so called direct duplex detection (DDD) methods, have been developed. Common to all is the psoralen-mediated in vivo RNA crosslinking followed by RNA Proximity Ligation to join the two interacting RNA strands. Sequencing of the RNA via classical RNA-Seq and subsequent specialised bioinformatic analyses which results in the prediction of intra- and intermolecular RNA-RNA interactions. Existing approaches adapt standard RNA-seq analysis pipelines, but often neglect inherent features of RNA-RNA interactions that are useful for filtering and statistical assessment. In this work, `RNAnue` is presented, a general pipeline for the inference of RNA-RNA interactions from DDD experiments that takes into account hybridisation potential and statistical significance to improve prediction accuracy. `RNAnue` was applied to data from different DDD studies and the results were compared to those of the original methods. This showed that `RNAnue` performs better in terms of quantity and quality of predictions.

Zusammenfassung

Intra- und intermolekulare RNA-RNA-Interaktionen sind für zahlreiche biologische Prozesse von grundlegender Bedeutung. Es gibt zwar etablierte Ansätze, um RNA-Sekundärstrukturen genomweit abzubilden, aber um zu verstehen, wie verschiedene RNAs interagieren, um ihre regulatorischen Funktionen auszuführen, müssen auch die intermolekularen Basenpaare berücksichtigt werden. Algorithmen zur Vorhersage von RNA-RNA-Interaktionen allein sind nicht in der Lage, alle biologischen Faktoren zu berücksichtigen, so dass sie eine geringe Genauigkeit aufweisen. In jüngster Zeit wurden verschiedene Strategien zum Nachweis von RNA-RNA-Duplexen in vivo, so genannte direkte Duplexnachweisverfahren (DDD), entwickelt. Allen gemeinsam ist die Psoralen-vermittelte in vivo RNA-Vernetzung, gefolgt von einer sogenannten RNA-Proximity-Ligation, um die beiden interagierenden RNA-Stränge zu verbinden. Die Sequenzierung der RNA mittels klassischer RNA-Seq und anschließender spezialisierter bioinformatischer Analysen führt zur Vorhersage von intra- und intermolekularen RNA-RNA-Interaktionen. Bestehende Ansätze passen etablierte RNA-Seq Analysepipelines an, vernachlässigen aber oft inhärente Merkmale von RNA-RNA-Interaktionen, die für die Filterung und statistische Auswertung nützlich sind. Im Rahmen dieser Arbeit wurde `RNAnue` entwickelt, eine allgemeine Pipeline für die Inferenz von RNA-RNA-Interaktionen aus DDD-Experimenten, die das Hybridisierungspotenzial und die statistische Signifikanz berücksichtigt, um die Vorhersagegenauigkeit zu verbessern. `RNAnue` wurde auf Daten aus verschiedenen DDD-Studien angewandt, und die Ergebnisse wurden mit denen der ursprünglichen Methoden verglichen. Dabei zeigte sich, dass `RNAnue` in Bezug auf Quantität und Qualität der Vorhersagen besser abschneidet.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

CLASH Cross-linking, ligation, and sequencing of hybrids

CM Covariance Model

DDBj DNA Data Bank of Japan

DDD Direct Duplex Detection

DEA Differential Expression Analysis

dsRNA double-stranded RNA

EMBL-EBI EMBL's European Biology Laboratory

EMBL European Molecular Biology Laboratory

FSM Finite-State Machine

GFF General Feature Format

INSDC International Nucleotide Sequence Database Collaboration

JGF JSON Graph Format

JSON JavaScript Object Notation

LCS Longest Common Subsequence

MFE Minimum Free Energy

miRISC miRNA-induced Silencing Complex

NCBI National Center for Biotechnology Information

ncRNA Non-coding RNA

NoSQL Not only SQL

ORF Open Reading Frame

PDB Protein Data Bank

PPV Positive Predictive Value

RNA-Seq RNA Sequencing

RNA Ribonucleic Acid

RPL RNA Proximity Ligation

SAM Sequence Alignment Map

SIF Simple Interaction File

SQL Structured Query Language

ssRNA single-stranded RNA

SVG Scalable Vector Graphic

TMM trimmed mean of M values

UMI Unique Molecular Identifier

Chapter 1

Introduction

With the advent of high-throughput technologies, scientists are no longer restricted to small-scale experiments. Instead, millions of biological screens can be carried out simultaneously. This has revolutionised the field of molecular biology and paved the way for a broad range of experiments that allow studying the cell's inner workings on a global scale. Consequently, large quantities of data are being generated that can no longer be analysed by hand. This led to the introduction of bioinformatics in almost all research fields, which deals with analysing large amounts of biological information. In addition, data needs to be stored and retrieved efficiently. Biological data refers to data obtained from so-called 'omics'-technologies that explore the molecules that make up the cells. These data streams include genomics, transcriptomics, and proteomics, among others. Data associated with scientific papers or genome projects are stored in publicly accessible data banks. These are usually in the hands of international consortia. For example, the International Nucleotide Sequence Database Collaboration (INSDC; http://www.insdc.org) operates between the DNA Data Bank of Japan (DDBJ; http://www.ddbj.nig.ac.jp), EMBL's European Bioinformatics Institute (EMBL-EBI; https://www.ebi.ac.uk), and National Center for Biotechnology Information (NCBI; https://ncbi.nlm.nih.gov), comprising the largest DNA and RNA database (Cochrane et al., 2016). Similarly, UniProt (The UniProt Consortium et al., 2021), which lists protein sequence as well as functional information, and the Protein Data Bank (PDB; https://wwpdb.org/; Burley et al., 2017), which contains macromolecular structures, are maintained by international consortia. As the amount of data continues to grow, the challenges of efficient storage need to be addressed. For example, the number of bases maintained by the INSDC doubles approximately every 18 months. Likewise, analysing these large datasets requires efficient algorithms to gain results in a reasonable time. This also includes predictions that extend the experimental data. Ultimately, bioinformatics aims to integrate combinations of different types of data to understand natural phenomena.

The basis of this work is data from transcriptomics studies that denote the high-throughput sequencing of the transcriptome, that is, the entirety of the RNA transcripts at a given time. Studies have provided scientists with more significant insights into biological pathways and molecular mechanisms. RNA functions as a messenger between DNA and protein and is also involved in regulating gene expression. These non-coding (nc)RNAs perform their task predominantly through interaction with other RNA molecules, and it is of great interest to decipher the RNA interactome. Although these regulators can be readily identified experimentally, laborious essays are still required to determine their targets. In this regard, computational approaches provide solid target predictions but with moderate prediction accuracy and are, therefore, unreliable. Other approaches aim to combine experimental methods with computational analyses. The work presented in this thesis focuses on methods to decipher a global RNA-RNA interaction map in different organisms. This includes data storage and retrieval of these interactions as well as their visualisation on a global scale.

1.1 Building Blocks of DNA

In the early 1950s, Rosalind Franklin used X-ray diffraction to determine the structure of DNA molecules. 'Photo 51' became her best X-ray picture and was instrumental to James D. Watson and Francis J. Crick in deducing the double-helix of DNA (Watson and Crick, 1953). DNA comprises repeating monomer units called *nucleotides* that are deoxyribose sugars connected to one or more phosphate groups and a base. DNA's bases include adenine (A), guanine (G), cytosine (C), and thymine (T), resulting in four distinct nucleotides. A strong chemical bond known as the β-glycosidic bond joins the bases to the deoxyribose. The bases are generally distinguished into chemically distinct groups named *purines* (A,G) and *pyrimidines* (C,T). Nucleotides are joined together by phosphodiester bonds. One ester bond links a phosphorus atom to the 3' carbon of the upstream ribose sugar, and the other ester bond links the same phosphorus atom to the oxygen atom attached to the 5' carbon atom of the downstream ribose sugar. The resulting nucleic acid starts with a 5' phosphate group and finishes with a 3' -OH group, giving a directional property denoted 5'-3'.

Figure 1.1 illustrates the general structure of a nucleotide and the different nucleotides that make up DNA. The primary structure of DNA is the sequence of the nucleotides. In its secondary structure, the DNA extends to a double-stranded helix, which forms between two complementary DNA strands over their entire length. In principle, there are three major forms of DNA, while the B-DNA commonly forms under normal physiological conditions. In it, the phosphodiester backbones of the nucleotide chains are on the outside, and the bases are on the inside. The arrangement of the two chains runs anti-parallel, meaning that

one phosphodiester backbone is in a 5'-3' orientation, and the phosphodiester backbone to which it is paired is in the opposite (3'-5') orientation. Gaps that are referred to as grooves lie between the phosphodiester backbone and spiral around the outside of the helix. These are known as deep major and shallow minor grooves and are essential for protein-DNA interactions. The bases projecting from the two phosphodiester backbones within the helix interact through hydrogen bonds. These bonds are known as Watson-Crick base pairs and form between A and T pairs, which involve two hydrogen bonds, or between G and C pairs, which are connected by three hydrogen bonds and are thus more stable than the A-T pairs. The helix is further stabilised by base stacking, in which interior bases are stacked on each other.

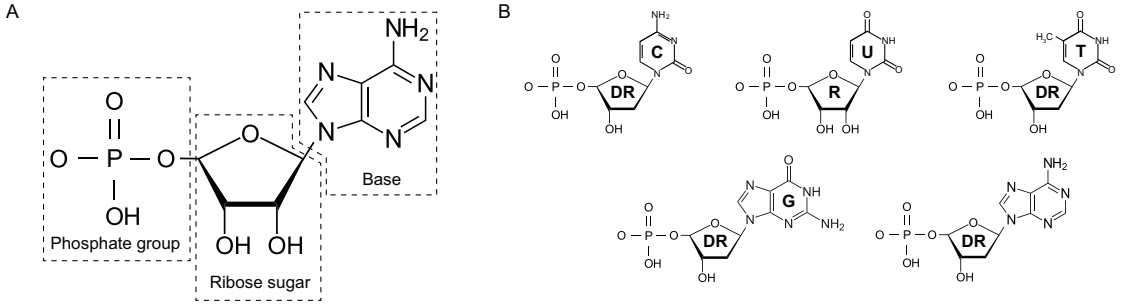

Fig. 1.1 (A) Nucleotides are assembled from ribose sugars, bases and phosphate groups. (B) Groups of nucleotides that are used in DNA and RNA. In RNA, uracil is used instead of thymine and a ribose sugar instead of deoxyribose.

1.2 The emergence of RNA

In 1958, Francis Crick first devised the *Central Dogma of Molecular Biology*, which describes the transcription of DNA to RNA in the nucleus, followed by protein synthesis in the cytoplasm. It was published over a decade later in a slightly modified version (Crick, 1970) following the discovery of reverse transcriptase. This shifted the research towards RNA, which became crucial for the development of modern molecular biology. RNA consists of similar building blocks as DNA but has a few differences. RNA molecules contain different sugars as DNA: ribose instead of deoxyribose. Ribose contains an additional 2'-OH that has two consequences for the function of RNA compared with DNA. First, the 2'-OH group in the ribose sugar is polar, making RNA more chemically reactive than DNA. Second, the ribose sugar is slightly twisted (sugar pucker) to minimise interactions with other non-bonding atoms attached to the ring. In the nucleotides, DNA and RNA use a different but overlapping set of bases. Although both DNA and RNA contain nucleotides with four different bases, a

clear difference is that RNA uses uracil (U), whereas DNA uses thymine (Figure 1.1). This is important as the spontaneous deamination of cytosine to uracil allows the cell to detect and repair nucleic acid damage.

As DNA has a mostly permanent structure, RNA can adopt various secondary and tertiary structures. This ability of RNA to fold into diverse structures enables it to be involved in a number of biological processes. In contrast, DNA's principal role is to store genetic information. A few years prior to the *Central Dogma* in 1955, Georges Palade identified the very first non-coding RNA that constitutes a part of the cytoplasmic ribonucleoprotein (RNP) complex: the ribosome. In addition, Francis Crick theorised that there was an 'adapter' molecule for the translation of RNA to amino acids. Eventually, this became to be known as another class of non-coding RNA: The transfer (t)RNA. Since then, different classes of RNAs have continuously emerged. In 1960, Francois Jacob and Jacques Monod showed the existence of an intermediate molecule carrying the genetic information leading to protein synthesis, known as messenger (m)RNA. In the late 1960s, heterogeneous nuclear (hn)RNAs were discovered that function as the precursor of rRNA and mRNA. The focus was thus directed on the study of rRNA processing that, in turn, led to the discovery of splicing. It wasn't long before small nuclear (sn)RNAs, which are part of the spliceosome, the RNP machinery responsible for intron splicing from pre-mRNA, were discovered, as well as small nucleolar (sno)RNAs, which are involved in the processing and maturation of ribosomal RNAs in the nucleolus. At that time, RNA was considered to function solely as the bridge between DNA and protein, and the role of different classes of RNA was ignored. However, this view was overturned when Thomas Cech and Sydney Altman discovered that RNA molecules could act as catalysts for a chemical reaction. These RNA enzymes were depicted as ribozymes as they have been shown to be part of both the ribozyme and the spliceosome (Altman, 1990; Cech, 1990). This observation led scientists to develop the *RNA World Hypothesis*, which states that prebiotic life consisted of simple replicating microbes (ribocytes) in which fundamental biochemical processes depended entirely on RNA (Gesteland Raymond et al., 2006). Interestingly, RNA still works as a genome for retroviruses (e.g., Ebola, HIV).

1.2.1 RNA is mostly non-coding

In the late 1970s, Frederick Sanger generated the first complete genomic sequence of the bacteriophage $\phi X174$ using the established Sanger sequencing technology (Sanger et al., 1978). This was then routinely used in the following years, and Walter Gilbert was awarded the Nobel Prize in Chemistry. In the 1990s, a worldwide sequencing effort, the Human

Genome Project (HGP), was established by the National Institute of Health (NIH) to sequence the human genome completely. One of the findings was the relatively low number of protein-coding genes compared to the initial expectation. It is now known that the human genome consists of about $\sim$22.287 protein-coding genes that do not include non-coding RNAs (International Human Genome Sequencing Consortium, 2004). However, the total length covered by the exons of protein-coding genes spans only about 1.2% of the euchromatic genome. Kapranov et al. (2002) identified about 90% of the transcripts spanning the human chromosomes 21 and 22 mapping to non-coding genomic regions. It is assumed that up

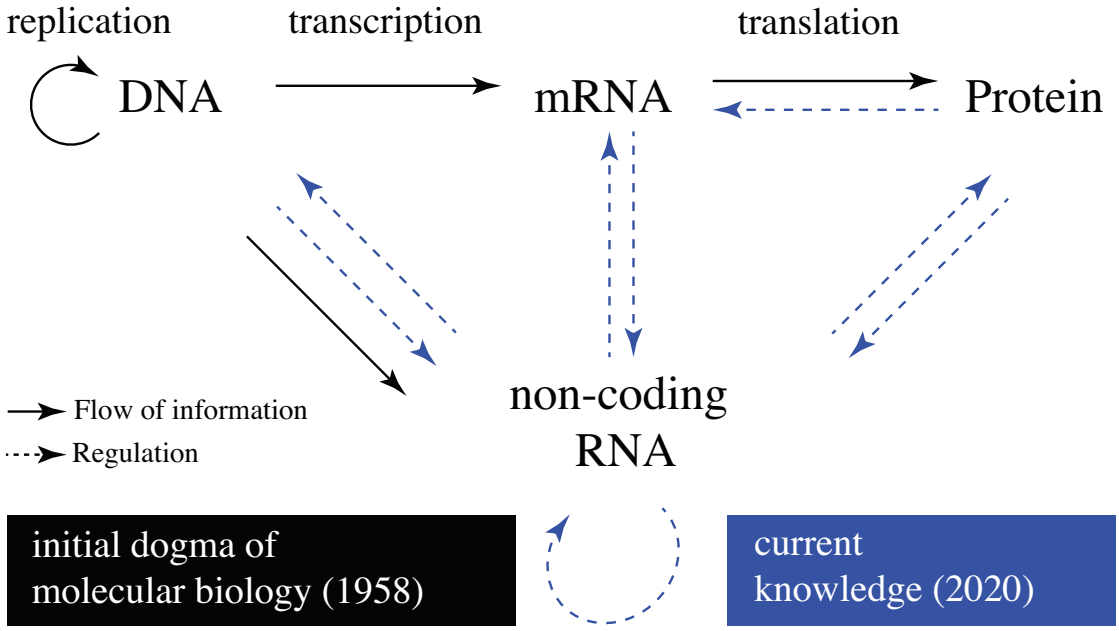

Fig. 1.2 The central dogma of molecular biology with today's knowledge.

to 85% of the human genome is transcribed into RNA (Hangauer et al., 2013), thereby producing a large portion of the transcriptome that is non-coding (Frith et al., 2005). This is apparent when looking at the ratio of non-coding to coding sequences, which is 47:1 in the human transcriptome. In comparison, this ratio is 43:1 in mice, 2.4:1 in *D. melanogaster* and 1.3:1 in *C. elegans* (Elliott and Ladomery, 2015). However, many transcripts overlap with protein-coding genes in sense, coding, or antisense strands. This is known as pervasive transcription and is widespread among eukaryotes (van Bakel et al., 2010). Similar pervasive transcription exists in prokaryotes (Georg and Hess, 2018; Lybecker et al., 2014). These discoveries have broadened our understanding of the role of RNA in various biological processes. Figure 1.2 illustrates the central dogma of molecular biology by today's standards. It is, meanwhile, common knowledge that ncRNAs form internal base pairs to determine their

function. Different types of ncRNAs are known to be involved in the regulation of DNA (e.g., chromatin structures), RNA (e.g., stability, decay), or protein (e.g., protein sequestration).

1.3 Regulatory RNAs

Various non-coding RNAs (ncRNAs), such as ribosomal, transfer, and messenger RNA molecules (rRNAs, tRNAs, mRNAs), represent the mediators of genomic information on the pathway to protein biosynthesis. In particular, rRNAs are present in cells at a high percentage of more than 95% (Peano et al., 2013). In addition, various regulatory RNAs have been identified within the last decade in all three domains of life. These include small regulatory RNAs (sRNAs) in prokaryotes that have been known to control bacterial adaption to environmental changes. In Table 1.1, different classes of ncRNAs are listed, illustrating the diversity of ncRNAs. In principle, these can be distinguished into housekeeping ncRNAs that merely adopt functions required for the cells' basic function that is expressed in all cells of an organism under normal conditions. In contrast, regulatory ncRNAs play a significant role in the modulation of gene expression in response to external stimuli. In the following, the function of the most common ncRNAs is described.

1.3.1 Bacterial small regulatory RNA

Bacterial small regulatory RNAs (sRNAs) were discovered in prokaryotes way before the first microRNAs (miRNAs) and small interfering RNAs (siRNAs) in eukaryotes. The regulation mediated by these ncRNAs is predominantly performed after transcription of their target, thus known as post-transcriptional regulation. The first bacterial sRNA was discovered in 1984. It was shown that *micF* regulates the translation of a target mRNA encoding an outer membrane protein in *E. coli*. Since then, many other sRNAs ranging in length from 50 to 300 nucleotides (Jørgensen et al., 2020) have been discovered. These are usually highly expressed when cells are undergoing stress (e.g., nutrient starvation). In many aspects, sRNA-mediated regulation is much more effective than regulation via transcription factors or protein-based mechanisms (Beisel and Storz, 2010). sRNAs targeting mRNAs are usually classified into cis- and trans-encoded sRNAs. Most commonly, cis-encoded RNAs are encoded on the DNA strand opposite of protein-coding genes and overlap its 5'- or 3'-UTRs. As a consequence, they show a high degree of complete complementarity with the individual target gene. In contrast, trans-encoded sRNAs exert a low degree of complementarity and may exceed cis-encoded transcripts in length and have multiple targets. These sRNAs are known to originate from intergenic regions but have also been shown to be encoded within the 3'-UTR of certain

mRNAs (Kawano et al., 2005). The resulting sRNAs are generated either by processing from the mRNA or by transcription from an internal promoter within the ORF. Notably, trans-encoded sRNAs in bacteria often rely on sRNA-binding proteins such as Hfq or ProQ, promoting the binding of the sRNA-mRNA complex. Moreover, sRNAs were shown to control the activity of regulatory proteins, such as CsrA (Storz and Papenfort, 2018). sRNAs bind to their mRNA targets by base pairing, inducing either positive or negative regulation. In the former, this leads to stabilization and/or translational activation of an mRNA target (Storz et al., 2004). This occurs by base-pairing within the 5'-UTR of the mRNA, thus, changing the folding of the ribosome binding site (RBS). As a consequence, the ribosome can bind and initiate translation (Gottesman, 2010). Regulation on the 5'-UTRs is accompanied by different mechanisms that include metabolites, proteins, and/or other sRNAs (Holmqvist and Vogel, 2018; Ignatov and Johansson, 2017; Kavita et al., 2018).

type	abbreviation	full name	size
Housekeeping ncRNAs	rRNA	ribosomal	
	tRNA	transfer RNA	
	snRNA	small nuclear RNA	
	snoRNA	small nucleolar RNA	
	TERC	telomerase RNA	
	tRF	tRNA-derived fragments	
	tiRNA	tRNA halves	
Regulatory ncRNAs	miRNA	micro RNA	
	siRNA	small interfering RNA	
	piRNA	piwi-interacting RNA	
	eRNA	enhancer RNA	
	lncRNA	long non-coding RNA	
	circRNA	circular RNA	
	Y RNA	Y RNA	

Table 1.1 Classification of non-coding RNAs

Another mechanism of sRNA-mediated regulation leads to translational repression and/or degradation. In this, the sRNA base pairs directly with the RBS and represses translation by preventing access of initiating ribosomes. In most cases, this results in irreversible rapid mRNA degradation. This is achieved by recruiting specific endoribonuclease E (RNAse E). One reason for the prevalence of sRNA-mediated regulation lies in metabolic costs. sRNA genes are encoded on small genome regions that require little energy to be transcribed into

RNA. In contrast, mRNA genes are greater in length, with an average size of ~ 1000 (Jones et al., 2007). Consequently, the sRNA is not translated into protein, so no further energy must be expended. This reduced energy consumption for the expression may be used for cell growth and maintenance, thus benefitting the organism. In addition, sRNA-mediated regulation provides another layer of regulation. It is known that protein regulators bind within a hundred nucleotides of -35 and -10 promoter elements. As a consequence, the regulation is limited to a few transcription factors. Instead, sRNA targets a different part of the gene with many sites on which regulation can occur. For example, the alternative sigma factor σ is modulated under different conditions (Hengge-Aronis, 2002) and is consequently regulated at the transcriptional and post-transcriptional level. van Nimwegen (2006) examined the transcription regulators in bacteria and found that bacteria with larger genomes use a more significant fraction of transcription regulators. This means that in bacteria with larger genomes, the genes are either controlled by more transcription regulators or each transcription regulator controls fewer genes. Another advantage of the prevalence of sRNA-mediated regulation is its regulatory speed. Shimoni et al. (2007) evaluated different modes of regulation and found that sRNA achieves faster regulation. This was detected for both positive and negative regulation. These findings seem plausible since the sRNA-mediated regulation acts at the post-transcriptional level, thereby reaching their target faster. This faster regulation is helpful for the organisms when sudden adaptation to external stimuli is needed.

1.3.2 Short non-coding RNAs

Short ncRNAs in eukaryotes consist of small interfering RNAs (siRNAs), microRNAs (miRNAs), and piwi-interacting RNAs that generally carry out diverse roles in the cell. Broadly, Elliott and Ladomery (2015) classified those into different groups: the siRNAs, which either target RNAs for degradation or chromatin for modifications and the miRNAs that regulate translation. However, microRNAs can also promote mRNA degradation and epigenetic changes in the nucleus (Elliott and Ladomery, 2015). The mechanisms of these short ncRNAs are similar; thus, one can assume that they have evolved early in the history of life (Farazi et al., 2008). In principle, short ncRNAs are generated from precursor molecules. In the case of siRNAs, the cytoplasmic endonuclease Dicer cuts the pre-siRNA into short dsRNA fragments. Afterwards, the short dsRNAs are loaded onto an Argonaute protein. In particular, one RNA strand remains bound to the Argonaute (guide RNA) while the other RNA strand is discarded (passenger RNA). The Argonaute protein, referred to as siRNA-induced Silencing Complex (siRISC), then targets RNAs in the cell through Watson-Crick base pairing. It performs cleavage of the target RNA or is involved in heterochromatin formation. In the biosynthesis of miRNAs, the primary transcript is first trimmed by an enzyme called

Drosha and the dsRNA-binding protein Pasha. This generates the pre-miRNA of ~ 70 nucleotides with a hairpin loop structure that is subsequently exported to the cytoplasm using the exportin-5 protein. As with the siRNAs, the enzyme Dicer processes the pre-miRNA into the final miRNA transcript of ~ 22 nucleotides that assemble into the miRNA-induced Silencing Complex (miRISC). RISCs can generally be divided based on their cleaving and non-cleaving activities that depend on the Argonaute protein. For example, mammals have four Argonaute paralogs, AGO1, AGO2, AGO3, and AGO4, that share an identity of 80% in their amino sequence (Sasaki et al., 2003). Although different Argonaute proteins in *D. melanogaster* associate with short ncRNAs, AGO1 preferably associates with miRNAs and AG02 with siRNAs (Caudy and Hannon, 2004; Okamura et al., 2004). This is consistent with the discovery that only AGO2 retains its slicer activity (Liu et al., 2004; Meister et al., 2004). Consequently, non-cleaving RISCs block translation but do not degrade the target mRNA. In any case, Argonaute proteins also exist in prokaryotes, where they are involved in DNA interference, which prevents the propagation of foreign DNA (Swarts et al., 2014). The first microRNA was discovered in the nematode *C. elegans* (Lee et al., 1993). The *lin-4* gene produces small RNAs from a longer non-coding protein precursor. The longer RNA forms a stem-loop structure, which is cut to generate the mature microRNA with antisense complementarity to the 3'-UTR of the lin-14 transcript. Similarly, the *Drosophila* gene *bantam* also encodes a microRNA, which targets the 3'-UTR of the *Hid* mRNA. *Hid*, in turn, is a key regulator of apoptosis, and its downregulation by *bantam* microRNAs is required for normal cell development. Initially, their size was similar to siRNAs, leading to the hypothesis that they are part of the same process. It is now known that microRNAs are present both in the nucleus and cytoplasm. In the former, they are involved in epigenetic regulation affecting transcription and alternative splicing (Huang and Li, 2012). Furthermore, microRNAs have been implicated in cancer as they can act as tumour suppressors or oncogenes. They can regulate processes such as cell differentiation, proliferation, and apoptosis. In mammals, about 30% of microRNA genes are found in intergenic regions, but most originate from larger transcription units. They can be located within intronic and sometimes even exonic sequences. It is known that miRNAs regulate up to 90% of all human mRNAs. A plethora of microRNAs have been identified in eukaryotic cells in animals, plants, and even viruses. In particular, the miRBase v22 has been extended continuously, containing a plethora of hairpin precursors and mature miRNAs across various species (Kozomara and Griffiths-Jones, 2011). Most notably, microRNAs are usually found in clusters, suggesting that they might have arisen from duplication events. Nevertheless, several are polycistronic, originating from a single primary transcript. Its genomic location varies, as they are found in humans on all chromosomes except the Y chromosome. Finally, PIWI-interacting RNAs (piRNAs) are

a class of animal-specific ncRNAs used to control transposons. In principle, piRNAs are encoded on the genome in chromosomal regions called piRNA clusters that act as master regulators to control transposable elements. At first, a long pre-piRNA is transcribed from the piRNA cluster and is subsequently cut up to generate a number of 24-30 nucleotide-long piRNAs. These are either complementary to sense or antisense sequences of transposable elements. In that regard, the antisense piRNAs associate primarily with PIWI (p-element induced wimpy testis) and a PIWI-like protein known as Aubergine to form RNA-protein complexes. PIWI was first discovered in *D. melanogaster*, in which the destructive effect on the testis upon mutating the PIWI gene was observed (Grivna et al., 2006). It has been shown that PIWI proteins are essential for germ cell maintenance and spermatogenesis in D. melanogaster and mammals (Thomson and Lin, 2009). In this complex, the antisense RNA acts as guide RNA and binds through sequence complementarity to RNAs encoded by the transposons. Aubergine belongs to the Argonaute protein family, thereby having slicing activity that cuts the sense transposon-encoded RNAs bound to AGO3. This complex cleaves antisense piRNAs in the ling pre-piRNA. This is known as a ping-pong amplification loop (Bamezai et al., 2012). The piRNA clusters seem to be sites of highly efficient transposon insertion. As a consequence, transposable elements moving around the genome are most likely to move into the piRNA cluster. However, in some systems, these piRNAs are derived from repeated sequence elements (Brennecke et al., 2007), thereby being characterized as repeat-associated small interfering RNAs (rasiRNAs). Gan et al. (2011) argue that gene expression during mouse spermatogenesis is regulated post-transcriptionally and correlates with the production of piRNAs. Furthermore, piRNAs have also been associated with disease, indicating that piRNAs are aberrantly expressed in human cancer cells (Cheng et al., 2011).

1.3.3 Long non-coding RNAs

In late 1980, studies of genomic imprinting discovered the paternally expressed protein-coding gene *Igf2r* and the maternally expressed *H19*. These were localised on mouse chromosome 7 in proximity to each other, forming the *H19/IGF2* cluster (Bartolomei et al., 1991). It was observed that *H19* was not being translated even though the gene contained open reading frames (ORF). Furthermore, it showed high conservation across mammals, sharing features of mRNAs. Expression of *H19* in transgenic mice proved to be lethal in prenatal stages, suggesting that it has an important role in embryonic development. Since then, *H19* has been investigated thoroughly and represents the class long of non-coding RNAs (lncRNAs). These lncRNAs are now known as transcripts that are longer than 200nt. Based on their location in regard to protein-coding genes, lncRNA can be divided into different categories (Ma et al., 2013). Long intergenic ncRNAs (lincRNAs) are transcribed

from introns within protein-coding genes. In contrast, sense lncRNAs are transcribed from the sense strand and contain exons of protein-coding genes. This means that they may overlap with protein-coding genes or even cover their entire sequence. Similarly, antisense lncRNAs are transcribed from the antisense strand, overlap with exonic or intronic regions or cover the protein-coding sequence through an intron. The most commonly used definition of those transcripts is based on the length exceeding at least 200 nucleotides. They are observed in a large diversity of organisms, that include *h. sapiens* (Clemson et al., 1996), *m. musculus* (Chen et al., 2017), plants (Chen et al., 2020), and yeast (Till et al., 2018). However, lncRNAs have been proven to be poorly conserved, making it a challenge to deduce functional capabilities. In addition, these transcripts are usually lower expressed than protein-coding genes, which makes it difficult to distinguish their expression from transcriptional noise. Evidence suggests that lncRNAs are involved in various cellular functions that predominantly affect transcriptional and post-transcriptional regulation. In the former, this includes regulation through transcriptional inference (Kornienko et al., 2013) and chromatin remodelling (Senmatsu et al., 2021). In the latter, lncRNAs are involved in splicing regulation and translational control. This functions through binding to (Tsuiji et al., 2011) or modulating (Tripathi et al., 2010) splicing factors or by direct hybridisation with the mRNA sequence (Rintala-Maki and Sutherland, 2009), thus blocking the splicing. In contrast, transcriptional control functions through binding to either transcription factors (Muddashetty et al., 2002) or ribosomes (Zeng et al., 2018). Apart from those mechanisms, evidence suggests that lncRNAs may also be involved in small ncRNA-mediated mechanisms due to the association with siRNAs and miRNAs (Paraskevopoulou and Hatzigeorgiou, 2016). Moreover, many lncRNAs interact directly or indirectly with miRNAs to stabilise the target mRNA (Cesana et al., 2011). These are referred to as competing endogenous RNAs (ceRNAs).

1.4 RNA binding proteins

RNA-binding proteins play a central role in all the co-transcriptional and post-transcriptional processes described above. Therefore, it is important to consider the structure of RNA-binding proteins and the RNA-binding domains that enable these proteins to bind to RNA. They bind both single-stranded and double-stranded RNA. This includes the recognition of RNA sequences and structures at the three-dimensional level. In principle, the proteins can bind to the bases, ribose sugar or phosphate groups of the RNA. RNA-binding proteins were first studied in late 1960 when they were first described as chromatin-associated RNA-binding proteins, but it became apparent that they bind to nascent transcripts (pre-mRNA).

These proteins were known as heterogeneous nuclear ribonucleoproteins (hnRNPs) that bind to hnRNA. However, the term hnRNA is no longer used. Instead, they are known as pre-mRNAs. Elliott and Ladomery (2015) give a comprehensive listing of the known RNA-binding domains. The RNA recognition motif (RRM) is the best-studied domain. Hfq has originally been identified in *E. coli* as a host factor required for the efficient replication of the RNA bacteriophage Qβ (Carmichael et al., 1975) and is now known to function as an RNA chaperone in bacterial cells. It has been classified as a member of the Sm-like (Lsm) protein family that is predominantly found in eukaryotes and archaea. Sun et al. (2002) detected amino acid similarity and conserved pattern of Hfq in about 50% of all available bacterial genomes in the NCBI databases. It is widely known as a global regulator that is involved in the cell response to stress factors. Most notably, Hfq functions as an RNA chaperone, facilitating interactions between bacterial non-coding RNAs and their mRNA target. Consequently, modulating mRNA translation and stability. In recent studies, this regulatory role of Hfq has expanded to other processes (Kavita et al., 2018). For example, McaS is an Hfq-dependent sRNA that controls the expression of specific genes, either positively or negatively. McaS is induced during the transition into the stationary growth phase, and expression of McaS activates flagella synthesis by base-pairing with the 5'-UTR of the !hDC operon. This UTR forms a highly structured hairpin that sequesters the RBS and prevents translation. The binding of McaS to two regions in the leader sequence releases the secondary structure around the RBS, resulting in increased expression of FlhD and FlhC proteins and thus represents a classic example of sRNA-mediated positive regulation of translation. The RNA-binding protein CsrA (carbon storage regulator A) is conserved in a broad range of bacterial species encoding sRNAs. Initially discovered in *E. coli*, CsrA represents a global post-transcriptional regulator with pleiotropic effects. CsrA controls carbohydrate uptake and metabolism, biofilm formation, motility, quorum sensing, and more. Interestingly, the activity of CsrA is antagonized by sRNAs that act as decoys capable of sequestering CsrA. Two RNA 'sponges' are expressed in *E. coli*, CsrB and CsrC, which contain 22 and 13 GGA motifs, respectively. Current experimental methods to infer protein-RNA interactions include the crosslinking immunoprecipitation (CLIP) techniques (Ule et al., 2005).

1.5 RNA-RNA interactions

After the double helix of the DNA could have been deduced, the next challenge was to determine the molecular structure of RNA. This is of great interest as the RNA structure is often related to its function and, therefore, crucial for an understanding of its mechanism and

function. RNA molecules use three hierarchical levels of structural organisation. The *primary* structure is the linear sequence of nucleotides in a nucleic acid. In the *secondary* structure, helices form through base pairing within single molecules of RNA (intramolecular base pairing) and between different RNA molecules (intermolecular base pairing). The *tertiary* structure is the highest level of organisation, in which RNA molecules with secondary structures fold up into very compact structures. DNA has a relatively stable secondary structure with the double helix, while RNA can adopt a various secondary and even tertiary structures. This difference is due to the biosynthesis of both molecules. In particular, the parent strands are replicated during DNA replication to synthesise two identical double-stranded DNA molecules. In contrast, during transcription, the two parent strands of DNA are only transiently separated, and only one strand is used as a template for RNA synthesis. Consequently, the single-stranded RNA (ssRNA) that results from the transcription is left without a complementary partner and, thus, cannot form a double helix. Moreover, the helices in RNA may also include non-Watson-Crick base-pairing (Leontis et al., 2002). And what's more, most nucleotides can base-pair with each other within RNAs. This includes G-U wobbles or sheared A-A. In the former, the base pair has two hydrogen bonds, similar to the A-U pairing but with a slightly different shape, introducing a minor helix distortion. In the latter, ribose sugar forms hydrogen bonds between nucleotides. Moreover, evidence suggests that the formation of the RNA helix can be controlled (Wong et al., 2007). It has been shown that in *E. coli*, the RNA polymerase can stall during transcription to enable newly synthesised RNA to fold properly. The property of RNA to form shorter helices connected by single-strand regions allows the RNA to have diverse secondary structures as opposed to the double helix of the DNA. Five different structure motifs are most commonly found in RNA molecules. As described above, helices are the basic secondary structures that are formed through base pairing between antiparallel complementary sequences. Another form of secondary structure is loops, which are single-stranded regions within helices. The so-called hairpin stem loop is formed when nearby regions of complementary nucleotides form a short hairpin helix that is separated by a sequence which forms the loop. Similarly, internal loops are symmetrical and occur where two strands of a helix have an equal number of unpaired bases. In contrast, bulges are regions of non-complementarity in which one strand bulges out of the helix. Another form of secondary structure is pseudoknots that are formed by base pairing RNA sequences from a single-stranded region of an RNA and a loop. Kissing loop complexes are formed by hydrogen bonding between the single-stranded regions of loops. In turn, helical junctions join regions, which act as intersections to link different helices together. Usually, RNA molecules contain a multitude of different individual secondary structure motifs. Mortimer et al. (2014) detected more RNA secondary structures

in the untranslated regions of mRNAs than in the ORF. This is consistent with the findings that UTRs are targeted for binding by ncRNAs for RNA-mediated regulation. This means that secondary structure is important in preventing the base pairing between ncRNA and their target. In that regard, regions of secondary structure can be found within mRNAs at key points between the coding information for protein domains. This is important when slowing down the ribosome and stalling translation. This could be a mechanism to allow nascent protein sequences to fold before new protein sequences are translated. However, in yeast, the typical mRNA is more structured in the ORF than in either 5' or 3'-UTRs (Mauger and Weeks, 2010).

1.5.1 High-throughput methods for RNA structure interrogation

RNA structures have been studied extensively using magnetic resonance spectrography or X-ray crystallography (Butcher and Pyle, 2011). However, the experimental conditions often thoroughly characterise the RNA of interest, thus restricting the throughput. In recent years, high-throughput methods emerged that allow studying RNA structures on a global scale (Strobel et al., 2018). These so-called structure-probing approaches share a similar principle. In brief, enzymatic or chemical probes are used to modify single-strand nucleotides. As a consequence, modified nucleotides can no longer be copied by reverse transcriptase, thereby stopping cDNA synthesis. These stops in the cDNA were mapped back to the genomic location, showing the single-stranded regions in the RNA molecule. Subsequently, a reactivity profile is calculated for each nucleotide in the RNA that reflects the underlying RNA structure. There are numerous methods available that employ a similar protocol and are mainly distinguished by the type of probe used and the subsequent strategy to detect the modifications. For example, `PARS` (Kertesz et al., 2010) and `FRAG-Seq` (Underwood et al., 2010) use enzymatic probes that have been applied in transcriptome-wide RNA structure probing. However, these in-vitro methods are not able to fully represent the folding in the cellular environment (Mauger and Weeks, 2010). In contrast, chemical probes can determine the RNA structures on different levels. For example, selective 2'-hydroxyl acylation analysed by primer extension (SHAPE-Seq; Lucks et al., 2011) has been used to determine the structure of rRNAs (Deigan et al., 2009), tRNAs (Kladwang et al., 2011), and ribozymes (Lucks et al., 2011). Strobel et al. (2018) provide a comprehensive review of current high-throughput RNA structure probing techniques.

1.5.2 RNA structure prediction

In some cases, structure probing experiments are impractical, or the experimental results need to be interpreted with additional information. Here, RNA structure prediction can play an important role. In addition, it can aid in experimental design when probing for function. Numerous tools have been developed to predict the RNA's secondary or tertiary structure starting from its sequence. Most commonly, they utilise dynamic programming combined with thermodynamics to determine the secondary structure for a given RNA sequence. In doing so, the algorithms determine the structure ensemble that provides the lowest free energy change between the unfolded and folded state of an RNA. The structure with the minimum Gibbs free energy will be the most prevalent in the solution at equilibrium. These free energy changes can be estimated using the nearest neighbour model, which assumes that the free energy change for forming a base pair depends only on the sequence identities of that pair and the one immediately neighbouring it. These parameters for this model have been determined using optical melting experiments (Mathews et al., 1999; Mathews and Turner, 2002b; Mathews David et al., 2004). Based on the assumption that the structural motifs are independent, the free energy change for a given structure can be computed by simply adding up all the energies associated with forming the motifs. However, this assumes that the RNA is at equilibrium and the parameters of the nearest neighbour model are without error. This may not hold true for all RNAs. For that, a partition function is introduced that indicates the pairs that are more likely to be correct. It defines the sum of the equilibrium constants for all possible secondary structures for a given sequence. Then, the probability corresponds to the constant for that structure divided by the partition function.

`RNAstructure` (Bellaousov et al., 2013; Reuter and Mathews, 2010), `RNAfold` (Lorenz et al., 2011), and `NUPACK` (Dirks and Pierce, 2003, 2004) report the minimum free energy (MFE) structure combined with base pairing probabilities. Similarly, `sfold` (Ding et al., 2004; Ding and Lawrence, 2001, 2003) computes partition functions but uses a more simplistic model in which the base pair probabilities are computed directly. In contrast, `mfold` (Zuker, 1989, 1994, 2003; Zuker et al., 1999) merely relies on the computation of optimal and suboptimal foldings. An alternative approach considers low-free energy ensembles using stochastic sampling. This is implemented in `UNA-fold` (Markham and Zuker, 2008). However, algorithms that predict the RNA structure from a single sequence of length L still suffer from high complexity, that is $\mathcal{O}(L^3)$ when pseudoknots are absent and $\mathcal{O}(L^6)$ otherwise. In terms of prediction performance, the overall prediction accuracy ranges on average from around 65% to 70% for both the sensitivity and positive predictive value (PPV) (Andronescu et al., 2007; Mathews et al., 2004). This is mainly caused by imperfect thermodynamic parameters

and the limitations of the secondary structure model in which tertiary structures, pseudoknots and other properties are not accounted for (Lorenz et al., 2016a). Most commonly, pseudoknots are omitted in the prediction due to their computational complexity. In fact, it has been shown that the prediction of pseudoknots in RNA structure prediction is an NP-hard problem (Lyngsø and Pedersen, 2004). However, given the importance of pseudoknots in RNA processing and gene expression regulation (Peselis and Serganov, 2014), several approaches have been developed to predict these motifs. `NUPACK` makes it possible to predict pseudoknots with limited topology in non-interacting strands. `IPknot` (Sato et al., 2011) uses integer programming to determine the maximum expected accuracy structure.

Comparative approaches

The most accurate method for RNA structure is the use of comparative sequence analysis. It is based on the principle that the structure is conserved by evolution; thus, a large set of homologous sequences provides sufficient information to determine the structure. This is due to the rules of base pairing in which any changes in the RNA sequence between two species take place in pairs to maintain the hydrogen bonding. One of the comparative prediction strategies is only to use phylogeny information. For that, `Pfold` (Knudsen and Hein, 1999, 2003) performs a *a posteriori* estimation of the secondary structure using the multiple sequence alignment. In that regard, `PETfold` (Seemann et al., 2008) extends `Pfold` and additionally integrates an energy-based model to predict the folding of multiple aligned sequences. In other approaches, the input sequence and its homologous counterparts are simultaneously aligned and folded. For example, `Dynalign` (Mathews and Turner, 2002a) both discard base pairs that contradict the MFE ensemble or alignment that exceeds the maximum distance. Another method that employs a similar strategy is `LocARNA` (Smith et al., 2010; Will et al., 2007), which maximises base-pairing probabilities and performs a local pairwise structural alignment. In another approach, `RNAalifold` first performs multiple sequence alignments before predicting the MFE structure, which results in a significant speed-up (Bernhart et al., 2008). Another approach is implemented In `RNAshapes` (Steffen et al., 2006), in which the sequences are folded to an abstract representation of an RNA secondary structure and subsequently aligned. In principle, algorithms that predict secondary structures significantly improve prediction accuracy (Puton et al., 2013; Sloma and Mathews, 2015). However, homologous sequences are not always present and are primarily available to known or well-characterised RNAs. In the end, considering multiple approaches for the RNA of interest seems reasonable. Puton et al. (2013) provide a continuous benchmarking of methods for RNA structure prediction to illustrate their performance. Also, Zambrano

et al. (2022) provide a profound review of RNA structure prediction algorithms and lists their accessibility on public web servers.

Guided prediction

A method to improve the accuracy of these algorithms is to incorporate experimental data, thereby guiding the RNA structure prediction. In traditional approaches, so-called hard constraints (Mathews et al., 2004) prevent specific bases from pairing or enforcing certain base pairs, thereby restricting the folding space. However, these approaches are not robust, which means that even a single error in the constraints may distort the results. As an alternative to enforcing these hard constraints, soft constraints guide the folding process by adding specific pseudo-energy terms that are included when evaluating individual structure motifs. In recent years, these soft constraints have gained considerable attention as the reactivities from structure probing experiments have been used in structure prediction algorithms and shown to improve their performance (Sükösd et al., 2013). `RNAfold` implements different algorithms to transform normalised reactivities from SHAPE experiments into meaningful pseudo-energy terms (Lorenz et al., 2016a). Similarly, in `RNAstructure`, SHAPE reactivities are converted to position-specific destabilising energies for base-pair stacks. While all methods incorporating reactivity data result in an improved prediction performance as opposed to no constraints, the results differ for distinctive sets of RNAs.

As of today, `AlphaFold2` and `RoseTTAFold` can accurately predict protein structures given amino acid sequences (Baek et al., 2021; Jumper et al., 2021). In particular, the constant increase of RNA structure data paved the way for the application of deep learning algorithms in the prediction of RNA structure and function. Sun et al. (2017) state that the success of deep learning is mainly attributable to the availability of large-scale annotated data. Yu et al. (2022) discuss successful applications of deep learning in the predictions of RNA structures. In general, using deep learning models makes it possible to neglect the specific features of the RNA structure and instead accept the entire sequence into the model. This is a significant advantage over traditional prediction algorithms. For example, before `RNAfold v2.1`, G-quadruplex structures (GQS) were not supported but gained more attention due to their emerging role in gene regulation. In addition, not all RNA secondary structure motifs are thermodynamically well-characterised. This is the case for pentaloops, for which Saon and Znosko (2022) present a specific thermodynamic model that can be incorporated into RNA structure prediction software. Although there has been some effort in applying deep learning to RNA structure analysis, a significant problem remains the rare training data. This often leads to overfitting, in which the model only works well on the training data. Similarly,

Flamm et al. (2022) argue that the training sequences are unbalanced, comprising a majority of rRNAs or tRNAs with poor performance in other classes. In addition, profound features of the RNA structure, such as multi-loops and pseudoknots, are hard to learn.

1.5.3 RNA-RNA interaction prediction

It seems reasonable to extend the progress in RNA structure prediction towards hybridising two RNA sequences. In traditional RNA-RNA interaction prediction algorithms, intramolecular base pairs were neglected. These are distinguished into sequence- and structure-based approaches. In the former, these merely search for sequence complementarity. For that, performing pairwise sequence comparison makes it possible to determine regions of hybridisation. `BLAST` (Altschul, 2014) and `FASTA` (Pearson and Lipman, 1988) search for stretches of complementarity between a query and target sequence. Similarly, indexing methods such as `REPuter` (Kurtz et al., 2001) and `Vmatch` (Abouelhoda et al., 2002) compute suffix trees/arrays to determine such potential helices but neglect non-Watson-Crick base pairs such as the G·U wobble. Gerlach and Giegerich (2006) introduced GUUGLE, which employs suffix trees and can to handle these G·U base-pairs. However, the absence of a thermodynamic energy model renders these approaches impractical, especially when looking for naturally occurring RNA-RNA duplexes. In any case, sequence-based methods can reduce the search space when incorporating the results into algorithms with more sophisticated models. In that regard, `TargetRNA` (Tjaden et al., 2006) predicts mRNA targets in bacterial sRNAs and includes separate sequencing- and energy-based models. In the individual base pair model, a scoring scheme is applied similar to the one the Smith-Waterman algorith uses. In the stacked base pair model, the minimum free energy is calculated by favouring stacked bases and penalising loops. Similarly, `RNAhybrid` (Krüger and Rehmsmeier, 2006; Rehmsmeier et al., 2004) predicts miRNA targets using an energy-based model. At its core, it is a variation of the classical RNA structure prediction algorithm and, in contrast, determines the most favourable hybridisation site between two sequences. Both `TargetRNA` and `RNAhybrid` restrict the length of structural elements such as loops and bulges. On the one hand, this speeds up the computation but is also helpful when predicting targets (e.g., plant miRNAs) in which only a few nucleotides remain unpaired (Rhoades et al., 2002). In contrast, `RNAplex` (Tafer and Hofacker, 2008) lifts this restriction and uses a simplified energy model in which an affine function scores the energy of loops and bulges. In most of these approaches, an energy model is employed, which provides a reliable estimation for the prediction of RNA-RNA interactions.

Considering internal structures

Rajewsky (2006b) argues that the free energy of the entire RNA duplex is not sufficient as several authors have shown that the secondary structure of the target mRNA (Ameres et al., 2007; Hiller et al., 2007) or the ncRNA (Köberle et al., 2006) affects the target recognition. For that, two different approaches are commonly used. In the concatenation-based approaches, the two RNA sequences are concatenated using a unique linker character. The resulting single sequence is then folded using regular RNA structure prediction algorithms. In principle, the inner workings are identical to single structure prediction, but a few special cases need to be considered separately, such as when the linker falls within a loop. `RNAcofold` (Bernhart et al., 2006), `NUPACK`, and `PairFold` (Andronescu et al., 2005) implement this approach. This has the advantage that concepts from classical structure prediction algorithms can be incorporated into the dimerisation of two RNA sequences. For example, this includes the calculation of base pairing probabilities. However, this also comes with known limitations, particularly in predicting pseudoknotted structures like the kissing loop complexes. For that, accessibility-based methods like `RNAup` (Mückstein et al., 2008) and `IntaRNA` (Busch et al., 2008; Mann et al., 2017; Wright et al., 2014) have been introduced that handle these interactions. Rather than folding the concatenated RNA sequences, the structure ensemble of both sequences is examined individually. In particular, an interaction site needs to be accessible, meaning it is not covered by an intramolecular base pair. This is represented by the free energy required to break up intramolecular base pairs derived from the partition function. In principle, both approaches restrict the considered intramolecular base-pair to a certain degree, as the unrestricted RNA-RNA interaction problem is NP-complete Alkan et al. (2006).

In the concatenation-based approaches, the prediction of intra- and intermolecular interactions are mutually exclusive. In other words, the predicted intermolecular base pairs are not covered by an intramolecular interaction. On the other hand, accessibility-based methods assume single interactions, thereby neglecting bases which are part of an intramolecular pairing. As a result, this fails to predict RNA-RNA interactions involving more than one kissing loop complex, such as between the sRNA *OxyS* and its target *fhlA* (Altuvia et al., 1998; Argaman and Altuvia, 2000). For that, Pervouchine (2004) introduced `IRIS` which makes it possible to predict these structures in a reasonable time using an energy model that maximises the number of base pairs.

Comparative RNA-RNA interaction prediction

Generally, these RNA-RNA prediction algorithms mentioned above exhibit a high false positive rate when applied on a genomic scale. A common approach to improve prediction accuracy is the use of comparative information, similar to RNA structure prediction algorithms. Richter and Backofen (2012) showed that in bacterial sRNA-target interactions, both the sRNA and its target interaction sites are highly accessible, and the interaction sites in sRNAs are highly conserved. An explanation for this is the ability of sRNAs to base-pair with multiple targets, which impairs their evolution. Based on these findings, a strong sequence conversation hints towards a target-binding region, which can be utilised to remove false-positive interactions. It seems reasonable to transfer the established concepts in RNA structure prediction to RNA-RNA interaction prediction. For example, `PETcofold` (Seemann et al., 2011) first scans the alignments of both the ncRNA and target sequence for not accessible regions. In the next step, this information is used to constrain `PETfold` when predicting the structure ensemble of the concatenated sequences. In principle, most target prediction algorithms incorporating sequence conservation have been specifically designed for sRNAs in bacteria. This is probably due to the fact that most experimental verified RNA-RNA interactions with well-characterised structures for both RNAs can be found in bacteria. In addition, bacterial genomes are generally more conserved. In that regard, the `TargetRNA2` (Kery et al., 2014) web server predicts sRNA targets by combining conserved regions with the accessibility of both sRNA and target, as well as hybridisation energy.

In contrast, `CopraRNA` (Wright et al., 2014, 2013b) combines bacterial genome-wide target predictions using `IntaRNA` with information on conserved target genes. In the former, this results in a set of interaction energies for the provided homologous sequences that are subsequently transformed into p-values. Then, the p-values are weighted based on a 16S rDNA phylogenetic tree and combined into a single p-value for the orthologous genes. Pain et al. (2015) assessed available sRNA target prediction algorithms and showed that `CopraRNA` outperforms all the other tools. However, the success of such comparative approaches depends on the availability of conservation data. Freyhult et al. (2006) reviewed methods to detect homologous sRNA sequences, which, in principle, are based on sequence comparisons or combine sequence and structure information. In the former, using the primary sequence, `BLAST` identifies possible homologs in public nucleotide or protein databases. Other methods, such as `Infernal` (Nawrocki and Eddy, 2013), create probabilistic profiles (e.g., covariance models; CMs) to integrate the sequence and secondary structure of RNA families. This aids in detecting the sRNA homologs with low sequence conservation. CMs are built from combined multiple sequence alignments with consensus structure annotation (Nawrocki and Eddy,

2013). In that regard, the Rfam database (Gardner et al., 2009; Kalvari et al., 2021) contains pre-built CM models, which can be used to scan (meta) genomic or transcriptomic datasets using Infernal. Most commonly, RNA homologs are detected by searching for homologous sequences and the subsequent generation of multiple sequence/structure alignments. This is then used to build the CM to detect homologs with high accuracy. However, this is both computationally expensive and requires a basic understanding of the command line from which it needs to be executed (Kalvari et al., 2021). The tool RNAlien automated (Eggenhofer et al., 2016) the construction of CMs starting from a single sequence. It provides a web server to build CMs with similar sensitivity and specificity as manually curated ones from Rfam. However, this still requires using the CMs to search for homologs in sequence databases. GLASSgo (Lott et al., 2018) aims to close this gap in bacteria and provides an automated workflow based on an iterative BLASTn strategy with pairwise identify filtering. However, the sRNA sequences are not always conserved, and algorithms based on sequence data alone are still needed.

1.6 High-throughput methods

Although interaction prediction algorithms are constantly increasing in prediction accuracy, experimental methods are still necessary to determine RNA-RNA interactions. In the classical approach, the fact that target mRNAs are rapidly degraded following the expression of sRNAs is adopted. The levels of cellular mRNAs are monitored after the expression of regulatory sRNAs using microarray or RNA-seq. Most commonly, this corresponds to a positive regulation, which is associated with an increase in mRNA stability and expression levels (Nouaille et al., 2017). However, these methods are restricted to known RNAs with limited throughput. In recent years, methods have been established that allow mapping RNA duplexes on a global scale. A drawback of these methods is that ncRNAs also exert their function following cell lysis, which makes it hard to elucidate the interactions and how they occur *in-vivo*. In that regard, CLIP (crosslinking immunoprecipitation) is based on the UV-crosslinking of proteins to fixate RNA-protein complexes *in vivo*, followed by immunoprecipitation. This reveals the RNAs directly bound by the protein. HITS-CLIP or CLIP-Seq utilise this protocol and combine it with high-throughput sequencing to generate genome-wide RNA-protein interaction maps. RNA-binding proteins bind both single- and double-stranded RNA, allowing to capture tripartite RNA-RNA-protein complexes. Consequently, profiling of protein-RNA interactions can also detect the corresponding RNA-RNA interaction (Sanford et al., 2009). Similarly, crosslinking and analysis of cRNA (CRAC) is an advanced CLIP approach identifying RNA-protein complexes. Granneman et al. (2009)

used this approach in yeast to map the binding sites on Nop1, Nop56, Nop58 and Snu13 on U3 snoRNA. In another study, Bohnsack et al. (2009) used CRAC to identify binding sites of Prp43 on pre-rRNA. It was discovered that a multitude of chimeric cDNAs contain regions of the snR52 box C/D snoRNA fused to an rRNA region. This led to the assumption that base-paired RNA molecules could also be ligated together, thereby generating chimeric RNAs. Kudla et al. (2011) re-analysed the CRAC datasets and searched for chimeric reads. These are composed of consecutive fragments that could be mapped to distinct locations of the genome, associated with different RNA molecules or to distinct regions of the same molecule. In total, 0.46% of all reads were identified as those. The protocol with the sole aim of generating RNA-RNA interactions was henceforth referred to as crosslinking, ligation, and sequencing of hybrids (CLASH). Since then, multiple studies have been conducted to provide a global view of the RNA interactome. Helwak et al. (2013) used the CLASH approach to identify miRNA-mRNA interactions associated with AGO1 in humans. Several studies, e.g., using the RNA chaperone Hfq (RIL-Seq; Melamed et al., 2018, 2016), RNase E (RNase E-CLASH; Waters et al., 2017) or ProQ (RIL-Seq; Melamed et al., 2020) were performed in *Escherichia coli*. However, in a typical CLASH experiment, only $\sim$ 1% of the sequencing reads provide information about RNA-RNA interactions (Waters et al., 2017). A more holistic approach was proposed with the concept of RNA Proximity Ligation (RPL) (Ramani et al., 2015). In order to capture *in vivo* RNA-RNA interactions, the biochemical reactions are carried out in the crude cell extract. First, ssRNAs are depleted by Nuclease digestion, RNA duplexes ligated, the so-called Proximity Ligation step, and subsequently sequenced. Chimeric reads containing the inter- and intramolecular interaction partners are detected bioinformatically to decipher the RNA-RNA interactome. Recently, the RPL approach has been extended by Psoralen-mediated crosslinking and adapted independently to human, mouse and yeast in different studies, termed Direct Duplex Detection (DDD) methods (Weidmann et al., 2016): LIGR-Seq (Sharma et al., 2016), SPLASH (Aw et al., 2016) and PARIS (Lu et al., 2018, 2016). In addition, DDD experiments have been performed in *E. coli* (Liu et al., 2017), referred to as mCLASH. The methods differ in the experimental protocols as reviewed in Schönberger et al. (2018), and also in their bioinformatics analyses, although the input data is similar, namely sequencing reads with a fraction of chimeras. According to Schönberger et al. (2018), the latter is in the range of $\sim$0.5%-3.9%. Nevertheless, these were subjected to a rigorous analysis that overlooks many interactions within the reads. It is, therefore, of interest, to focus on the data analysis of RNA-seq data to utilise all interaction information.

1.7 RNA-seq data analysis

In the last decade, RNA-seq emerged from advancements in sequencing technologies and has become the most widely used method for transcriptome analysis. In principle, first-generation sequencing typically refers to the Sanger dideoxy sequencing. It is based on the detection of labelled chain-terminating nucleotides incorporated by a DNA polymerase during the template replication. In second-generation sequencing, also known as next-generation sequencing technologies, similar sequencing by synthesis chemistry is used but performed in a massively parallel format. Finally, third-generation sequencing are methods that use sequencing by synthesis chemistry, but have templates of DNA or RNA molecules. In principle, the experimental setup of the sequencing procedure involves the isolation of RNA from cells or tissue, preparation of the sequencing library to represent the RNA transcripts, chemical sequencing of the library, and subsequent bioinformatics analysis. RNA-seq has many applications leading to many different directions in data analysis.

1.7.1 Pre-processing

In a typical RNA-seq experiment, quality control is critical for the subsequent analysis. It aims to reduce low-confidence bases, PCR artefacts, untrimmed adapters, and contaminated sequences. This results in trimming and filtering of the reads. The tools `FastQC` (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) and `PRINSEQ` (Schmieder and Edwards, 2011) provide quality metrics and report the read quality combined with informative visualisation. It should be checked before and after pre-processing. In another matter, PCR duplicates should be removed, although this can be neglected in regular RNA-seq datasets. More importantly, de-duplication is hard to achieve without *Unique Molecular Identifiers* (UMIs) as these make it possible to remove reads sharing the same UMI, hence deriving from the same input molecule. This is of particular importance in single-cell RNA-seq. For example, Sena et al. (2018) discovered that reads with identical UMIs map to different but adjacent positions on the genome. This affects such studies with both false positives and negative results. However, in a typical RNA-seq experiment, attaching UMIs is rather the exception. Typically, without UMIs, PCR duplicates are identified by simple sequence comparison or by matching the alignment coordinates. Parekh et al. (2016) argue that specifically in DEA studies, removing these duplicated sequences will also remove valid biological data, and, therefore provide no benefit for the accuracy or precision and, worse, potentially skew the result. It should also be considered that the library preparation itself introduces various biased steps, such as fragmentation or ligation, whose removal would distort the data. Also, in organisms with poor genome complexity, a few transcripts

dominate the sequencing results. However, UMIs or removal of PCR duplicates, in general, is recommended when the libraries have been sequenced very deep or are of low sample size. For that, `Dedupe` from `BBtools` (https://github.com/BioInfoTools/BBMap) is able to remove PCR duplicates from raw sequencing reads. Other tools that operate on aligned reads and utilise the mapping coordinates for de-duplication include Picard MARKDUPLICATES (https://broadinstitute.github.io/picard/), `EAGER DeDup` (Peltzer et al., 2016) and `SAMtools rmdup`. Fu et al. (2018) examined the transcript abundance by removing PCR duplicates using only mapping coordinates compared to mapping coordinates and UMIs. It turns out that most reads mapped to identical coordinates originate from different molecules when inspecting the UMIs and should be accounted for in the transcript abundance. This is further enhanced in small RNA-seq data, in which 56-76,8% of the reads were flagged as PCR duplicates, but only 1.05-13.6% identified as such using UMIs. Moreover, a similar bias can be observed when considering transcript quantification, in which the abundance of short and highly expressed transcripts is underestimated. It must be kept in mind that these results are affected by the alignment procedure, in which a more tolerant approach groups reads that are not completely identical. Consequently, sequence comparison on the raw reads to identify PCR duplicates seems to be less prone to ambiguous alignment results. Subsequently, the reads are subjected to filtering and trimming procedures.

A variety of tools exist to pre-process raw RNA-seq data. Most notably, these methods are based on the idea of searching for adapter sequences using a semi-global alignment such as `Cutadapt` (Martin, 2011) or `Flexbar` (Dodt et al., 2012). In contrast, `BBDuk` operates on k-mers to search for adapter sequences. However, in the case of paired-end reads, this involves merging forward and reverse reads. `Pear` (Zhang et al., 2014) is able merge these reads of varying length. In principle, all overlaps between forward and reverse reads are assessed using an alignment score that is calculated with a scoring matrix (match: 1, mismatch:-1). In addition, a statistical test further removes false positives. `SeqPrep` combines pre-processing and merging of paired-end reads in its workflow.

1.7.2 Sequence alignment

In the next analysis step, the goal is to find the point of origin for every read. If a reference genome or transcriptome is not available, then the reads can be mapped against the transcriptome. For that, the transcript sequence must be built from the overlapping read sequences. For that, two different strategies are commonly used. If a reference genome is available, its information can be utilised to guide the transcriptome assembly. In another approach, the *de novo* assembly, the reads are assembled without the use of any external information.

In the latter, methods for *de novo* assembly from sequencing data are classified based on their underlying method, namely overlap consensus layout (OLC) graph (Batzoglou et al., 2002), *de Bruijn* graph (Compeau et al., 2011), and string graph (Idury and Waterman, 2009). Liao et al. (2019) reviewed the aforementioned frameworks for *de novo* assembly algorithms and the challenges associated with it. In principle, numerous tools are able to align the sequencing reads against a reference genome. Fonseca et al. (2012) comprehensively classify available aligners using different properties. As the alignment procedure is computationally intensive, the alignment programs typically use certain indexing schemes for the reference genome or transcriptome to speed up the execution time. One of the main considerations when aligning reads against a reference is whether or not the organism contains introns. If this is not the case, continuous aligners such as `Bowtie` (Langmead, 2010), `Bowtie2` (Langmead and Salzberg, 2012), and `BWA` (Li and Durbin, 2009) are sufficient. In any case, these aligners can also be used when aligning against the transcriptome. However, if the reads are aligned against genomes containing introns, spliced aligners such as `segemehl` (Hoffmann et al., 2009; Otto et al., 2014), `TopHat` (Kim et al., 2013) or `STAR` (Dobin et al., 2013) need to be used.

1.7.3 Quantification & differential expression analysis

Once the reads have been mapped to the reference genome, their genomic position can be matched with genomic annotations. This allows for the quantification of gene expression, which is done by simply counting reads per gene, transcript, or other features. In that regard, `featureCounts` (Liao et al., 2014) and `htseq-count` (Anders et al., 2015) are tools most commonly used for gene expression quantification. However, the number of reads that are generated by each transcript needs to be taken into account. For that, the quantification tools either report the abundances in a table of raw counts or normalise them towards specific factors. Subsequently, this can be subjected to other downstream analyses, such as comparing the expression between conditions as in differential expression analysis. For that, Zyprych-Walczak et al. (2015) compared different normalisation methods and their impact on the results of the differential expression analysis. In the differential expression analysis (DEA), features that are expressed in significantly different quantities in distinct groups of samples are identified. These features usually include genes, but other genomic features, such as transcripts or exons, are also of interest. In this work, the interest lies in RNA-RNA interactions between biological conditions (treated vs. controls). It is clear that these features are not independent of one another, and a DEA is done in a univariate way. This is because the number of samples is much smaller than the number of features, making it hard to fit a statistical model that considers all features as a whole. Another thing

that needs to be considered in the DEA is the replication. It is considered as one of the core points of proper experimental design, as outlined, by Basu (1980). Its purpose is to estimate the variability between and among groups. In that regard, technical replication is used to estimate the variability of the measurement. Similarly, biological replicates are used to find the variability within a biological group. This means that the change in the expression between two groups is only significant if the difference between the groups is large compared to the variability within the group while taking the sample size into consideration. Expression levels of genes have been shown to follow a log-normal distribution as measured by quantitative PCR (Bengtsson et al., 2005). A broad distinction between models for DEA are parametric and non-parametric models. In the former, the basic idea is to model the expression of each gene as a linear combination of some explanatory variables. Love et al. (2014) introduced DESEQ2, which is widely used to perform DEA using this approach. In contrast, non-parametric models do not assume anything about the underlying distribution and instead rank the result using statistical analyses.

1.8 Data warehousing

The advent of the 'omics' sciences and the emergence of high-throughput techniques led to a comprehensive understanding of complex biological processes. These technologies generate large data sets, and their interpretation remains a challenge in modern science. For example, next-generation sequencing (NGS) captures millions of sequences, often in a single experiment. As of today, a wide range of bioinformatics tools and workflows are available to analyse the raw data from such technologies and return the results in more interpretable forms. This knowledge is typically stored in scientific databases and other repositories that need to be readily accessible for further analysis. However, these tools often need ways for deeper scientific interpretation of the data, such as correlation with data coming from other platforms. This means that for a given task, researchers need to browse numerous databases and web servers. It is, therefore, of interest to integrate heterogeneous biological data from multiple sources (Rubin et al., 2008). This becomes apparent when considering that genomics generates a similar data volume as other Big Data sciences in terms of data acquisition, storage, distribution, and analysis (Stephens et al., 2015). However, this is challenging due to the heterogeneous nature of the data and their diverse access methods and formats. For that, a data warehouse unifies the accessible data from various sources and has been widely adopted. In contrast to decentralised approaches, a single access point maintains the control of various data, which allows to define use cases for the user's requirements. Similarly, it improves the query performance as these are executed within the data warehouse,

thereby achieving fast response times (Zhang et al., 2011). However, the data needs to be continuously updated to keep up with changes in the source data. Kormeier (2014) provides an in-depth review of the integration of biological data into data warehouses.

1.8.1 Database models

In its simplest form, a database is a collection of information stored in a computer-readable format. As new biological databases are consistently introduced, it is important to consider more elaborate structures to store an increasing volume of biological data.

Relational databases

Codd (1970) proposed the concept of a relational database based on the relational data model that has been widely adopted ever since. At its core, data is organised into tables, also termed relations, consisting of rows and columns with a unique key that identifies each row. Other objects include procedures, functions, and views. The rows in tables are then linked to rows in other tables by adding their respective key in a column (known as foreign keys). Consequently, this results in either one-to-one, one-to-many, or many-to-many relations representing relationships of arbitrary complexity. The structure of a relational database is described in the schema that defines the objects and their relationships. Having a defined rigid schema for a database guarantees performance and scalability. However, changes to the predefined schema may disrupt the functionality. Moreover, these modifications are usually difficult and resource-intensive. The *Structured Query Language* (SQL) is commonly used for querying the database. Initially, SQL databases were thought for general purposes focusing on reducing data redundancy, a process known as database normalisation. However, in an extensive database, the lookup between numerous tables can slow down the overall processing of the data. SQL databases are highly structured, which allows them to follow different standard properties, known as the ACID principle. This keeps the tables synchronised and guarantees the validity of transactions. SQL databases are best suited when applications have no room for error and need high data integrity. However, these were initially developed for single servers. Consequently, its scaling happens vertically while incrementing the server's hardware capabilities. Most commonly, SQL databases that are widely used include the open-source MySQL (https://www.mysql.com/) database, Oracle Database (https://oracle.com), and PostgreSQL (https://postgresql.org). In the last decade, many databases emerged that followed a non-relational approach and were hereafter referred to as NoSQL databases. As of today, the market is still dominated by SQL databases, but NoSQL is gaining more interest, especially in big data applications. In that regard, NoSQL

offers a variety of benefits to the classical SQL approach. The data model is usually very flexible as there is no rigid database schema. It allows changes to the database to be made as the requirements change. Moreover, these databases can be scaled up horizontally, meaning that commodity servers can be added when needed. Most importantly, the queries can be much faster in large databases. This is due to the fact that the data is already optimised for the queries. In addition, the underlying data structure is usually optimised, which allows the integration of multiple APIs in a straightforward manner. However, most NoSQL databases lack true ACID transactions in favour of scalability and resilience. As NoSQL is an umbrella for this new generation of databases, the underlying model categorises these roughly into key-value, document, or graph databases.

NoSQL databases

Key-value databases store the data as a collection of values to a unique key. This data concept is well established in many programming languages, known as *associative array*, *dictionary*, or *hash*. The advantages of key-value databases are high-performance and flexible scalability. Both result from the simple structure of the model. Since the key-value store does not require or prescribe a uniform schema, changes to the database can be made on the fly. For example, a new field can be introduced while actions occur in other entries at the same time. The high speed of this database model is made possible by its simple link between key and value. This means that information can be retrieved by accessing the value directly via the specific key. The data is directly available. However, this is also one of the disadvantages of the key-value store because no other access method is actually provided. Relational databases, in particular, allow complex queries. The content of such databases can be searched for in different ways. In a key-value store, in contrast, only access via the key is planned. Further indices and search options usually have to be dispensed with. Areas of application for key-value stores result from both the advantages and limitations of databases. Key-value databases are used whenever fast access times are required for large amounts of data. For example, Redis (https://redis.io/) implements the key-value model in which the entire data is loaded into memory, resulting in fast performance but ultimately limiting its size to the available memory (Han et al., 2011). In contrast, document-based databases store data as objects that are in JSON, YAML, or XML format. In principle, their model is the same: There is a key to which a value belongs. In fact, the boundaries are fluid and not always clearly distinguishable. However, the document store is designed to simplify its integration into modern programming languages. As with the other NoSQL data models, no schema is specified here. The documents can, therefore, be designed and supplemented as desired without such changes being made known to the system beforehand. Although no structure is

defined, and each document can be structured completely differently, as a rule, one will not create fields indiscriminately but will follow a certain (indirect) scheme corresponding to the application to create a prerequisite for meaningful queries afterwards. Since document-oriented databases, similar to key-value databases, have a very general data model, they are certainly versatile. Commonly used database systems based on the document store include MONGODB (https://mongodb.com) and Apache COUCHDB (https://couchdb.apache.org/). Finally, databases with an underlying graph model use the graph structures to represent the data as it is conceptually viewed. In this database model, the nodes stored data, and edges represent the relationship between the nodes. Unlike traditional models, graph databases also rely on algorithms from graph theory to simplify complex data queries. This allows for combining storage with network analysis. This concept is mainly used for heterogeneous data that is highly connected and semi-structured (Henkel et al., 2015). It allows for depicting all relationships within a large-scale dataset, making it helpful for representing complex linked data. In that regard, Neo4J (https://neo4j.com/) has been widely used in different applications. However, in some instances, a single model is not sufficient. For example, the underlying graph structure describes the relationships within the data but requires additional document storage to describe the data comprehensively. For that, multi-model databases were introduced, which integrate multiple data models in one core system using a single unified query language. This allows to combine different models in a single query. Most notably, ARANGODB (https://arangodb.com/) and ORIENTDB (https://orientdb.org/) are widely used databases that implement all the above-discussed database models.

1.8.2 Biological networks

Upon integrating multiple biological data sources, the data warehouse contains related data. However, biological entities (e.g., RNAs, proteins) do not act in isolation but rather interact with one another, which is hard to deduce from the data alone. It is useful to consider the biological data as networks with connections between nodes, and they are predominantly represented as graphs (Pavlopoulos et al., 2011). This means that the nodes are representations of the biological molecules, and edges connect the nodes depicting some kind of interaction (e.g., activation, inhibition). Moreover, a multi-edge connection consists of two or more edges with the same start- or endpoints. This can occur in complex networks with multiple layers of information in which linkage by more than one connection indicates a different relationship. In scientific literature, the terms *networks* and *graphs* are used interchangeably. Zhu et al. (2007) provide an overview of some of the major biological networks. Formally, a graph G is defined as a pair (V,E), where V is a set of *vertices* also known as *nodes* and E is a set of *edges* between the *vertices*, also referred to as *links*. For that, $E = \{(i,j)|i,j \in V\}$

corresponds to a single connection between nodes i and j. Similarly, a directed graph is an ordered triple $G = (V, E, f)$, where f is a function that maps each element in E to an ordered pair of vertices in V. In that regard, directed graphs are best suited to represent biological pathways that show interaction flow through the network. Most commonly, these include metabolic signal transduction pathways or regulatory networks.

RNA-RNA interaction networks

As of today, a variety of RNA-RNA interaction databases have been published. Most notably, these are restricted to a specific type of transcript and/or stored in a simple tab-delimited format. For example, in sRNATarBase (Wang et al., 2016), the authors manually collected sRNA-target interactions from published studies and sRNA target predictions. In *E. coli*, this results in 403 sRNA-target interactions (50 distinct sRNAs) that were experimentally verified. Additional 368 sRNA-target interactions are distributed among 50 other bacterial strains. This is complemented with additional information such as binding regions, regulatory networks, and pathway annotations. sRNATarBase uses a MySQL database to store the entries. It implements a PHP front-end to browse the data and allows them to be exported in CSV format. Similarly, miRTarBase v8.0 (Huang et al., 2020) contains manually curated miRNA-target interactions. In total, 479,340 miRNA-target interactions (4,312 distinct miRNAs) are supported with experimental evidence. 65,331 miRNA-target interactions are supported by so-called *strong* evidence, which means that they are validated by western blot, qPCR, or reporter assays. In addition, a variety of other databases and tools were integrated into miRTarBase. Most notably, these include miRNA information from miRBase (Kozomara et al., 2019), TransMiR (Tong et al., 2019), MiRSponge (Wang et al., 2015), SomamiR (Bhattacharya and Cui, 2016), and target information from NCBI Entrez (Maglott et al., 2011) and RefSeq (Pruitt et al., 2005). Other sources include disease information from the Human MicroRNA Disease Database (HMDD) (Huang et al., 2019; Li et al., 2014), expression profiling from Gene Expression Omnibus (GEO) (Barrett et al., 2013), The Cancer Genome Atlas (TCGA) (Tomczak et al., 2015), and Circulating MicroRNA Expression Profiling (CMEP) (Li et al., 2019). In a similar manner, snoDB (Bouchard-Bourelle et al., 2020) unifies annotations from RefSeq, Ensembl (Yates et al., 2020) and RNAcentral (RNAcentral Consortium, 2021).

1.8.3 Visualisation of interaction networks

Complex interaction networks are hard to interpret by looking at the data alone. For that, visualisation of these networks/graphs provides a bird's-eye view of the system. As of today,

various desktop-based applications that can visualise large-scale graphs exist. Most notably, Cytoscape (Shannon et al., 2003) has been introduced as a general software platform for visualising molecular interaction networks. It is used in a wide range of life science applications. Other tools include GEPHI (Bastian et al., 2009) and TULIP (Auber, 2004). In addition, the Graphviz software package provides a comprehensive collection of open-source tools for the visualisation of graphs based on the Dot graph description language (Jünger and Mutzel, 2012). It is widely distributed because it can be called from general-purpose languages via specific interfaces. The emergence of the internet has led to the development of Rich Internet Applications. These applications deliver a dynamic and engaging user experience across different platforms using standard web browsers. This paradigm is particularly advantageous for visualising large-scale graphs, as modern web development technologies enhance both interactivity and scalability. Touchgraph© (Touchgraph, LLC, USA), Tom Sawyer Visualization©(Tom Sawyer Software, USA), and ManyEyes (Viegas et al., 2007) follow the rich-internet paradigm and can visualise graph data in a sophisticated manner but are still lagging behind the possibilities offered by current web standards and are, beyond that, not optimised for biological networks. Similarly, Cytoscape Web (Lopes et al., 2010) is a web-based network visualisation tool that is modeled after Cytoscape. Hyperscape (Cromar et al., 2015) implements hypergraphs to capture complex hierarchical structures but has its limitations in generalising biological networks. Nevertheless, hypergraphs provide an interesting concept in visualising and analysing biological networks. In recent years, several JavaScript libraries emerged that enable to manipulate the Document Object Model (DOM) created by a browser when rendering a document such as an HTML page. These use Scalable Vector Graphics (SVG), HTML5, and Cascading Style Sheets (CSS) to modify the content when selecting elements within the DOM. Most notably, the Data-Driven Documents (Bostock et al., 2011) library, referred to as d3.js also allows visualising graphs using different drawing layouts. Similarly, cytoscape.js (Franz et al., 2016) can visualise graphs and integrate them into web user interfaces by incorporation of graph elements into the DOM. It borrows several concepts from the Cytoscape application, allowing a similar integration. Consequently, it includes graph algorithms that are missing in d3.js. Other libraries that allow the visualisation of large graphs, include sigma.js (https://sigmajs.org/), but is not particularly suited for biological networks.

Data formats

As of today, a variety of established formats that describe biological networks. The *Systems Biology Markup Language (SBML)* (Hucka et al., 2003) is based on XML and makes it possible to store computational models of biological processes. This includes cell-signalling

pathways and metabolic as well as regulatory networks. However, it serves as a format for computational models and is not intended to be human-readable. Other formats that can represent biological networks include *Proteomics Standard Initiative Interaction (PSI-MI)* (Hermjakob et al., 2004), *Chemical Markup Language (CML)* (Murray-Rust et al., 2001) or *BioPAX* (Cary, 2007). Other file formats exist that simply describe the network. For example, the *Simple Interaction File (SIF)* format contains lists of interactions that include the type of interaction but no additional information. Similarly, the *Nested Network Format (NNF)* is a simple interaction format that unlike SIF makes it possible to specify nested networks for each node. In contrast, the *Graph Modeling Language (GML)* is designed to represent arbitrary structures. It employs a hierarchical ASCII-based structure with non-defined attributes making it widely applicable. *GraphML* (http://graphml.graphdrawing.org/) allows in particular to specify graphs and is based on XML. It contains a `graph` element that, in turn, includes sequences of `node` and `edge` elements with distinct attributes. The *JavaScript Object Notation (JSON)* is a universal data interchange format that has been widely used in different applications and can also be used to describe graphs. It is human-readable and uses key/value pairs and arrays in JavaScript syntax to describe the data. It is language-independent and can be easily parsed in multiple programming languages. In recent years, there have been efforts to create a JSON graph specification to standardise the description of graphs using the JSON Schema. For that, the *JSON Graph Format (JGF)* (https://jsongraphformat.info/) has been introduced, which captures the basic graph structure. At its core, it specifies a `graph` object that represents a single conceptual graph and contains objects for the nodes and edges. It has certain predefined key/value pairs but also allows the specification of user-defined values, thereby allowing the specification of metadata objects. This makes it the most comprehensive data format that is able to capture graphs in full.

1.9 Structure of the thesis

RNA-RNA interactions provide an additional layer of post-transcriptional regulation and are important for gaining a comprehensive overview of the cell's inner workings. This chapter explained the general methodology of RNA-based regulation and outlined the current experimental and computational methods to infer those. Chapter 2 lists the libraries, external programs, and data that has been used in the development of the tools described in Chapter 3, which is divided into three parts. At first, a workflow is described that combines data from differential expression analysis on RNA-seq experiments with comparative prediction algorithms. Here, part of this work was the development of `GLASSgo` (Lott et al., 2018), which allows the prediction of homologs from a single sRNA input sequence. An updated version

was then introduced and incorporated into Galaxy (Schäfer et al., 2020b). Subsequently, the homologous information can be used for the target prediction using CopraRNA (Wright et al., 2013b). In another approach, the focus lies on RNAnue (Schäfer and Voß, 2021), which provides a comprehensive workflow for the analysis of DDD methods from raw sequencing data. Finally, this chapter is concludes with a prototype of a data warehouse that allows the storage of these RNA-RNA interactions. In that regard, this work involved the development of VisualGraphX (Schäfer and Voß, 2016), which makes it possible to visualise these interactions as large-scale graphs and has been incorporated into Galaxy as well. In Chapter 4, the developed methods are discussed and compared with other approaches. Finally, Chapter 5 outlines their limitations and provides ideas for further improvement of the methods and additional application areas.

Chapter 2

Materials and methods

2.1 Programming languages & libraries

RNAnue complies with the C++17 standard and has been compiled with GCC v9.30, which has full C++20 support. CMake v3.19.6 (Martin and Hoffman, 2007) was used for build automation, testing, and installation. Boost v1.75 (https://www.boost.org) was used for tasks involving usability, logging, and data preparation: ProgramOptions, Filesystem, Log, PropertyTree. In addition, SeqAn v3.02 (Reinert et al., 2017) was used for input/output operations of sequence files. This includes the modules Alphabet, IO, and Range. For multiprocessing, OpenMP v5.0 (https://www.openmp.org) was used. The visualisation with VisualGraphX makes use of the D3 (Data-Driven Documents) JavaScript library (https://d3js.org). Other dependencies include defiant v1.2.5 (https://www.defiantjs.com), which allows querying large JSON structures and select2 v3.5.3 (https://select2.org) to enhance basic select boxes. Filesafer (https://github.com/eligrey/FileSaver.js) enables saving files on the client side. Other scripts for the benchmarking procedures were implemented using Python 3.x and awk. In R v3.6.0, the ggplot2 v3.3.2 package was used to visualise the data and for the differential expression analysis DESeq2 v1.31.16 was used.

2.2 Packages & external programs

FastQC v0.11.8 (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) was generally used before and after pre-processing of all data sets. In addition, BBtools dedupe v38.98 (https://sourceforge.net/projects/bbmap/) and SAMtools markdup v1.11 (Danecek et al., 2021) were used to remove the PCR duplicates. In RNAnue, the initial split read mapping was done using Segemehl v0.3.4. Hybridisation energies and probabilities were

calculated using the ViennaRNA package v2.4.14. In particular, the routines corresponding to RNAcofold and RNAfold were used. Other tools/methods were used to benchmark the results, which, in turn, use different external programs. These include Aligater, which comes with LIGR-seq (https://github.com/timbitz/aligater) and custom scripts for SPLASH (http://csb5.github.io/splash/) and PARIS (https://github.com/qczhang/). These in turn require preprocessing of the data with Trimmomatic v0.36 (Bolger et al., 2014), Flexbar v3.5.0 (Dodt et al., 2012) and SeqPrep (https://github.com/jstjohn/SeqPrep). Similarly, the alignment tools that were used by these methods include Bowtie2 v.2.2.9 (Langmead and Salzberg, 2012), STAR v.2.7.5a (Dobin et al., 2013), and Bwa-mem v.0.7.15 (Vasimuddin et al., 2019). GLASSgo v1.5.3 makes use of BLAST v2.2.30+, Clustal Omega v.1.2.4 and RNApdist v.2.4.14. Containers images of both GLASSgo and RNAnue were created using Docker v19.03.5 (https://www.docker.com). In that regard, the GLASSgo container was used to integrate GLASSgo into Galaxy (Afgan et al., 2018). Here, Galaxy v19.10 was used. DockerHub was used as continuous integration (CI) with Docker Compose, enabling automated build and testing of the containers. IntaRNA v2.4.1 and CopraRNA v2.1.4 were used for the validation of the workflow. Initially, VisualGraphX was integrated into Galaxy v15.10, but it also works until v19.10. For that, Grunt (https://gruntjs.com/) was configured to automate its installation on a Galaxy instance with admin privileges. ArangoDB v3.9.2 was utilised to store the data of RNA-RNA interactions.

2.3 Genomes, annotations, databases

The databases LNCipedia 5 (Volders et al., 2019), snoDB (Bouchard-Bourelle et al., 2020), and miRTarBase 7.0 (Chou et al., 2018) were used. A current version of the NCBI 'nt' database (November 2020) was used in the analysis with GLASSgo. The following reference genome sequences from NCBI RefSeq (Pruitt et al., 2005) were used: human genome release GRCh38.p13 (RefSeq assembly: GCF_000001405.39), mouse genome release GRCm38.p6 (RefSeq assembly: GCF_000001635.26), and genome release *S. cerevisiae* S288C (RefSeq assembly: GCF_000146045.2). *E. coli* MG1655 (NC_000913) was then used to create artificial reads to assess the PCR duplicates.

2.4 Data

In this thesis, the following method-specific data sets were used: LIGR-Seq (GEO:GSE80167), SPLASH (SRA:PRJNA318958), PARIS (GEO:GSE74353), and MCLASH (SRA:SRP103891). These include experiments in the human embryonic kidney (HEK) 293T cells

(LIGR-Seq, PARIS), HeLa cells (PARIS), Lymphoblastoid cells, and human embryonic stem (hES) cells as well as retinoic acid (RA) differentiated ES cells (SPLASH). It is to be noted that the SPLASH data sets have been pre-processed with `SeqPrep` using undisclosed parameter settings. Nevertheless, the intrinsic pre-processing of `RNAnue` was also used for these. Furthermore, the data from wild-type and a Prp43 helicase mutant of *S. cerevisiae* (SPLASH) and mouse embryonic stem (mES) cells (PARIS) were analysed. Benchmarking of the alignment tools were based on a dataset from a study by Seo et al. (2014).

Chapter 3

Results

This chapter covers methods developed in this thesis for detecting RNA-RNA interactions and consists of three parts. At first, the focus lies on a workflow that combines comparative prediction algorithms with correlation measures from RNA-Seq differential expression analyses to detect RNA-RNA interactions. The subsequent section explores a data-driven method which infers RNA-RNA interactions from specific RNA-Seq experiments. Finally, this chapter concludes with the downstream analysis, which allows the visualisation of RNA-RNA interactions and their data storage.

3.1 Prediction of RNA-RNA interactions

The following section covers the workflow for detecting RNA-RNA interactions using comparative prediction algorithms. It revolves around `GLASSgo` (Lott et al., 2018), which detects sRNA homologs from a single input sequence. Part of this work involved the benchmarking procedures and the visualisation of the taxonomic tree. `GLASSgo` is now part of the Freiburg RNA tools (Raden et al., 2018) and Wright and Georg (2018) introduced it into their computational analysis for sRNA candidates. At first, the general workflow of `GLASSgo` is recapitulated. This is followed by the latest update of `GLASSgo` (Schäfer et al., 2020b), which was integrated into the workflow management system `Galaxy` (Afgan et al., 2018). Finally, the general workflow is presented, which combines comparative algorithms - in this case, `GLASSgo` and `CopraRNA` (Wright et al., 2013a), with RNA-Seq expression data.

3.1.1 Detection of sRNA homologs

`GLASSgo` performs a low-stringency iterative `BLAST` search (Altschul, 2014) against the `BLAST` nucleotide database. As the local hits need to be the same length as the query, the hits

are extended on both sides if necessary. This is done using local pairwise BLAST alignment. Next, the BLAST search is performed again using chosen hits that have varying degrees of sequence identity to the input sRNA, thereby increasing the sensitivity. Subsequently, graph-based clustering is performed, which includes the conservation of RNA secondary structures. Finally, the clusters that contain either the query sequence and/or perfect homologs are selected, and all enclosed sequences are reported. These main steps are illustrated in Figure 3.1. Initially, GLASSgo was made available via an easy-to-use web server. In its latest version,

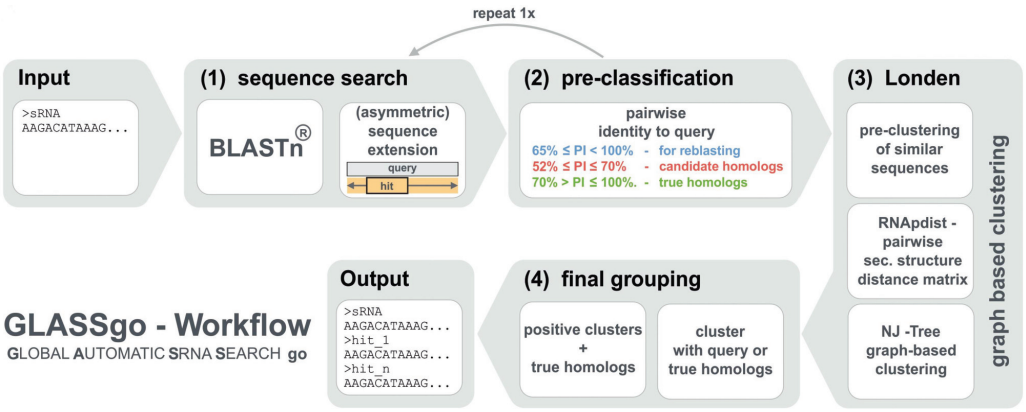

Fig. 3.1 Schematic overview of GLASSgo (Lott et al., 2018)

GLASSgo v1.5.2 can now report upstream regions of the found sRNA homologs. It also supports the use of NCBI accession numbers as unique identifiers. This makes it possible to restrict the search space to specific clades, which often achieve better results at shorter runtimes than unrestricted analyses. For the taxonomic classification, specific lookup tables are required, which have been stored in an open-access repository to ensure effortless retrieval for existing and new installations (Zenodo: https://zenodo.org/record/1320180). GLASSgo, with all its dependencies, is distributed via Docker Hub (https://hub.docker.com/r/lotts/glassgo). The container is built automatically from the source and subjected to automated tests upon new releases. Here, the build process is tested and includes functional tests of GLASSgo for different use cases. This Docker container can also be used on the command line, which allows its integration into custom analysis pipelines. In addition, GLASSgo has been integrated into the workflow management system Galaxy (Afgan et al., 2018). For that, the Galaxy ToolShed (Blankenberg et al., 2014) was used, allowing seamless installation of GLASSgo (https://toolshed.g2.bx.psu.edu/view/computationaltranscriptomics/glassgo) on custom-hosted Galaxy instances. This comes with comprehensive documentation of the installation and usage of GLASSgo and functional tests to ensure the correct installation. In addition, instruction videos guide through the installation and setup process and as its usage (Schäfer et al., 2020a). GLASSgo is also part of the RNA workbench, which provides a public Galaxy

instance with a set of tools for RNA-related tasks and is available at https://rna.usegalaxy.eu. The automatic interplay of GLASSgo with the local Galaxy environment and external web

Fig. 3.2 Galaxy integration scheme of GLASSgo v1.5.2 (Schäfer et al., 2020b)

resources for the Docker container and the lookup tables, which are hosted on Zenodo, is shown in Figure 3.2. In this context, the user interface follows the design of the GLASSgo web server. The starting point is the sRNA sequence of interest that has to be uploaded to the user's history in FASTA format. GLASSgo relies on BLAST, thus, a fundamental parameter is the database to search in. Most Galaxy instances will have a set of databases already available for standard BLAST searches, and GLASSgo can use the same databases for its tasks. If the user wants to use a specialised database, e.g. a clade-specific or a custom database, GLASSgo within Galaxy offers two options: First, the user can choose a clade to restrict the BLAST searches, which is achieved with the aforementioned lookup tables. Second, users

can use custom BLAST databases, for example, created from sequences in their own Galaxy history.

3.1.2 Workflow for RNA-RNA interactions

Figure 3.3 lists the different data-handling steps to determine RNA-RNA interactions on a transcriptome-wide scale. There are separate tools available for each step - listed in grey boxes, which can be used interchangeably but may require different handling. In principle, the workflow starts with a differential expression analysis (DEA) of selected RNA-seq experiments, which results in a list of differentially expressed transcripts. For that, DESeq2 (Love et al., 2014), edgeR (Robinson et al., 2010), or NOISeq (Tarazona et al., 2015) are commonly used. The identified ncRNAs are subjected to the prediction of homologous

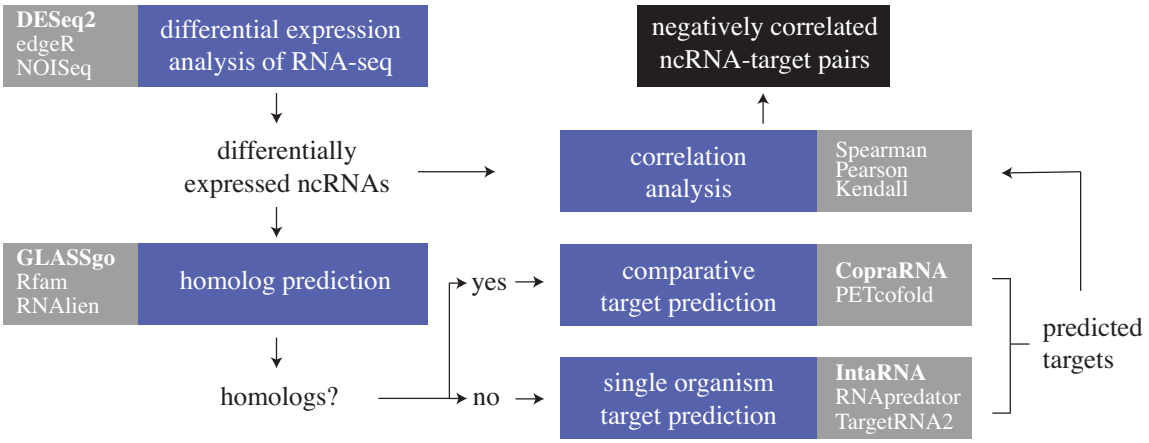

Fig. 3.3 Detection of RNA-RNA interactions by combining both differential RNA-seq data and comparative prediction algorithms. Here, the grey boxes indicate different tools that can be used interchangeably. Tools used in this work are marked in bold.

sequences, which is followed by ncRNA target prediction. As described, GLASSgo detects sRNA homologs from a single input sequence. Alternatively, covariance models (CM) can be used to search for RNA homologs. RNAlien automated the construction of CMs, and Rfam contains pre-built models for a predefined set of RNA families. In the presence of ncRNA homologs, comparative approaches can be applied. Here, CopraRNA (Wright et al., 2013b) was used, but PETcofold (Seemann et al., 2011) is also capable of predicting RNA-RNA interactions based on multiple sequence alignments. If this homologous information is unavailable, other prediction methods based on thermodynamic models can be applied. These include, among others, IntaRNA (Busch et al., 2008), RNApredator (Eggenhofer et al., 2011), and TargetRNA2 (Kery et al., 2014). This results in a list of ncRNAs and their predicted targets. In the following, these ncRNA-target pairs are matched against

the respective expression levels in the DEA. A correlation coefficient is calculated on the expression values between the ncRNA and its targets. Most commonly, Spearman's rank correlation coefficient, the Pearson correlation coefficient, or the Kendall rank correlation coefficient are used. This can be based on the raw counts itself, the fold-change, or any other transformed value. Finally, filtering for negative correlation identifies the putative ncRNA-target interaction pairs.

3.1.3 Validation of detected RNA-RNA interactions

To assess this approach, sequencing data from four transcriptome-wide *E.coli* RNA-seq experiments were used (McClure et al., 2013). In those, the cells were exposed to α-methylglucoside (αMG) and 2-Deoxy-d-glucose (2-DG), accumulating to more than three-quarters of a billion reads. Quality control was done using `FastQC`, and the reads were trimmed off adapter sequences using `trimmomatic`. In doing so, reads that fall short of an average Phred score of 20 and a minimum length of 10nt were discarded. In total, $\sim 6\%$ of the raw reads were dropped in the pre-processing. This is followed by alignment of the reads against the reference genome of *E.coli* K-12 substr. MG1655 (NC_000913.3) using `segemehl` (Hoffmann et al., 2009). For that, the default settings with an accuracy of 90% were applied. In total, $\sim 80\%$ of the raw reads could have been aligned against the reference genome. Subsequently, `featureCounts` (Liao et al., 2014) was used to assign the sequencing reads to genomic features for which the sequence ontology (SO) terms (Eilbeck et al., 2005) 'ncRNA' and 'CDS' were used. In total, 70.5% of the sequencing reads that account for more than half a billion reads were assigned to these features. Among those reads, 7.6% falls into ncRNAs. Subsequently, the differential expression analysis was done using `DESeq2` (Love et al., 2014). In doing so, a local smoothed dispersion fit was applied as the parametric curve did not fit the observed dispersion mean relationship. sRNAs which exhibit a fold-change ≥ 1.5 or ≤ -1.5 in at least two samples were further considered, and thus, subjected to the target prediction. This resulted in 53 distinct ncRNAs in which most of the the sRNAs (40) satisfy the condition in exactly two samples. The remaining sRNAs are significantly up- or downregulated in three (11) or all four samples (2). In the case of mRNAs, 1402 transcripts are either up- or downregulated in two (941), three (339), or all four samples (122). In the following, the sequences of these sRNAs were subjected to `GLASSgo` for the detection of homologous sequences. Here, the default parameters were used and the search was restricted to bacterial clades. Each of the sRNAs was pairwise aligned to its detected homologous sequences using `ClustalW` (Sievers et al., 2011). The sequences with a mean pairwise identity $> 95\%$ and $< 80\%$ were discarded. In addition, the sequences were not further considered when the corresponding organism is not supported by `CopraRNA`.

For each sRNA, up to ten homologous sequences were randomly selected and subjected to the target prediction. This resulted in 28 sRNAs for which at least three homologous sequences could be detected, which were then subjected to `CopraRNA`. For the remaining 25 sRNAs, the target prediction was done using `IntaRNA`. Subsequently, the resulting sRNA-mRNA interactions were filtered for a p-value < 0.01. The surviving interactions were subjected to the correlation analysis in which the spearman's rank correlatio coefficient ρ was calculated on the fold-change values of the differential expression analysis. Here, the sRNA-mRNA pairs with $\rho < -0.75$ were identified as an interaction candidate. This results in 74 interactions (involving 18 sRNAs) and 23 interactions (involving 14 sRNAs) when using `CopraRNA` and `IntaRNA`, respectively. In order to assess the prediction performance, the detected interactions were matched against experimentally validated interactions. Here, the sRNATarBase v3.0 (Wang et al., 2016) was used, which contains 401 interactions involving 46 distinct sRNAs in *E.coli*. In the following, only the sRNAs were considered for which the sRNATarBase contains an interaction. This results in 25 interactions for both `CopraRNA` and `IntaRNA` involving 14 sRNAs. Of these interactions, 18 could have been found in the sRNATarBase, which results in a positive predictive value of 72%.

3.2 Data-driven inference of RNA-RNA interactions

In contrast to the prediction of RNA-RNA interactions, data-driven methods utilise experimental data. In this work, `RNAnue` (Schäfer and Voß, 2021) was developed to provide a comprehensive workflow to predict RNA-RNA interactions from raw sequencing data that originate from but are not restricted to *Direct Duplex Detection* (DDD) experiments. Figure 3.4 depicts the workflow of `RNAnue`. In short, the reads are either pre-processed, which includes clipping, trimming, filtering, and optionally merging or directly subjected to split read detection. In the former, `RNAnue` implements its own routine using a modified Boyer-Moore algorithm to prepare the reads for the subsequent analysis. This is followed by the read alignment with `segemehl`, which combines the detection of the split reads with the calculation of filter scores (e.g., complementarity, hybridisation energy). In addition, the split reads are clustered and subsequently merged with overlapping annotated features in the genome to so-called transcript interactions. These are evaluated statistically, and the p-value and the global filtering scores are reported in the transcript interaction table. In the following, the individual steps are described and evaluated in detail, and the overall prediction accuracy is assessed.

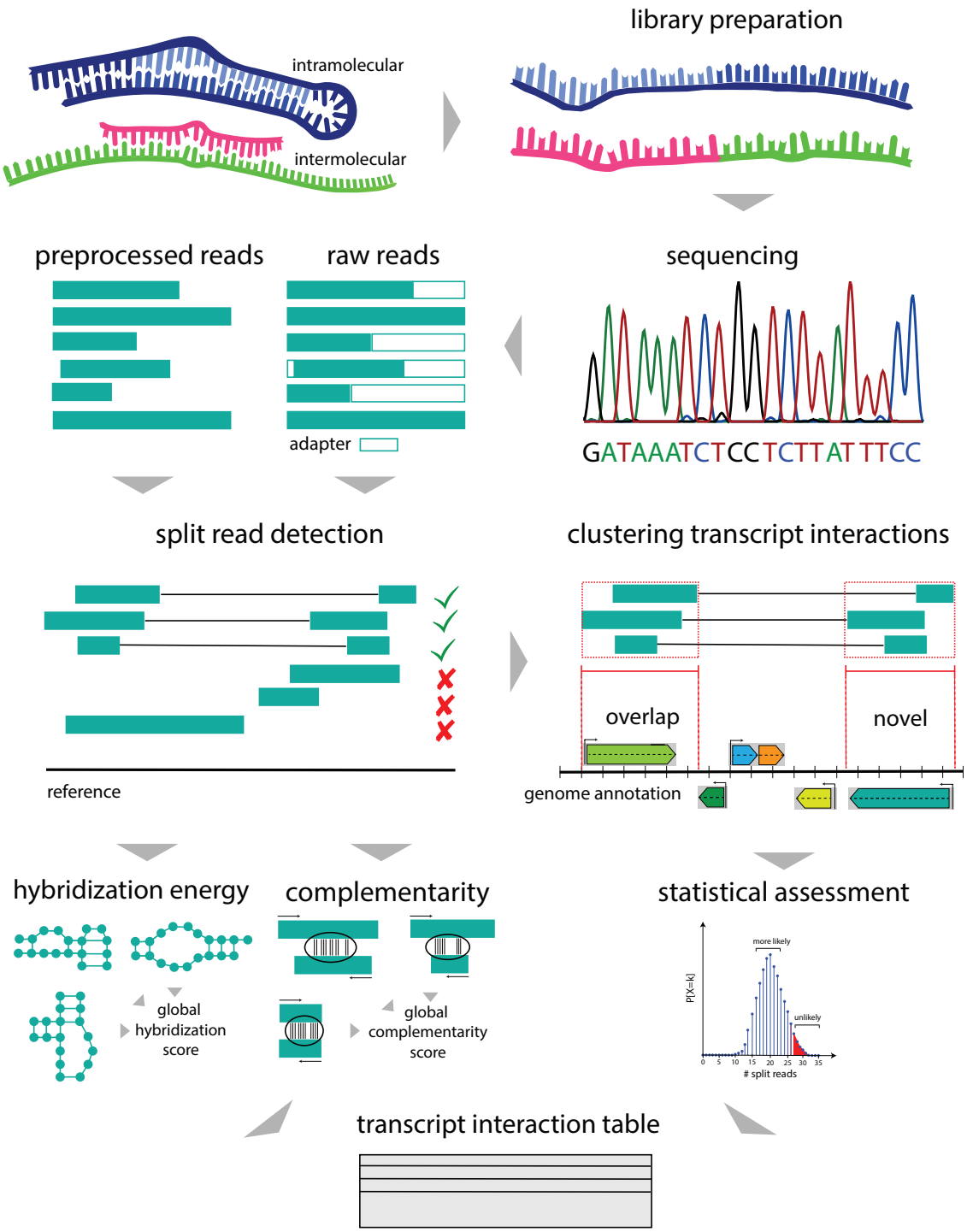

Fig. 3.4 Overview of `RNAnue`. Sequence reads are pre-processed (clipped, trimmed, and merged) or directly subjected to split read detection. This includes mapping and calculation of filtering scores (e.g., complementarity, hybridisation energy) and is followed by the clustering of the identified split reads. Clusters are merged with overlapping annotated features to so-called transcript interactions. These are evaluated statistically, and the p-value and global filtering scores are reported in the transcript interaction table.

3.2.1 Pre-processing of RNA-seq data

In a typical RNA-Seq experiment, the raw reads are pre-processed, which often includes the removal of PCR duplicates. Without Unique Molecular Identifiers (UMIs), computational strategies are typically used to remove PCR duplicates. However, `RNAnue` does not include the removal of PCR duplicates. Consequently, the reads need to be de-duplicated before the analysis or following the alignment procedure. For that, multiple tools are available that either work directly on the raw sequencing reads or first align the reads and subsequently use the mapping coordinates to de-duplicate the reads. These approaches were assessed using 100 million simulated sequencing reads of 150 bases in length. This was based on the genome sequence and associated annotations of *E. coli* MG1655 (NC_000913.3). In principle, within a randomly selected feature, a subsequence of random length (up to 150 bases) was extracted. In the case of $\sim$30% of the generated reads, an additional random subsequence from another feature was concatenated to the read to mimic a split read. In addition, the reads were extended with a sequence of 14 bases that resemble the UMIs, and subsequently, the read was filled up with bases from Illumina adapter sequences. Finally, each read was duplicated by a random number between 1 and 100, with sequencing errors introduced in about 10% of the duplicated reads. This resulted in $\sim$ 2 million reads with different numbers of replicates, making up 100 million reads. Subsequently, `BBtools Dedupe` (Bushnell, 2014) was applied on the raw reads, whereas `SAMtools markdup` (Danecek et al., 2021) was applied to the alignment results subsequent to mapping against the aforementioned reference genome using `BWA` (Vasimuddin et al., 2019). As listed in Table 3.1, `BBtools dedupe` could detect $\sim$ 93% of the introduced duplicates, whereas `SAMtools markdup` removed 85% of the duplicates. Similarly, `BBtools dedupe` retained more than 99.9% of the original reads, as opposed to 67% in the case of `SAMtools markdup`. On the basis of these results, `BBtools dedupe` was generally used in the analysed RNA-Seq datasets prior to the subsequent processing to remove any bias introduced by PCR duplicates.

method	duplicates	unique reads
`BBTools dedupe`	91,496,665 (93%)	1,999,318 (99,9%)
`SAMtools markdup`	83,111,550 (85%)	1,342,138 (67%)

Table 3.1 Removal of PCR duplicates on an artificial data set using `BBtools dedupe` and `samtools markdup`. In the former, these are detected using sequence similarities on the raw reads, whereas the latter uses mapping coordinates of the alignment results to identify duplicated sequences.

Trimming & filtering

In the sequencing procedure, `RNAnue` utilises a modified Boyer-Moore string-search algorithm (Boyer and Moore, 1977) to remove adapter contamination from the sequence reads. It is based on the idea that by matching the pattern from the right rather than from the left, regions containing matches can be quickly identified and skipped, which results in a significant speed-up. However, this is less efficient on small alphabets (e.g., DNA) because substrings re-occur frequently. As a result, skips get shorter. Sustik and Moore (2007) introduced a variant of the algorithm that also works efficiently on small alphabets. The algorithm memorises two previously matched blocks, thereby enabling longer shifts. This requires a

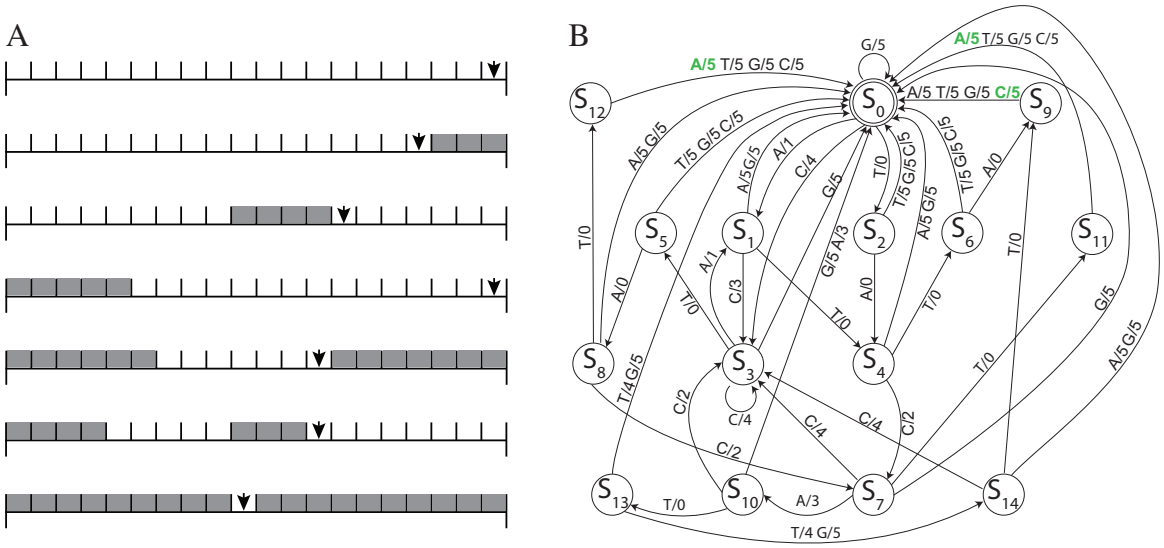

Fig. 3.5 (A) Illustration of the different configurations for the two matching blocks in the algorithm. A black downward arrow indicates the current reading position, and the grey boxes indicate the previously matched positions. (B) The state diagram of the Mealy Machine for the example pattern CATAT. The states are represented by a node with the edges showing the transitions from one state to another. If an input does not change the initial state, this is represented by a circular arrow returning to the original state. Each edge is labelled with 'j / k', where j is the input (alphabet) and k is the output (shift amount).

more sophisticated preprocessing of the search pattern to determine the shift amount when matching with the text. In the following, these blocks are referred to as *left* and *right*. While the left block starts at the beginning of the pattern, the right block is always extended to the right-most position. In the case of an empty right block, the character that aligns with the last character of the pattern is read. Figure 3.5A depicts different configurations of the matching block (grey boxes) and the current reading position (downward arrow). The first row shows the starting state with no matched characters; thereby, both matching blocks are empty. This state also occurs when previously matched characters move out of scope, which happens

when the pattern is moved by its length. In the second and third rows, the right block contains a number of successful matches and will be extended at the right-most position. In principle, the right block will be moved to the right when an extended block that contains a mismatch fits to its left in the text. The configurations in the remaining rows depict a non-empty left block with different reading positions at the right-most position of the right block. In the last row, a state is illustrated in which a successful match at the reading position results in a full pattern match. The algorithm is based on a finite-state machine (FSM). Let m be the length of the read and n the length of the pattern/adapter. Then, a Mealy machine is built that is defined as a 6-tuple $\mathscr{A} = (Q, \sigma, \Omega, \delta, \lambda, q_0)$. In this regard, Q is a finite set of states with $|Q| \leq \Sigma m^3$ that capture the left and right matching block, and $\Sigma = \{A, T, G, C\}$ is the finite input alphabet corresponding to the bases in the pattern. Ω with $|\Omega| \leq n$ represents the finite output alphabet corresponding to the amount the pattern can be shifted. In principle, this is limited by the length of the pattern. In addition, δ and λ correspond to the transition function $\delta : Q \times \Sigma \to Q$ and the output function $\lambda : Q \times \Sigma \to \Omega$, respectively. Finally, q_0 is the starting and final state with $q_0 \in Q$. Figure 3.5B illustrates such a Mealy machine for the pattern CATAT as a state diagram. The states are represented by a node with the edges showing the transitions from one state to another. If an input does not change the initial state, this is represented by a circular arrow returning to the original state. As input, the automaton reads the character at the current reading position, and the output corresponds to the amount the pattern needs to be shifted. Each edge is labelled with 'j / k', where j is the input and k is the output. Each state captures the left and right matching block that is used to determine the current reading position. This means that the search pattern ultimately determines the number of states. Formally, $left$ indicates the number of characters in the left block. Let $align$ be the number of previously matched characters, then $T[align + i] = P[i]$ for $0 < i \leq left$. In this regard, $left = 0$ represents an empty left block corresponding to the initial and final state. The right block is described by its endpoints, denoted as $rightStart$ and $rightEnd$. This means that $T[align + i] = P(i)$ for $rightStart \leq i < rightEnd$. Table 3.2 lists the corresponding state-transition table in which the rows represent the individual states. On the left side of the table, the triplets in the row/column intersection contain the shift amount, the next state index, and the new reading position. The reading position within the pattern is also depicted in the middle part of the table marked with an '*', while an 'X' indicates the already matched positions. Finally, the right side of the table shows the actual values of the left and right blocks, as described before. In the main algorithm 3.2.1, the loop aligns the m characters long pattern against the n characters long text. This is done as long as the pattern length exceeds the remaining text.

state	A	T	G	C	C	A	T	A	T	*left*	*right*
0	(1,1,4)	(0,2,3)	(5,0,4)	(4,3,4)					*	0	(4,4)
1	(5,0,4)	(0,4,2)	(5,0,4)	(4,3,4)				X	*	0	(4,3)
2	(0,4,2)	(5,0,4)	(5,0,4)	(5,0,4)				*	X	0	(3,4)
3	(1,1,4)	(0,5,3)	(5,0,4)	(4,3,4)	X				*	1	(4,4)
4	(5,0,4)	(0,6,1)	(5,0,4)	(2,7,4)			*	X	X	0	(2,4)
5	(0,8,2)	(5,0,4)	(5,0,4)	(5,0,4)	X			*	X	1	(3,4)
6	(0,9,0)	(5,0,4)	(5,0,4)	(5,0,4)		*	X	X	X	0	(1,4)
7	(3,10,2)	(0,11,3)	(5,0,4)	(4,3,4)	X	X	X		*	3	(4,4)
8	(5,0,4)	(0,12,1)	(5,0,4)	(2,7,4)	X		*	X	X	1	(2,4)
9	(5,0,4)	(5,0,4)	(5,0,4)	**(5,0,4)**	*	X	X	X	X	0	(1,4)
10	(3,0,4)	(0,13,3)	(5,0,4)	(2,3,4)		X	*			0	(2,1)
11	**(5,0,4)**	(5,0,4)	(5,0,4)	(5,0,4)	X	X	X	*	X	3	(3,4)
12	**(5,0,4)**	(5,0,4)	(5,0,4)	(5,0,4)	X	*	X	X	X	1	(1,4)
13	(0,14,4)	(4,0,4)	(5,0,4)	(3,3,4)		X	X	*		0	(3,1)
14	(5,0,4)	(0,9,0)	(5,0,4)	(4,3,4)		X	X	X	*	0	(4,1)

Table 3.2 On the left side of the table, the triplets in the row/column intersection contain the shift amount, the next state index, and the new reading position. Triples in bold indicate a match. The middle of the table depicts the reading position within the pattern indicated with an '*', while 'X' represents the already matched positions. On the right, the actual values of the left and right blocks for the current position are shown.

Algorithm 3.2.1: SEQUENCESEARCH($read, adapter$)

$S \leftarrow calcStateTransitionTable(adapter)$
$m = length\ of\ read$
$n = length\ of\ adapter$
$align \leftarrow 0, state \leftarrow 1, readPos \leftarrow m-1$
while $align < |read| - |adapter|$
do
- $c \leftarrow reads[align + readPos]$
- $(shift, state, readPos, match) \leftarrow S[state, c]$
- **if** $match = true$
  - **then** *return align*
- $align \leftarrow align + shift$

if $read[align \cdots m] = adapter[0...(m-align)]$
then *return align*
else *return NIL*

Based on the rightmost unmatched character and the current state, the state-transition table S determines the next state. This leads either to subsequent states (according to the index) or eventually back to the initial state. The initial state occurs when the Mealey machine terminates, either having found a perfect match or not. In the case of the former, the algorithm terminates and returns the starting position of the alignment. In the latter, the starting position is extended by the shift amount determined through the state-transition table. Suppose a match could not be detected after all feasible pattern alignments were tested. In that case, a final search determines if the remaining text matches a substring of the pattern or returns NIL otherwise. Subsequently, `RNAnue` trims the reads at the corresponding positions *align*. It either trims the reads preceding from the 5'-end, the 3'-end or both (`--modetrm`). In addition, `RNAnue` allows mismatches, controlled by the parameter $\nu = \frac{t}{m}$ for t mismatches and a read length of m (`--mmrate`). `RNAnue` implements this by setting up wildcards, where the Mealey machine considers each allowed mismatch as a match. In the filtering procedure, `RNAnue` implements a two-step process. At first, an optional window trimming step (`--wtrim`) takes the trimmed reads and scans user-defined windows (`--wsize`) for their average Phred quality score. This starts from the 3'-end and moves window by window towards the 5'-end of the read. If this falls short of the user-defined value (`--quality`), the read is trimmed off the inspected window and continues with the next. This is done until the average Phred score exceeds or equals the user-defined value. If the read length then falls short of the user-defined minimal length (`--minlen`), the read is discarded. The actual filtering procedure then discards reads that fall short of the required average Phred quality score and minimal read length.

`RNAnue`'s pre-processing procedure was assessed using raw reads from a wild-type `SPLASH` sequencing run in *S. cerevisiae* (Aw et al., 2016). The forward sequencing reads with a length of 150bp from four replicates were concatenated into a single file and subjected to `FastQC`. The reports revealed a notable quality drop after 120bp and around the first 10bp in the reverse strand (see Figure A.1) and substantial adapter content starting around 100bp. The forward sequencing reads were then subjected to the pre-processing procedures implemented in `RNAnue`, `cutadapt` (Martin, 2011), `flexbar` (Dodt et al., 2012), and `trimmomatic` (Bolger et al., 2014). This was done in two different runs, applying strict and more tolerant settings. In the former, a perfect match is required to detect the adapter sequence, while the latter allows up to two mismatches. For the adapter trimming itself, the TruSeq index and small RNA adapters were removed (listed in A.1). In both modes, the reads were either filtered or window-trimmed for a Phred quality score of 20. In addition, reads with a length below 15nt were discarded. Subsequently, the surviving reads were subjected to the primary data analysis with `RNAnue`, and the mapping statistics were inspected. As shown in Table 3.3,

the number of reads removed with the considered tools is comparable. When looking at the alignment statistics, between 59.76% and 60.2% of the reads could be aligned using strict settings except for `trimmomatic` in which only 55.90% of the reads could be aligned. In terms of split read detection, applying `RNAnue`'s pre-processing procedures results in 3.903% split reads, which is a fraction more than the other tools (0.68% to 3.895%). Interestingly, the computation time of `RNAnue` and `flexbar` is four and three times slower than `trimmomatic` and `cutadapt` but yields better results. In the tolerant settings, the difference in the split read detection when pre-processing with `cutadapt`, `flexbar`, and `RNAnue` are marginal (4.78% to 4.89%), whereas `trimmomatic` falls behind (4.17%). Is it to be noted that `flexbar` detects most split reads. The runtime is comparable to the strict settings.

tool	reads	alignment	split reads	runtime (s)
unfiltered	13,426,800	38,500	22,872	N/A
strict settings				
`cutadapt`	13,424,732	8,089,403 (60.2%)	522,985 (3.895%)	4m 27s
`flexbar`	13,411,182	8,072,367 (60.1%)	520,235 (3.87%)	12m 01s
`RNAnue`	13,276,152	8,023,818 (59.76%)	**524,041** (3.903%)	18m 43s
`trimmomatic`	13,399,770	7,490,068 (55.90%)	491,165 (3.67%)	4m 45s
tolerant settings				
`cutadapt`	13,424,464	10,041,537 (74.79%)	643,366 (4.79%)	6m 44s
`flexbar`	13,408,690	10,209,714 (76.04%)	**655,998** (4.89%)	13m 32s
`RNAnue`	13,165,126	9,615,981 (71.62%)	645,781 (4.81%)	21m 02s
`trimmomatic`	13,397,045	10,719,884 (79.84%)	559,267 (4.17%)	5m 34s

Table 3.3 `RNAnue`'s split read detection of the `SPLASH` yeast dataset on the raw and trimmed data by each considered tool. Numbers in bold type indicate the maximum number of split reads.

Merging paired-end reads

In the case of paired-end reads, both pairs are merged before further processing. This is done by calculating the overlaps between the forward and the reverse read and joining both reads accordingly. At first, the reverse complement of the reverse read is calculated to have both forward and reverse read in the same orientation. Subsequently, the longest common substring (LCS) between the forward and the reverse read determines its overlaps, allowing the read pair to be joined together. Let $v, w \in \Sigma^n$ with $\Sigma = \{A, T, G, C\}$ represent a paired-end read of size n that results in an overlap $o_{ij} \in \Sigma^l$ where i and j correspond to the starting position within v and w, respectively. Then the merged read $m \in \Sigma^{i+l+(n-j)}$ is defined as

$m = v_1 \cdots v_{i+l} w_{j+l} \cdots w_n$ where $l \geq \Phi$. Φ is a user-defined cut-off (parameter `--minovlps`) that determines the minimum length of the overlap for reads to be merged. In the case of multiple overlaps of the same size, the ones with the highest position i and lowest position j are selected. The LCS is calculated using a generalised suffix tree. A suffix tree is a specific type of search tree that contains all the suffixes for a given string S of size n. Suffix trees contain exactly n leaves that are numbered from 1 to n, with each node having at least two children, except for the root. Moreover, each edge is labelled with a non-empty substring of S with different starting characters when originating from the same node. In that regard, a string obtained by concatenating all the edge labels from the root to leaf i is denoted as $S[i...n]$, for i from 1 to n.

Generalised suffix tree

Ukkonen (1995) devised an online algorithm to construct a suffix tree from a string in time linear of the length of the string. Gusfield (1997) depicts the algorithm and compares it with other approaches, such as Weiner's method (Weiner, 1973) of similar runtime. However, the space requirements differ significantly. For a given string S of length m, the algorithm constructs a sequence of implicit suffix trees I_i for each prefix $S[1...i]$, starting from I_1 and incrementing i until I_m is built. It is followed by the construction of the actual suffix tree from I_m. The algorithm begins with constructing the tree I_1, which consists of a single edge labelled with the character $S[1]$. Subsequently, $m-1$ phases are executed. In phase $i+1$, the tree I_{i+1} is constructed from I_i. Each phase $i+1$ is further divided into $i+1$ extensions, one for each of the $i+1$ suffixes of $S[1...i+1]$. In the j-th extension of phase $i+1$, the algorithm finds the path corresponding to substring $\beta = S[j...i]$. The detected path is then extended by adding the character $S[i+1]$, unless the substring $\beta + S[i+1]$ is already in the tree. As a consequence, each possible substring combination is considered, as shown in algorithm 3.2.2. However, different rules apply when extending the tree by adding $S[i+1]$ to the located β. Assume the path β ends at some leaf edge, then the character $S[i+1]$ is added to the end of that labelled leaf edge. In contrast, let β not end on a leaf edge and instead, at least one labelled path continues from the end of β. Then, a new leaf edge that starts from the end of β will be created and labelled with character $S[i+1]$. A similar situation occurs when β ends within an edge that leads to a new leaf edge starting with character $S[i+1]$. Naturally, nothing needs to be done if $\beta + S[i+1]$ can be found within the tree. One key element in the algorithm is to locate the path $\beta = S[j...i]$ in each extension j. For that, suffix links speed up the traversal starting from the root in the tree. Formally, let $x\alpha$ denote an arbitrary string in which x denotes a single character, and α denotes a substring that can also be empty. Let v be an internal node with path label $x\alpha$ in tree I. If there is another node $s(v)$ with the label α

in I, then a pointer from v to $s(v)$ is called a suffix link. In the last step, the implicit suffix tree I_m is converted to a suffix tree in $\mathcal{O}(m)$ time. It starts with adding the terminal symbol \$ to string S that is then subjected to the algorithm described above. This leads to a tree where each suffix ends at a leaf, thereby explicitly represented. In other words, no suffix is a prefix of any other suffix. In order to determine the LCS, a generalised suffix tree needs to be built, that is, a suffix tree on the basis of a set of strings $D = S_1, S_2, ..., S_d$ of total length n. In the scope of paired-end read merging, a generalised suffix tree with $d = 2$ is considered. For that, the algorithm mentioned above is applied to a concatenated string of both sequences. For the $D_1 = AGTCGTGAT$ and $D_2 = GATCAT$, Ukkonen's algorithm is then applied with the string $D_1\#D_2\$ = AGTCGTGAT\#GATCAT\$$. Here, the separator symbol # is used. The resulting suffix tree needs to be refined by removing unwanted substrings on the path label. This includes path labels that stem from both input strings. In such an instance, only the part of the initial string is preserved, and the string is trimmed after the occurrence of #. This results in a suffix tree in which each node has children with substrings of either D_1, D_2 or both. Finding the deepest internal nodes with leaf nodes belonging to both strings reveals the LCS.

Algorithm 3.2.2: CONSTRUCTSUFFIXTREE(S)

$T_1 \leftarrow createTree(suffixExtension(T_1, \{\}, S[1]))$
for $i \leftarrow 1$ **to** $m-1$
{ **for** $j \leftarrow 1$ **to** $i+1$
{ $\beta \leftarrow S[j...i]$
$T_{i+1} \leftarrow suffixExtension(T_i, beta, S[1])$

`RNAnue` implements this algorithm to determine the LCS, and subsequently concatenates the reads if the LCS exceeds a user-defined cutoff (`--minovlps`). This again was assessed using the sequencing reads from the SPLASH dataset in *S. cerevisiae*. For that, the reads were preprocessed using `RNAnue` with tolerant settings as before (see Table 3.3), resulting in 13,165,126 paired-end reads. These were merged using `RNAnue`, `SeqPrep` (https://github.com/jstjohn/SeqPrep), and `Pear` (Zhang et al., 2014) and subsequently subjected to the split read detection with `RNAnue`. In addition, the unmerged reads were aligned in paired-end mode (unmerged). As listed in Figure 3.4, using `RNAnue`'s read merging capabilities detects more split reads than `SeqPrep` but less than `Pear`. In total, however, `RNAnue` detects the most split reads. Interestingly, `Pear` detects more split reads using the merged transcripts but falls short when the split detection is applied to the unpaired reads.

tool	reads	aligned	split reads
total	13'165'126		
merged reads			
RNAnue	7,726'501	6,971,123	426,761
SeqPrep	7,701'651	6,761,716	397,811
Pear	8,686,152	7,412,560	**449,981**
unmerged reads			
RNAnue	5,438,625	3,126,841	296,671
SeqPrep	5,463,475	3,067,172	**307,722**
Pear	4,478,974	2,517,162	166,815

Table 3.4 Results of the split read detection using RNAnue. SPLASH sequencing reads in yeast were merged using the different tools (top). In addition, the unmerged reads were subjected to the split read detection (bottom).

3.2.2 Primary data analysis

In the primary data analysis, the alignment procedure is critical to the subsequent analysis. Multiple alignment tools are available. To assess the performance of these aligners, pairs of reads of a regular RNA-seq dataset (Seo et al., 2014) were randomly concatenated to mimic split reads. In addition, both segments were trimmed off their random 5'- and 3'-ends by a random number drawn from the normal distribution with $\mu = 1$ and $\sigma = 0$ to represent the length variability of real-world datasets. In total, 100 million reads of length 30 to 202 bases were generated. These reads were aligned against the reference genome of *E.coli* K12 substr. MG1655 (NC_000913.3) using BWA-MEM, STAR (Dobin et al., 2013) and segemehl. As illustrated in Figure 3.6A and Table 3.5, segemehl retrieves significantly more chimeric reads ($\sim 96\%$) than the other aligners. In addition, the unmodified RNA-seq was aligned with the identical parameters to test for false-positive splits. The results are summarised in Figure 3.6B and Table 3.6. All tools reported less than 1% false splits. It is to be noted that RNAnue, due to its complementarity, thermodynamic and statistical filtering, is able to further reduce the false-positive rate (FPR) from 0.8% (segemehl alone) to less than 0.1%. Based on this results, RNAnue utilises segemehl to perform the alignment procedure. This can be invoked using the RNAnue 'align' positional argument. Consequently, the parameters that control the alignment result are forwarded to Segemehl. These involve the seeds of the semi-global alignment that satisfy user-controlled values such as the number of allowed differences within the seeds (--differences, default: 1), the maximum number of hits for a query seed (--maxinterval, default: 100) or the minimum E-value (--evalue, default: 5). However, in the aforementioned assessment of segemehl, changes in these

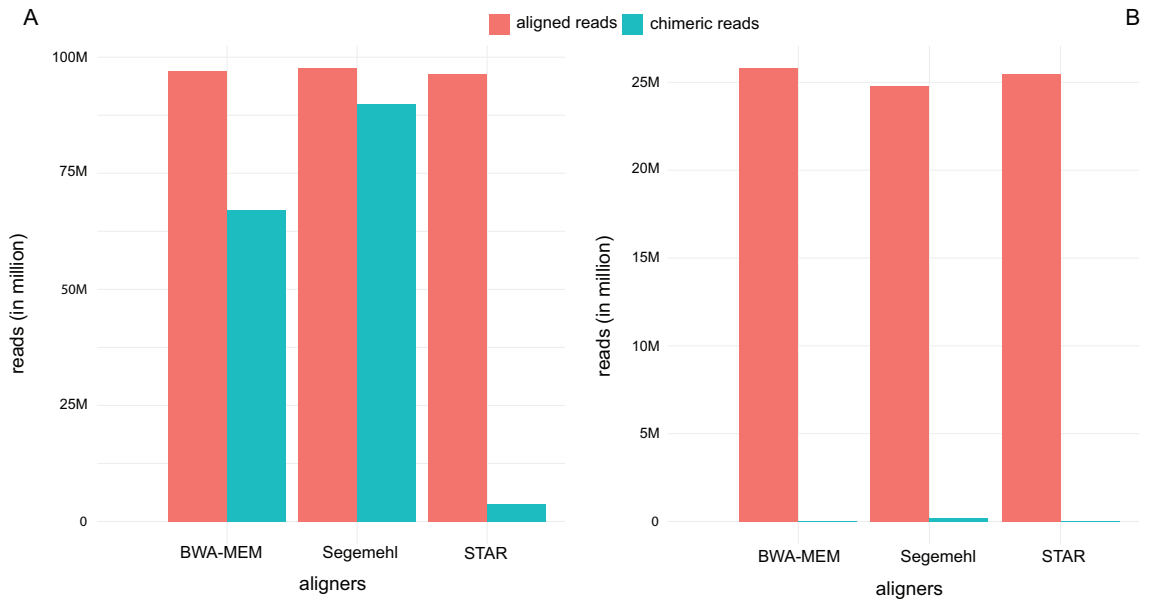

Fig. 3.6 Benchmark of different alignment tools for the split read detection. (A) Number of split reads detected using a dataset mimicking split reads (B) Number of split reads detected in a regular RNA-seq dataset

aligner	aligned reads	chimeric reads
`BWA-MEM`	96,973,939	67,045,679
`Segemehl`	97,631,828	96,357,854
`STAR`	89,841,641	3,701,360

Table 3.5 Mapping statistics for the set of artificial chimeric reads.

three parameters only contributed to an increased runtime and did not significantly improve the performance in terms of detected chimeric reads or number of false positives. As a consequence, `RNAnue` neglects these parameters and utilises its default values. In contrast, the `--accuracy` parameter affects the alignment results as it specifies the minimum percentage of matches within the reads to be considered as alignment. In doing so, other parameters guide

aligner	aligned reads	chimeric reads (FPR)
`BWA-MEM`	25,806,551	52,613 ($\sim$ 0.2%)
`Segemehl`	24,810,363	204,188 ($\sim$ 0.8%)
`STAR`	25,451,115	12,137 ($\sim$ 0.04%)

Table 3.6 Numbers of chimeric reads (false positives) for a regular RNA-seq dataset using different aligners. Please note that, depending on the library preparation protocol, chimeric reads can be present in the sequencing data. These would than actually be true positives.

the reported split reads. These include the minimal length of the fragments (`--minfraglen`) that make the split read. This is closely related to the minimum score of the fragment (`--minfragsco`). Moreover, a split read is accepted as such if the alignment covers at least a specific value (`--minsplicecov`).

3.2.3 Split read calling

In the split reads detection procedure (positional argument `detect`), `RNAnue` takes the alignment results from `segemehl` that also include initially detected split reads and determines the reads used in the subsequent analysis. This results in output files for single, split, and multi-split reads. At first, `RNAnue` iterates through the alignments, discards the unmapped reads, and sorts them into blocks of n reads with identical `QNAME`. These blocks then represent either single or split reads. Subsequently, the aforementioned `SAM` tags are scanned to reveal more information about the read. For that, the tag `XJ` is examined first. The absence of this tag is indicative of a read that aligned consecutively against the reference genome. Similarly, even though `XJ=1` corresponds to a split read, it only consists of one fragment and is thereby identified as a single read. In the event of `n=1`, meaning that the read could be mapped unambiguously against the reference genome, it will be written to the output file for single reads. In the case of `XJ>1`, `segemehl` reports the split reads in either a single entry or spanning multiple entries. In the former, consecutive fragments that make up the split reads are reported in a single entry when located on the same strand. In contrast, fragments with alternating orientations are reported in multiple entries. Subsequently, the `CIGAR` strings of the split reads are examined. Upon detecting skipped regions from the reference (`CIGAR` operator `N`), a split read can be deduced. The segments are then stored in separate entries for which the left-most mapping position (`POS`), the aligned sequence (`SEQ`), the base quality (`QUAL`) and the `CIGAR` string itself have to be modified accordingly. This results in m read groups that, in turn, consist of k non-overlapping fragments. The interval B+ tree (see 3.2.5) is used to match the fragments with exon information. This results in removing fragments that match outside exon annotated regions. Also, consecutive fragments are considered as one if their genomic distance corresponds to an intron and their matching position is either at the start or end position of the annotated feature. In the case of `k>2`, all 2-permutations of k fragments need to be considered to determine the most likely split read. Complementarity and hybridisation energy are considered. In principle, the pair with the highest complementarity is considered further. In the unlikely case of an identical complementarity, the hybridisation energy is considered as well. Similarly, for optimal pairs in multiple read groups (`m>1`), these measures are compared as well. This can lead to an unambiguous split read that is stored in the corresponding output file. However, if this approach can make no assumption about

the correct split reads, It is stored in the output for multi-splits. In the case of a high ratio of multi-split reads, the soft clips in the alignment can be omitted. By doing so, fewer overhangs that are not part of the alignment are subjected to the calculation of the hybridisation energy (`--exclclipping`).

3.2.4 Filtering of split reads

The aforedescribed primary data analysis ideally identifies a plethora of chimeric reads that need to be assessed for further analysis. For that, `RNAnue` employs global filtering measures to assess potential interactions. These include a statistical assessment of the interaction level and scoring of the chimeric reads using complementarity and hybridisation energy.

Statistical assessment

In order to assess the significance of detected interaction features, `RNAnue` adopts the strategy of Sharma et al. (2016) to estimate the likelihood of ligation by chance. This uses the multinomial distribution (`k=2`) to model the discrete probability distribution for the ligation by chance of a *transcript interaction* between two transcripts t_x and t_y. The probability of success (ligation by chance) is proportional to the relative abundances of each of the transcripts. The joint probability density function for a random ligation event between the transcripts t_x and t_y with r_x and r_y reads, respectively, is defined as

$$P(t_x{:}t_y) = \begin{cases} 2P(t_x)P(t_y), & \text{if } t_x{:}t_y \text{ is observed and } t_x \neq t_y \\ P(t_x)P(t_y), & \text{if } t_x{:}t_y \text{ is observed and } t_x = t_y \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

where

$$P(t_x) = \frac{r_x}{\sum_{\forall t_i} r_i} \tag{3.2}$$

For pairs $t_x : t_y$ that have not been observed, the probability is set explicitly to zero because it cannot be faithfully decided if they are missing because they are impossible or have simply not been observed, e.g., due to insufficient sequencing depth. As a result, P(t_x:t_y) has to be re-normalised to sum up to 1. The number of split reads X for an interaction t_x:t_y is modelled as

$$X \sim B(n, p = P(t_x{:}t_y)). \tag{3.3}$$

For each interaction, a binomial test is applied to generate a p-value and using the Benjamini-Hochberg adjustment corrects for multiple testing with a standard α value of 0.1.

Complementarity

The complementarity of two putative interaction sites is computed as the fraction of matches in a modified local alignment procedure, where A aligns with U, and G aligns with C, and U. Matches are scored with 1, mismatches with -1 and gap open and extension with -3 and -2, respectively. This scoring scheme is inspired by States et al. (1991), where these scores proved to be optimal for sequences with 75% sequence conservation, which is in the range that is to be expected for the complementarity of interactions. Furthermore, this favours contiguous over fragmented alignments, a typical feature of the seed region of interactions (Fabian et al., 2010; Kai et al., 2010). Here, the Waterman-Eggert algorithm is used to compute the alignments between the segments of all k split reads while considering the opposing segment in reverse order. As this also reports suboptimal alignments the one is selected that exhibits the highest ratio between the number of matches and the length of the alignment that satisfies the alignment-to-read ratio. Assuming that the alignment of all k split reads results in j optimal/suboptimal alignments, then the sets $\mathscr{M}_i = \{m_{i1}, ..., m_{ij}\}$ and $\mathscr{L}_i = \{l_{i1}, ..., l_{ij}\}$ for split read i correspond to the number of matches in the respective alignment and the alignment length, respectively. We define the complementarity c_i for split read i as follows:

$$c_i = \max_{1 \leq p \leq j} \frac{m_{ip}}{l_{ip}}, \; where \; \frac{m_{ip}}{l_{ip}} = max(\frac{m_{i1}}{l_{i1}}, ..., \frac{m_{ij}}{l_{ij}}) \; with \frac{l_{ip}}{2 \cdot r_i} \geq \theta \tag{3.4}$$

r_i corresponds to the length of read i. θ is a user-defined cutoff (parameter `--sitelenratio`) for the aligned portion of a read. On the level of transcript interactions, the global complementarity score *gcs* summarises the complementarity information of several split reads. Let $\mathscr{T}$ be a transcript interaction that contains k split reads with complementarity scores $\mathscr{C} = \{c_1, ..., c_k\}$, we define the *gcs* as follows:

$$gcs(\mathscr{T}) = \sqrt{\tilde{\mathscr{C}} \cdot \max(\mathscr{C})} \tag{3.5}$$

where $\tilde{\mathscr{C}}$ denotes the median of $\mathscr{C}$. In addition to the *gcs*, the fraction of reads that pass θ and the ratio of unaligned to total read length cut-offs is reported.

Hybridisation energy and probability

The interaction of two RNAs is driven by the thermodynamics of the hybridisation reaction, resulting in the loss of free energy. RNAlib v2.4.14 (Lorenz et al., 2011) was used to

estimate the minimum free energy hybrid structure and its probability in the ensemble of all possible interactions. To be precise, $\Delta\Delta G = \Delta G_p + \Delta G_u$ is computed, where ΔG_p is the free energy loss of the hybridisation and ΔG_u is the free energy gain needed to unpair the interacting sites. Similar to the complementarity, a summarised score for transcript interactions is provided, termed as global hybridisation score $ghs(\mathscr{T}) = \sqrt{\tilde{\mathscr{G}} \cdot \min(\mathscr{G})}$, where $\mathscr{G} = \Delta\Delta G_0, \ldots, \Delta\Delta G_k$ and k is the number of split reads that support the interaction. It is noted that $\Delta\Delta G_i \leq 0$, $\forall G_i \in \mathscr{G}$, otherwise `RNAnue` discards the split read. Similarly, the probability of the hybridisation is computed as the product of the probabilities of the two interactions to be unpaired times the probability of the hybridisation. Accordingly, for probabilities $\mathscr{P} = \{c_1, ..., c_k\}$, global probability score is defined as $gps(\mathscr{T}) = \sqrt{\tilde{\mathscr{P}} \cdot \max(\mathscr{P})}$.

3.2.5 Clustering & annotation

An individual interaction is expected to be supported by several split reads, and a group of such split reads is referred to as *interaction*. Such an *interaction* is described by a pair of non-overlapping genomic segments. To derive *interactions*, the detected split reads are clustered if both their pairs of locations on the genome overlap. One or both segments of the *interactions* may overlap with annotated genomic features, e.g. exons and ncRNAs. In this case, the interactions are further grouped into so-called *transcript interactions*. In more detail, the split reads are clustered into *interactions* as follows: Let split reads and clusters be given by pairs of mapping coordinates $(a,b) : (c,d)$. Two split reads, a split read and a cluster, or two clusters $(a_1,b_1) : (c_1,d_1)$ and $(a_2,b_2) : (c_2,d_2)$ are merged if, both, $d_{ab} = \max(a_1 - b_2, a_2 - b_1)$ and $d_{cd} = \max(c_1 - d_2, c_2 - d_1)$ do not exceed a threshold δ, i.e. $\max(d_{ab}, d_{cd}) \leq \delta$. By default, δ equals 0, such that a minimum overlap of one base in both segments is required for merging. Setting δ to values greater than 0, which can be done via the `--clustdist` parameter of `RNAnue`, also merges clusters/split reads in close proximity ($\leq \delta$). This results in a cluster than is defined by the following genomic coordinates $(\min(a_1,a_2), \max(b_1,b_2)) : (\min(c_1,c_2), \max(d_1,d_2))$. Figure 3.7 illustrates possible combinations of clustered split reads. The coloured bands indicate the clustered segments within each split read; their width represents its boundaries. (A),(B) and (C) describe different clusters that partly overlap with one segment. (D) corresponds to only a single split read, which is its cluster, whereas (E) is simply an isolated cluster. The starting point of the clustering procedure is a pre-sorted list of interval pairs that correspond to the split reads. To be more precise, the intervals are sorted by the starting positions of the first and second segment in ascending order. Moreover, the pairs in themselves are sorted such that the first interval is always located to left of the second interval on the genome. Formally, let

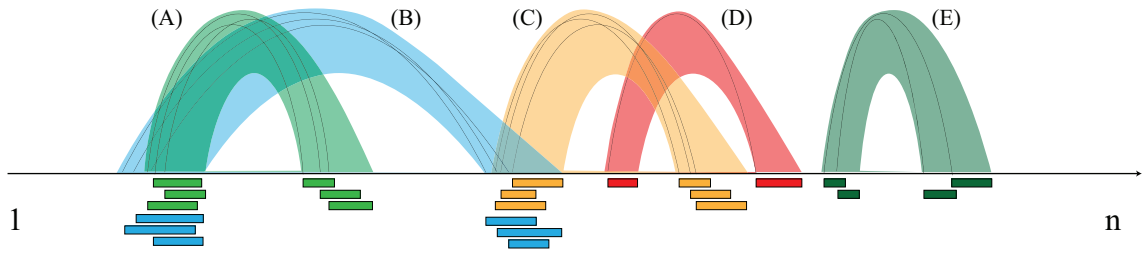

Fig. 3.7 Clustering of the split reads according to the start position of both segments. Black arcs connect the start positions of the segments within a split read. Overlaps between the segments of individual split reads determine the affiliation to a cluster. Coloured bands indicate the clusters that span from the start to the end of each segment. Clusters (A),(B) and (B),(C) differ in the second and first segment, respectively. (D) consists of a single split read (singleton), and (E) occurs isolated from other split reads.

$S = \{s_1, ..., s_m\}$ be the list of sorted split reads. If $i, j, 1 \leq i < j \leq m$ are any two given entries in S with $s_i = (a_i, b_i) : (c_i, d_i)$ and $s_j = (a_j, b_j) : (c_j, d_j)$ then $a_i \leq a_j$ with $a_i < c_i$, $b_i < c_i$ and $a_j < c_j$, $b_j < c_j$. In the case of $a_i = a_j$ then $c_i \leq c_j$. The algorithm 3.2.3 then iterates over the list, merging the pairs of intervals with its neighbour if there is an overlap as defined above.

Algorithm 3.2.3: SPLITREADCLUSTERING(S)

$m \leftarrow number\ of\ elements\ in\ S$
$cluster \leftarrow S[0], interactions \leftarrow []$
for $i \leftarrow 1$ **to** $m-1$
  if $overlap(cluster, S[i]) = true$
    then $merge(cluster, S[i])$
    else $interactions.append(cl)$
      $cluster \leftarrow S[i]$
  $interactions.append(cluster)$
return $(interactions)$

The resulting *interactions* are compared with the existing genome annotation based on the locations of their segments. If an *interaction* segment overlaps with an annotated feature, it is assigned to the respective feature. An *interaction* segment that does not overlap with any annotated feature is treated as a putative new feature and assigned a unique ID. As a result, *transcript interactions* may consist of two annotated transcripts, one annotated and one new transcript, or two new transcripts. Efficient matching to the annotation is done with the help of a modified interval B+ tree that is pre-filled with all annotations.

Interval B+ tree

Bozkaya and Ozsoyoglu (2006) introduced the concept of the interval B+ tree that combines the principles of an interval tree and a B+ tree. That is a hierarchical tree structure with a root value and subtrees with a parent node. The tree is filled with both the cluster information and feature annotation that is, in turn, used to assign features to the split reads. It consists of internal nodes whose children are other nodes and leaf nodes that store the feature intervals and have no children. In the latter, intervals are accompanied by information from their respective entry in the provided annotation. These are currently supported in the General Feature Format (GFF) that can be specified using `--features`. This results in multiple trees corresponding to the number of different `seqids` in the annotations file. In the tree, internal nodes contain three lists. For a node with children 1 to k, the list $C = c_1, ..., c_k$ represents pointers to the respective children. Moreover, $A = a_1, ..., a_k$ and $M = m_1, ..., m_k$ correspond to the smallest lower bounds and the maximum endpoints of downstream intervals, respectively. In the following, the order $k = 7$ has been used. In contrast, the leaf nodes only contain a pointer to the right sibling. `RNAnue` starts with an empty interval B+ tree and subsequently fills the tree using the provided annotation. In practical terms, the top-level intervals are stored with pointers to the entries in the subjacent level. Optionally, this multi-tree structure is then extended with the cluster information that leads to an extension of existing intervals or utterly new ones. When the clusters overhang the existing elements to the left, right, or both, it is extended in the tree accordingly. If no element is present, the corresponding interval is added to the tree. However, adding nodes/intervals can lead to readjustments of the overall structure of the tree. In such an instance, the insertion leads to a node that exceeds the number of allowed children for a node. Consequently, the node needs to be divided to preserve the balance of the tree, resulting in two new nodes. In particular, when an interval is inserted into the tree, it ends up next to an interval that starts in close proximity. The hierarchical nature of the feature annotation can lead to overlapping intervals. This is increased when the cluster information is added to the tree. As a consequence, clusters that expand the given annotations at the start or end are sorted in the tree to the left or right, respectively. The `analysis` step of `RNAnue` uses the interval tree to match the split reads to annotated transcripts. This is listed in algorithm 3.2.4 that returns a list of all overlapping intervals for a given interval $[I_{start}, I_{end}]$, beginning from node N with order k. The resulting list of intervals contains all overlapping intervals and their corresponding elements on different hierarchical levels. Subsequently, the overlaps that are usually on the `gene` level are then further localised. Information from fields such as `ID`, `Name`, `strand`, and `attributes` is used to classify the matched split reads. This is complemented with information from the filtering (see 3.2.4) to provide a comprehensive list of all interactions to be examined.

Algorithm 3.2.4: ALLINTERVALSEARCH($[I_{start}, I_{end}], N, k$)

$overlaps \leftarrow []$
if not $hasChildren(N)$
then return $(findOverlaps([I_{start}, I_{end}], N))$
else
$i \leftarrow 1$
while $i < k$
do
if $intersects([I_{start}, I_{end}], [a_i, m_i])$
then $overlaps.push(IntervalSearch([I_{start}, I_{end}], c_i, k))$
else *break*
$i \leftarrow i + 1$

3.2.6 Validation of detected interactions

RNAnue was applied on human datasets from LIGR-Seq (Sharma et al., 2016), SPLASH (Aw et al., 2016), and PARIS (Lu et al., 2016) (see Tables A.3, A.4, and A.5). For that, the interactions were required to have a p-value of less than 0.05, a *gcs* of at least 0.75 and a *ghs* of less than 0. In addition, an interaction has to be supported by at least two chimeric reads. On the level of predicted interactions, the results of RNAnue to the original analysis pipelines on the respective datasets are summarised in Figure 3.8. Except for PARIS, RNAnue recalls 88–97% of the originally predicted interactions. PARIS needs to be considered separately because the analysis pipeline performs neither a statistical assessment nor a rigid filtering. Figure 3.8 also illustrates that RNAnue is able to capture novel interactions (LIGR-Seq: $\sim$ 29%, SPLASH:$\sim$ 43%, PARIS: $\sim$ 12%). Among these, 2.5–7.6% involve transcripts that do not overlap any annotation (numbers in brackets) and could, therefore, only be detected due to the RNAnue annotation independent clustering procedure. Moreover, RNAnue was benchmarked in comparison to the original data analysis pipeline based on experimentally validated targets from miRTarBase v7.0 and snoDB v1.2.1. In that regard, miRTarBase classifies interactions into strong and less strong, depending on their experimental support. To identify potential differences based on this classification, the benchmark was done for both classes, and the results are shown in Figure 3.9. Interestingly, for the class with weak support RNAnue achieves a lower PPV compared to the original analysis pipeline of LIGR-SEQ, but higher values for the other two. For those with strong support, RNAnue outperforms the other methods. It does not only achieve higher PPVs but also larger absolute numbers of true positives. Taking both classes together, RNAnue achieves a PPV of 0.74, compared to 0.70, 0.44, and 0.67 for LIGR-Seq, PARIS and SPLASH, respectively. For snoRNA-rRNA

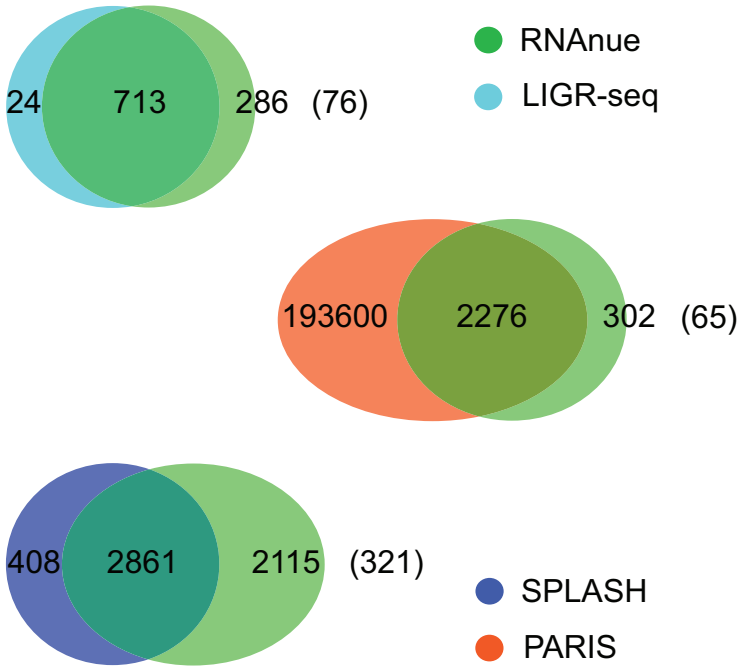

Fig. 3.8 Detected interactions of the corresponding datasets in human samples using `RNAnue` in comparison to the original analyses. Numbers in brackets indicate interactions without annotated features.

interactions, `RNAnue` consistently achieves higher PPVs (between 0.55 and 0.72) than the original tools, up to twice as high as the competitors. Except for the SPLASH data, it also performs better regarding of the total number of true positives.

3.2.7 Runtime & memory consumption

To compare the runtime and memory consumption of `RNAnue` to its competitors, the human datasets (HEK293T, Lymphoblast) were analysed with the original analysis pipelines and `RNAnue`. Figure 3.10 shows the runtime of the individual phases (e.g., pre-processing, alignment, detection). `RNAnue` is faster than `Aligater` (i.e., LIGR-SEQ) but slower than the pipelines from SPLASH and PARIS. The alignment step is one of the main causes in all cases. The extensive filtering, statistical assessment, and the additional clustering step increase the computation time of `RNAnue`. Nevertheless, it is only 2.4 times slower in the worst case. The upper chart in Figure 3.10 displays the time needed to build the genome indexes for the respective mapping tools. Although these are one-time costs and heavily depend on the size of the genome to be indexed, they may significantly impact the total time of analysis. The maximum resident set size (max. RSS) was 183GB, compared to 3.9GB (`Aligater`), 4.7GB (SPLASH) and 11.3GB (PARIS). In all cases, the alignment tools, due

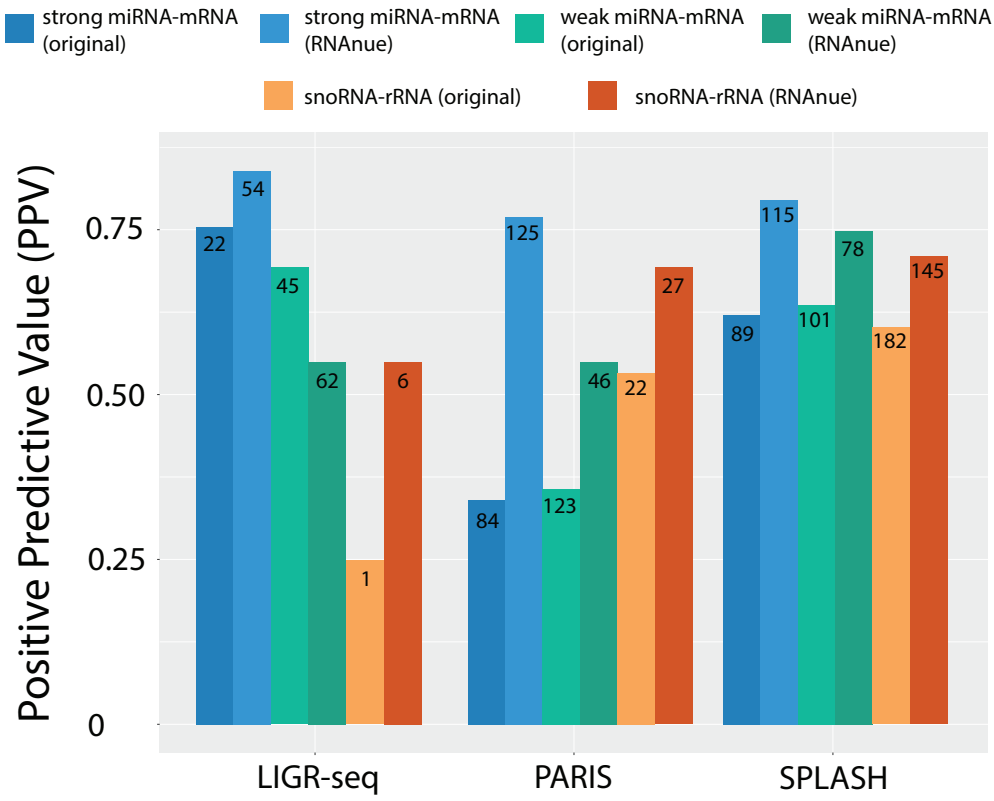

Fig. 3.9 Performance of `RNAnue` in comparison to the original analyses. The positive prediction value (PPV) corresponds to the fraction of detected interactions involving microRNAs that are listed in miRTarBase v7.0 and snoRNA-rRNA interactions listed in snoDB v1.2.1. Here, the numbers within each bar correspond to the total number of true positives.

to the in-memory indices, are responsible for the peak memory consumption. In the case of `segemehl`, and likely also the other tools, the peak is reached during index building. This step must be done only once per genome and can be carried out independently on a large memory server. Without index building, the maximum memory consumption of `segemehl` drops to 60GB.

3.2.8 Implementation

`RNAnue` reports the results of each step in the analysis. At first, the positional arguments `align` or rather `detect` determine different types of reads stored in separate output files in `SAM` format. These are reads that align consecutively against the reference genome (single reads) or consist of unambiguous multiple segments (split reads) or split reads that map to multiple locations on the genome (multi split-reads). In the `clustering` procedure, the results are reported in a single file, describing the detected clusters, their location, size,

and the number of split reads. Finally, `analysis` provides a count table with additional information summarising all detected interactions combined with the filter information.

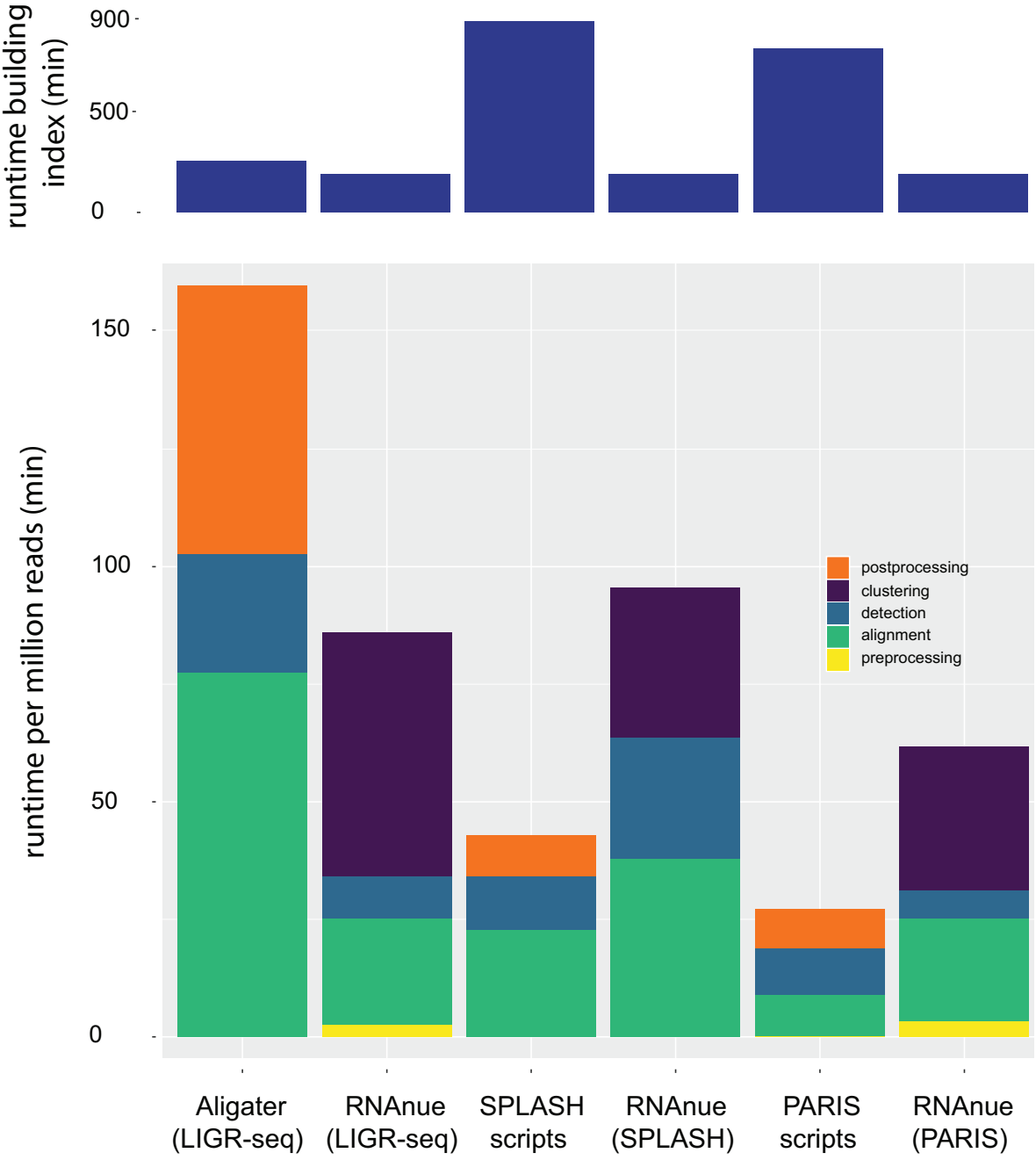

Fig. 3.10 Comparison of the runtime of RNAnue and the original methods for the different analysis steps. The upper graph shows the CPU time needed for building the genome index (GRCh38) of the respective method. It is to be noted that the individual steps correspond to the workflow in the original analyses. Therefore, not all analyses include the same steps (e.g., clustering)

3.2.9 Reconstruction of the secondary structure

As shown in the previous section, RNAnue reveals a multitude of RNA-RNA interactions with high confidence. In principle, this comes with detailed information about the interaction sites; most notably, the filtering scores give an insight into paired bases. In RNAfold, hard constraints on the calculated MFE structure (--constraint) can be induced to guide the structure prediction using prior knowledge. This is done by preventing base pairs from forming or explicitly enforcing them. For that, CompaRNA (Puton et al., 2013) was used, that is a curated benchmark dataset of $\sim$2000 known RNA secondary structures compiled from RNAstrand v2.0 (Andronescu et al., 2008a). In addition, all non-canonical base-pairs, hairpin loops shorter than three bases, and pseudoknots were removed as RNAfold is not able to predict these motifs. In doing so, for each reference structure, three different configurations of (perfect) hard constraints were generated. In these configurations, base pairing is prohibited if the corresponding region in the reference structure is unpaired, and it is enforced if the corresponding region in the reference is paired or a combination of both. The corresponding reference sequence was then subjected to RNAfold (Lorenz et al., 2011) while applying different configurations separately (e.g., unpaired, paired, both). In addition, the prediction was done without constraints (unconstrained). The resulting MFE secondary structure was then assessed by means of the positive predictive value PPV = $\frac{\text{number of correctly predicted base pairs}}{\text{total number of predicted base pairs}}$ and sensitivity = $\frac{\text{number of correctly predicted base pairs}}{\text{total number of true base pairs}}$. Consequently, the 95% confidence interval was bootstrapped using 1000 iterations. As illustrated in Figure 3.11A, guiding the structure prediction using perfect hard constraints (paired and unpaired) yields results with high accuracy (PPV: 0.982, Sensitivity: 0.923). In contrast, the structure prediction without constraints results in low accuracy (PPV: 0.768, Sensitivity: 0.718). The results are comparable when predicting the secondary structure using one-dimensional constraints, in which selected bases are either paired or unpaired. Interestingly, constraining only unpaired positions yield a higher PPV at 0.959, than to 0.914 when only constraining the paired positions. However, in terms of sensitivity, this is the other way around in which paired constraints exhibit a sensitivity of 0.913 as opposed to 0.853 when constraining the unpaired bases. Similarly, information about the interaction sites gained from RNAnue has been integrated into RNAfold. However, for only a fraction of the detected RNAs (see section 3.2.6), a corresponding reference structure could be found in RNAstrand v2.0. This was complemented with information from RNA Frabase v2.0 (Popenda et al., 2010) and URSDB (Baulin et al., 2016). As a result, 138 intramolecular transcript interactions with a known reference structure remain. These include partly redundant 5S, 16S and 23S rRNAs. Consequently, these are supported by at least ten split reads and have a *gcs* of greater than 0.75 in which the complementarity covers at least 50% of the split read ($\theta \geq 0.5$). For each transcript interaction, the alignment

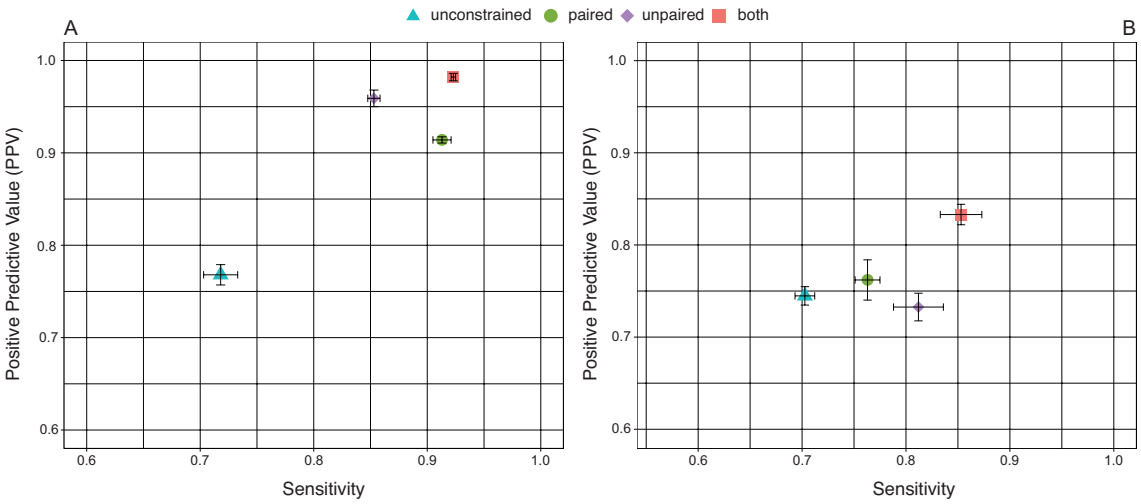

Fig. 3.11 Prediction performance of `RNAfold` using hard constraints. This has been assessed using a benchmarking dataset consisting of RNA structure information build from `RNA Frabase v2.0`, URSDB and `RNAstrand 2.0`. The corresponding dot-bracket notation was used to determine PPV and sensitivity for which a 95% confidence interval was estimated using bootstrapping. (A) `RNAfold` performance using (perfect) constraints extracted from reference structures. Constraining both paired and unpaired bases yields high prediction results. When constraining either one, both PPV and sensitivity are less pronounced but still significantly increased. However, using only paired constraints exhibits a lower PPV as opposed to the restriction of the unpaired positions, whereas the sensitivity is close to the guided prediction using both constraints. (B) `RNAfold` performance using constraints extracted from the filter information of `RNAnue`. This exhibits a similar pattern regarding of PPV and sensitivity but with higher variance and less overall prediction performance.

that has been used to derive the *gcs* reveals a set of complementary bases that are used to build the different configurations (paired, unpaired, both) as before. In principle, within the alignment, a base is considered paired if at least 70% of the corresponding alignments exhibit a complementary base pair at that position. Positions that fall outside of the alignment are unspecified. Subsequently, the rRNA sequence is then subjected to `RNAfold` using the aforementioned configurations. As illustrated in Figure 3.11B, the results are comparable to the prediction with perfect hard constraints. When subjecting `RNAfold` to constraints that either prohibit or enforce base pairing, the correspondig PPV and sensitivity are of similar value (0.73 to 0.76). Finally, using both constraints results in high PPV and sensitivity, above 0.8. However, this is significantly less than with perfect constraints.

3.3 Visualisation and storage of RNA-RNA interactions

Methods to detect RNA-RNA interactions typically report their results in tab-delimited format. For that, a global view of the interaction data can provide insights into RNA-mediated regulation networks. Therefore, visualisation of RNA-RNA interactions is of interest for which graphs are a suited representation. In this work, I developed an interactive graph visualisation named `VisualGraphX` (Schäfer and Voß, 2016) that makes it possible to visualise large-scale graphs. `VisualGraphX` aims to provide a universal graph visualisation tool that empowers users to efficiently explore the data for themselves on a large scale. It has been incorporated into the `Galaxy` platform as an visualisation plugin that is directly accessible through the user history. The JSON Graph Format (JGF) (https://jsongraphformat.info/) is the supported input file format. Both single and multigraph files can be specified. In its minimal form, arrays are required for both the nodes and the edges encapsulated in the graph object. In the case of multiple graphs the (graph) objects are gathered in the graphs array.

3.3.1 Interactive graph visualisation

`VisualGraphX` follows the Model-View-Controller (MVC) concept in order to isolate the *data* (M) from the *presentation* (V) and its *manipulation* (C). For that, the Backbone library (https://backbonejs.org/) provides an unopinionated set of primitives to build single-page applications. Views handle the user input and create the models needed to render the visualisation. In principle, the graph model fetches the user-defined settings, and input data and renders the initial graph onto the viewport that is part of the user interface. This is intertwined with other models that separately store the nodes and edges. In the former, nodes can be either elements that are rendered as shapes or external links to image objects that further reduce the overhead for large-scale graphs. In this regard, a wrapper allows different types of visualisations to be loaded into the graph model. For now, a generic approach is implemented to visualise large-scale graphs interactively. This means, upon setting the parameters for graph depth and the start (root) node, `VisualGraphX` starts precomputing the corresponding subgraph, which is then displayed. Here, the precomputed subgraph can be expanded, contracted and moved by common user interactions, such as double-clicks on nodes. This has been realised using a force-directed layout with the D3 (https://d3js.org/) JavaScript library. As D3 is merely capable of binding data to a Document Object Model (DOM) and then applying data-driven manipulations to the document, it does not include routines for graph traversals as necessary for extension of the subgraph. For that, the graph model is extended to procedures that, among others, allow to the graph to traversed further from a given node. In particular, this is realised using the depth-first search (DFS) algorithm.

Subsequently, changes in the viewport are induced using the user interface that, in turn, modifies the graph model and renders it back to the viewport. `VisualGraphX` aims to be a generic approach for the visualisation of graphs. Therefore, a converter for `Galaxy` is provided that takes as input a simple interaction format (SIF) file with an optional attributes list in tabular format and converts it to a JSON file in the JGF format. It can be installed directly through the `Galaxy` Tool Shed (repository: vgx_converter).

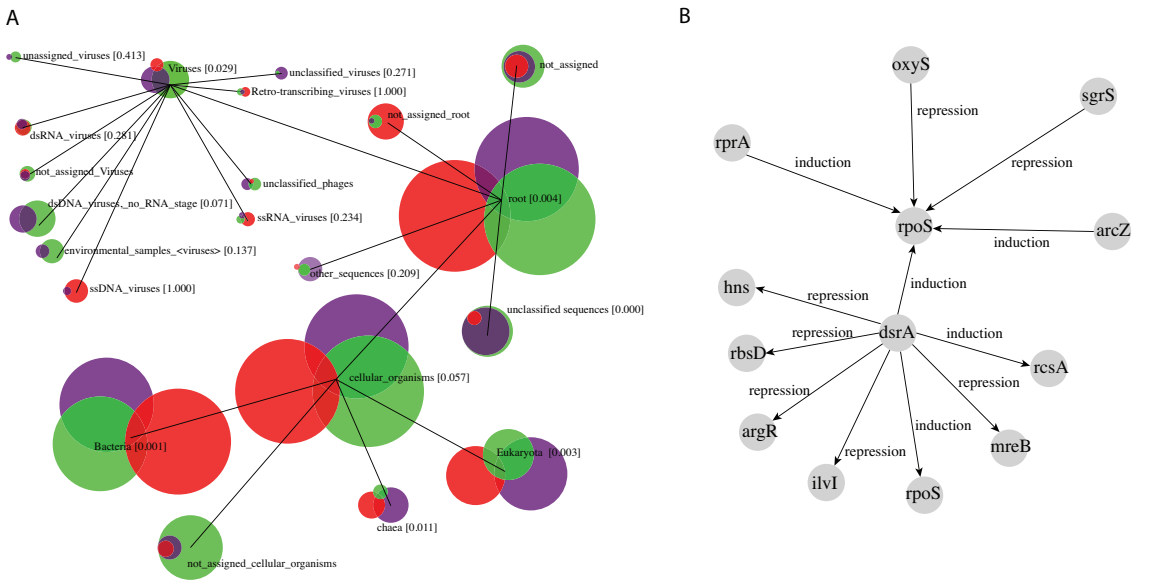

Fig. 3.12 Network visualisation using `VisualGraphX`. (A) Visualisation of the results from a comparative analysis of three 16S rRNA datasets with `CoVennTree` (Lott et al., 2018). (B) Visualisation of an sRNA interaction network in which the nodes are internally generated by `VisualGraphX`.

As an initial use case, `VisualGraphX` has been applied to the results of `CovennTree` (Lott et al., 2015), which is a method for the comparative analysis of large datasets. It generates a rooted tree based on the NCBI taxonomy, in which the nodes are associated with weighted Venn diagrams to illustrate the relation of the different datasets. The creation of the diagrams is outsourced because they are given as URLs to the Google Chart API, which `VisualGraphX` can handle directly. The output of `CoVennTree` consists of a network file that defines the tree's structure and a corresponding attribute file that contains the properties of the nodes. Figure 3.12A illustrates the visualisation of this data using `VisualGraphX`. In a similar manner, Figure 3.12B depicts the visualisation of an sRNA interaction network. Here, the nodes are generated directly within `VisualGraphX`.

3.3.2 Data warehousing of the RNA interactome

Initially, a benchmark of NoSQL databases was performed to compare the capabilities of these containers[1]. Here, the multi-model databases ArangoDB (https://www.arangodb.com) and OrientDB (http://orientdb.org), the document-oriented database MongoDB (https://www.mongodb.com), the graph database Neo4j (https://neo4j.com) and the relation database PostgreSQL jsonb (https://www.postgresql.org) were used. As illustrated in Figure 3.13, ArangoDB, among other metrics, achieved the fastest response time for single document reads from within 100'000 documents. Whereas the document-oriented database MongoDB achieves a similar execution time, PostgreSQL takes moderately longer to access all documents. On the other hand, the execution time of OrientDB and Neo4j is increased by a factor of ~ 1.5. When performing 100'000 write operations, the execution time of ArangoDB, OrientDB, and PostgreSQL is comparable. However, Neo4j and MongoDB take significantly more time for this task, which corresponds to a slow down of factor 2 and 10, respectively. ArangoDB has the second highest memory footprint which is in the same

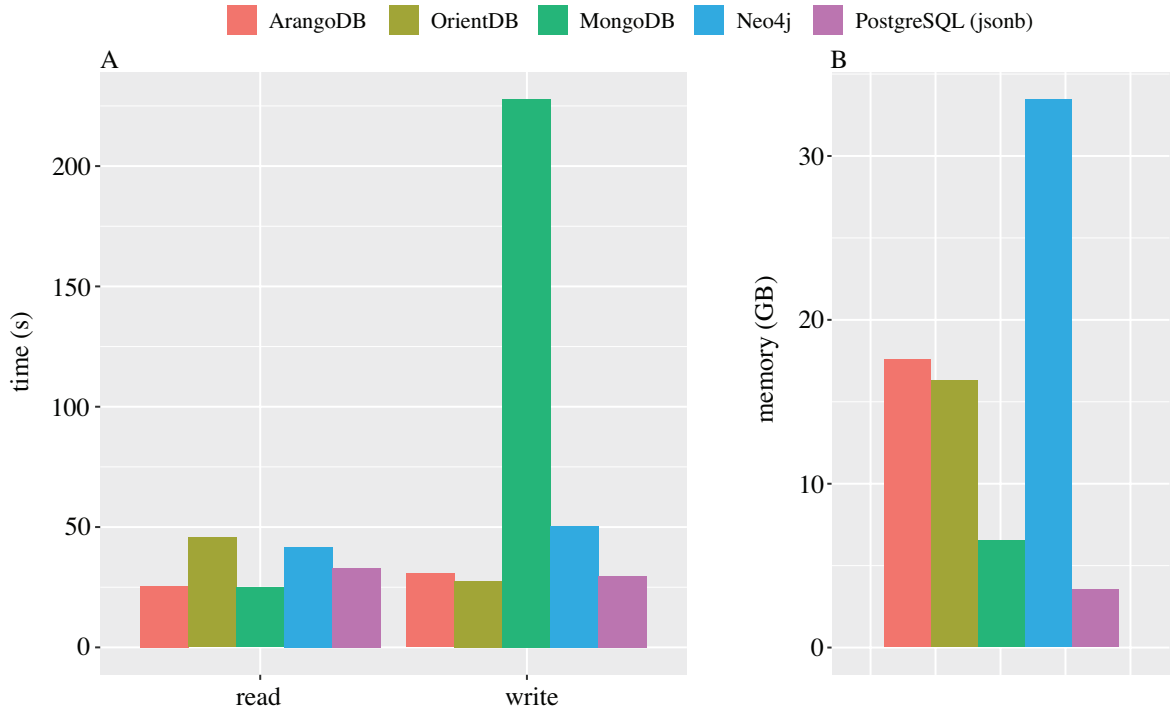

Fig. 3.13 (A) Execution of different databases for 100,000 read and write operations. (B) Memory usage of the databases

range as OrientDB and only exceeded by Neo4j. Consequently, MongoDB and PostgreSQL require only a fraction of ArangoDB's memory. Based on this information, the multi-model

[1] https://github.com/weinberger/nosql-tests

database ArangoDB was used to create a prototype to store RNA-RNA interaction data. It supports multiple models seamlessly in one core system using a single query language. The prototype contains entries of previous analyses done with `RNAnue` (see section 3.2.6) and other databases such as sRNATarBase (Cao et al., 2010), LNCipedia 5 (Volders et al., 2019), snoDB (Bouchard-Bourelle et al., 2020) and miRTarBase v7.0 (Hsu et al., 2011). ArangoDB drivers provide support for native programming languages, which allows the manipulation of data from within native programs. In that regard, `RNAnue` connects to the database and retrieves documents using its specific query language (AQL). This involves queries either using mapping coordinates or identifiers matching those from ENSEMBL (Yates et al., 2020), RefSeq (Pruitt et al., 2005), and other RNA-specific resources. This returns interactions

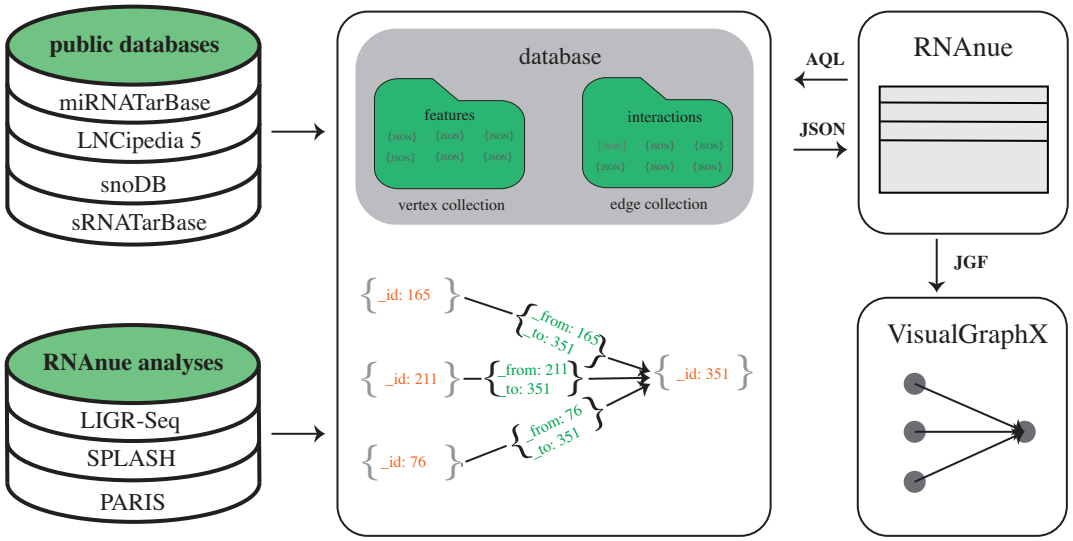

Fig. 3.14 Framework of ArangoDB and its interplay with the tools developed in this work (e.g., `RNAnue`, `VisualGraphX`)

that are associated with the query terms, accompanied with additonal information. This may include scores specific to `RNAnue`, such as the complementarity (*gcs*) and hybridisation energy (*ghs*). `RNAnue` employs this information on intramolecular interactions if available to guide the RNA structure prediction in the reconstruction of the native secondary structure for that RNA (see chapter 3.2.9). In addition, `RNAnue` provides routines to export interactions given a set of aforedescribed identifiers. This returns interaction networks in JGF format, which can be visualised using `VisualGraphX`. Figure 3.14 illustrates the interplay of the database with `VisualGraphX` and `RNAnue`. In ArangoDB, the documents follow the JSON format, although these are internally stored in a binary format. Consequently, the documents contain attributes corresponding to key/value pairs or ordered list of values. The documents are grouped into collections that in turn contains zero or more documents. Most importantly,

the documents are schema-less, which means that the attributes are not predefined and may differ. Collections are distinguished into *vertex collections* and *edge collections* that make up the graph model. For a document within the edge collection, the special attributes *_from* and *_to* describes the relationship between two documents within the vertex collection. This means that in contrast to the mathematical representation of a direct graph, this allows to define documents for the edge relationship. Most importantly, these also define relationship between different vertex collections. Here, the vertex documents correspond to individual RNA entities which are grouped organism-wise into different collections. This allows to define relationships between the entities of different organisms (e.g., homology, identity). The edges then represent the interaction and contain additional information such as the coordinates of the interaction sites or other supporting evidence.

Chapter 4

Discussion

RNA-RNA interactions are crucial in the post-transcriptional regulation of cells in all kingdoms of life. A multitude of different experimental and computational approaches exist to determine these interactions. In the last chapter, two different strategies were introduced. The first utilises prediction algorithms to guide correlation analysis in differential expression profiles. In another data-driven approach, experimental data forms the basis for the algorithmic inference of RNA-RNA interactions. In this chapter, the individual steps will be recapitulated, and what needs to be considered when applying these strategies will be discussed. Subsequently, it will be elaborated on how accurate these different strategies are and will further discuss the limitations of these analyses. In the remainder of this chapter, the means to visualise these large RNA-RNA interaction networks are discussed, followed by their storage in a data warehouse.

4.1 RNA-RNA interaction prediction

RNA-seq experiments do not capture RNA-RNA interactions directly and merely provide hints of RNA-mediated regulation when correlating expression levels between pairs of RNAs, typically ncRNA-mRNA combinations. This is insufficient to determine a connection between them as the number of combinations is inexhaustible. However, performing target prediction for all ncRNAs of interest allows to narrow down the list of potential interactions. As shown in section 3.1, combining RNA-seq data with target prediction algorithms in bacteria allows to determine sRNA-mRNA pairs with high confidence. However, the number of true positives is low, which makes this approach inapplicable for mapping whole interactomes. On the other hand, the target prediction alone is not able to reliably map RNA-RNA interactions on a transcriptome-wide scale. In that regard, the prediction performance of `CopraRNA` which achieves the best results for general bacterial non-specific RNA-RNA inter-

actions (Pain et al., 2015), is shown to decrease with varying input size (Lai and Meyer, 2016). In addition, `CopraRNA` requires a set of homologous sequences for each sRNA of interest from a limited list of compatible genomes. `GLASSgo` and `CopraRNA` have been developed for bacterial small RNAs. As a consequence, the introduced workflow solely predicts bacterial sRNA-mRNA interaction pairs. Furthermore, this provides no information about the RNA structure. `PETcofold` provides a more general approach but requires information from both the query and target sequence. In addition, transcripts with certain properties are hard to detect with this approach. Kozar et al. (2017) showed that the expression levels of numerous miRNAs exhibit a strong co-expression with genes, which, however, could not be predicted as their target. This means that other data-driven methods are required that capture direct interaction signals in a high-throughput manner. Here, *Direct Duplex Detection* (DDD) has emerged as a method to capture RNA-RNA interactions on a global scale. However, in a typical DDD experiment, the number of chimeric reads in relation to all sequenced reads is very low, ranging from 0.5% to 3.9% (Schönberger et al., 2018). When increasing the sequencing depth, single nucleotides are read more often that allows to capture more rare transcripts. However, this also introduces more PCR duplicates than unique reads which is problematic as PCR amplifies different molecules with unequal probabilities (Cha and Thilly, 1993). As a consequence, the analysis of these datasets requires a more elaborate data analysis.

4.2 Pre-processing as a necessity of RNA-seq data

In DDD experiments, chimeric reads are of particular interest and ignoring PCR duplicates would distort the statistical assessment in favour of those reads. For example, some protocols already enrich crosslinked RNAs and ignoring de-duplication would further enhance this biased relation. In addition, chimeric reads occur through controlled ligation events, thereby minimizing the chance of capturing identical chimeric reads randomly. In the absence of UMIs, a computational strategy to remove PCR duplicates involves searching for sequence similarities or using the mapping coordinates to identify identical reads that are removed accordingly. `RNAnue` relies on external tools for the removal of PCR duplicates. As shown in 3.1, `Bbtools dedupe` that works on sequence similarities achieves good results and removes $\sim 93\%$ of the artificially introduced PCR duplicates. In contrast, `SAMtools markdup` removes only $\sim 85\%$. Also, the sequence similarity approach retains almost all of the unique reads (99.9%), while the approach based on mapping coordinates only retains two-thirds ($\sim 67\%$). These findings are interesting because high-throughput technologies increase the chance of observing reads with identical sequences but originating from different cDNA

molecules. This effect is even more pronounced in small genomes or techniques that capture only a subspace of the genome. For example, miRNAs are predicted to account for 1-5% of the human genome (Rajewsky, 2006a). Moreover, a shorter gene is more likely to produce the same RNA-seq reads than a longer gene with an identical transcript level. In contrast, using the mapping to identify identical reads is error-prone to mapping ambiguity, such as in the human genome in which repetitive elements comprise about two-thirds (de Koning et al., 2011). Similarly, rRNAs often contain paralogs of high similarity in sequence such that the corresponding RNA-seq reads map to multiple loci. Fu et al. (2018) showed that PCR de-duplication on the basis of mapping coordinates identifies many false positives, thereby removing transcripts that originate from different molecules. In another study, Ebbert et al. (2016) examined the results in a variant calling analysis using tools for the removal of PCR duplicates on the basis of mapping coordinates. The results suggest that ignoring or removing PCR duplicates has only non-significant effects on the accuracy of variant datasets. Summing up, in RNA-seq experiments with the identical treatment of all transcripts, the bias through PCR duplicates can be neglected. All the more so when the library preparation excludes the use of UMIs. In that regard, different computational methods for the analysis of DDD data that have been examined in this work remove PCR duplicates to a different degree. For example, `Aligater`, introduced for the analysis of LIGR-Seq, completely neglects any PCR removal whatsoever. In contrast, the scripts from SPLASH remove PCR duplicates using mapping coordinates and identical CIGAR strings. In the case of the scripts accompanied by the PARIS protocol, PCR duplicates were removed on the raw sequencing reads. In summary, whenever the removal of PCR artefacts is part of the RNA-Seq workflow, this should be done on the basis of sequence similarities.

Pre-processing

The sequencing reads are subjected to the pre-processing procedure in the following. However, Liao and Shi (2020) argue that read trimming is not required for mapping RNA-Seq reads as these can be removed afterwards using the ‘soft-clipping‘ information. As this seems reasonable in regular RNA-Seq datasets, DDD experiments are more susceptible to these appended sequences. In principle, actual DNA inserts are short fragments originating from different RNA molecules. This alone affects the alignment procedure as short reads may not be mapped unambiguously against the reference genome, and additional adapter contamination distorts the result. In the subsequent analyses, `RNAnue` can remove these ends, as these distort the filtering mechanisms. Nevertheless, this drags along low-quality reads that the pre-processing procedure would have removed. In addition, `RNAnue`’s split read detection is controlled by the read coverage of the spliced segments, which would be affected

by adapter contamination. Similarly, aligners that implement a local alignment procedure allow the pre-processing to be omitted to a certain degree. This is the case in `Aligater`, which internally calls `Bowtie2` for the alignment procedure. However, this does not prevent the mapping of low-quality reads and requires additional computation in the case of split read mapping (Sharma et al., 2016). Moreover, these reads may evade specific filter cutoffs and eventually distort the results. In that regard, other computational approaches considered in this work use different pre-processing routines. However, these need to be called manually and are not incorporated into the analysis workflow. Specifically, `trimmomatic` is applied as part of the PARIS workflow, and `SeqPrep` is used in the pre-processing with the SPLASH scripts. In principle, the analysed datasets from these methods were single-end reads. In `SPLASH`, the initial reads were sequenced in paired-end mode and hence pre-processed and merged using `SeqPrep`. However, only the pre-processed reads could be accessed. Moreover, MCLASH makes use of `Flexbar` but has not been considered further due to the low data quality.

In principle, `trimmomatic` performs a local alignment between the user-defined sequences and the reads. If the alignment scores exceed a user-defined cutoff, the read is trimmed starting from the first aligned position. This approach allows the reliable detection of the sequences, regardless of the location within the read. However, the alignment score is low when the aligned region is short and the adapter sequences match only partly. As a consequence, short adapter sequences remain within the read. This is avoided when paired-end reads are available, and both forward and reverse reads are aligned, thereby revealing technical sequences as overhangs. Similarly, `cutadapt` computes a semi-global alignment. Given the low quantity of split reads in DDD experiments, it is important to reduce adapter contamination as much as possible. In particular, the downstream analysis is susceptible to adapter contamination, which can distort the filtering scores. This is also reflected in initial benchmarks (see Table 3.3), in which pre-processing a DDD dataset with `trimmomatic` results in the highest number of reads but with a moderate number of detected split reads. Based on these insights, `RNAnue` implements a pre-processing procedure based on the Boyer-Moore string-matching algorithm (Boyer and Moore, 1977). As of today, it is the standard benchmark for the practical string-search literature (Hume and Sunday, 1991). It is based on the idea that by matching the pattern from the right rather than from the left, regions containing matches can be quickly identified and skipped, which results in a significant speed-up. This means, however, that the algorithm runs faster as the pattern length increases. Let m denote the length of the input text to be searched and n the length of the pattern to be searched for. In its original form, the Boyer-Moore algorithm had an upper bound of $\mathcal{O}(n+m)$ if the pattern does not appear in the text and $\mathcal{O}(nm)$ otherwise. Cole (1991)

Table 4.1 Overview of pre-processing and alignment tools for DDD data analysis

method	pre-processing	alignment
Aligater	-	Bowtie2
Mclash scripts	Flexbar	BLAST
PARIS scripts	Trimmomatic	Star
RNAnue	Boyer-Moore-based	Segemehl
SPLASH scripts	-	Bwa, STAR

gave proof with an upper bound of $3n$ comparison and other variants guaranteed at most $2n$ comparisons (Crochemore et al., 1994). It has further been shown that, on average, the shift amount is linear in the alphabet size, thus requiring less than n comparisons. RNAnue integrates a variant introduced by Sustik and Moore (2007) that, on the one hand, also works on small alphabets such as DNA, but also reads the characters of the text, not more than once. It involves a pre-processing step in which a transition table is calculated in polynomial time, storing $4m^3$ entries. Although large state transition tables may hurt the performance, moderate lengths in the adapter sequences are bearable, and these have to be calculated only once. This is coherent with the analysis of DDD data, in which the overhead regarding the runtime per million reads is barely perceptible. However, when analysing paired-end reads, this is more pronounced as RNAnue needs to search for the pattern in both pairs separately. The same applies to cutadapt. In contrast, trimmomatic specifically adopts the paired-end nature of the reads and is able to detect adapter read-throughs with high sensitivity and specificity. However, so far, paired-end DDD data is rare, and the strategic advantage of trimmomatic needs to be further investigated. In addition, trimmomatic has a considerable speed benefit over cutadapt and RNAnue. Sustik and Moore (2007) introduced a further improvement to their proposed algorithm that is not further considered in this work but is of general interest. However, a substantial drawback of this approach is that RNAnue fails to detect gapped reads. It is based on an exact matching algorithm, thereby either detecting perfect or imperfect matches, but does not consider indels or deletion. As the benchmark in 3.3 suggests, this seems to be not an issue, and it is more important to allow errors when searching for adapter sequences. However, the pre-processing procedure in RNAnue remains an optional step and can, in principle, be performed using alternative tools beforehand.

4.3 Improved split read detection

In the primary data analysis, the read alignment is critical and determines the quality of the data used in the downstream analysis. In the context of DDD experiments, it is particularly

interesting to detect reads containing parts of two interacting RNA molecules, hereinafter referred to as split reads. Common alignment tools, such as Bowtie2 (Langmead and Salzberg, 2012), are not able to assign reads to several locations on the reference sequence and, thus, suboptimal mappings have to be inspected to find a compatible pair that represents the individual mappings of the parts. It can be seen that split reads resemble spliced transcripts that are non-contiguous and, therefore, originate from different locations on the genome. For that, several alignment tools are available, e.g., Tophat2 (Kim et al., 2013), Hisat2 (Kim et al., 2019) and BBMap (Bushnell, 2014) but they rely on splicing-specific features, such as donor- and acceptor-sites, which renders them unsuited for general purpose chimeric read mapping. In that regard, these tools produce favourable results but also report false exon junctions in the output that further need to be filtered out based on the number of supporting alignments (Engström et al., 2013). In contrast, BWA-MEM (Vasimuddin et al., 2019), STAR (Dobin et al., 2013) and segemehl (Hoffmann et al., 2014) offer direct chimeric read mapping.

Initially, the performance of these aligners in detecting split reads was assessed on a dataset with artificially introduced chimeric reads. The results are illustrated in Table 3.5 and Figure 3.6. This illustrates that segemehl could retrieve significantly more chimeric reads than STAR and BWA-MEM. This is also reflected in a benchmark done by Hoffmann et al. (2014), where different read aligners (including STAR, TopHat2) were assessed on simulated data sets with different sets of regular and non-regular splice junctions. In the latter, these include splice junctions that connect opposite strands and splice junctions that connect distant exons. Only segemehl could recall more than 90% of both sets. When only considering non-regular splicing junctions resembling the nature of DDD-derived split reads, STAR only recalls about 55%. Moreover, Otto et al. (2014) compared segemehl with other methods (including STAR, BWA-MEM) when aligning single reads with default parameters. Here, segemehl outperforms the other tools regarding sensitivity and the number of false positives but requires a higher running time. In principle, the considered alignment tools all follow a similar seed-and-extend strategy. At first, the longest exact match for each position within the query read is determined. For that, both segemehl and STAR use suffix arrays that have a linear space-time complexity. Whereas segemehl utilises enhanced suffix arrays (ESA) that come with additional data structures, STAR makes use of uncompressed suffix arrays. In the former, these include the suffix array, tables for the longest common prefix and additional child and suffix link tables. One drawback of this is that the index structure used by segemehl is significantly larger. In contrast, BWA-MEM utilises the Burrows-Wheeler transform and aims to align the reads to the genome with a maximum of three mismatches or gaps. Following this, segemehl aligns complete reads to all unique seed positions within the reference genome,

utilising Myers' semi-global bit-vector alignment technique. For reads that are spliced or part of fusion transcripts, a successful alignment using this semi-global method is often unattainable. Instead, the ESA-based approach locates multiple seeds corresponding to different locations or strands. The algorithm to detect splicing, trans-splicing, or gene fusion sites employs a greedy, score-driven seed-chaining process, which is then refined by a Smith-Waterman-like transitional alignment. This allows `segemehl` to retrieve more split reads than the other methods, as these miss the exact information about the location. Similarly, `STAR` implements the seed search with the computation of a Maximal Mappable Prefix (MMP) that is implemented through uncompressed suffix arrays (SAs). In the second phase of the algorithm, `STAR` constructs the read alignments by piecing together all the seeds that were initially aligned to the genome. If an alignment within one genomic window does not cover the entire read sequences, `STAR` then tries to find multiple windows that cover the entire read, resulting in a split alignment. In contrast, `Aligater` (Sharma et al., 2016) first performs a local alignment of the sequencing reads using `Bowtie2` (`--local`) and subsequently chains the aligned reads to detect the chimeric reads. As mentioned before, this allows `Aligater` to neglect the pre-processing procedure since the local alignment considers substrings of the query sequence. Setting the correct chaining penalty is critical as these alter the resulting chimeric reads. For example, less stringent penalties are more suited for detecting miRNAs. In other words, a more rigorous penalty score is hurtful for small transcripts because the overall score is lower, and the read is eventually filtered out. This was set to the lowest recommended value to guarantee comparability, resembling a more tolerant primary data analysis. As a key idea of `RNAnue` is to process a high number of chimeric reads that are filtered subsequently, `segemehl` was used in the split read alignment. When applied to real datasets obtained from LIGR-Seq, PARIS and SPLASH, this results in an increase of split reads of roughly seven-fold, three-fold and six-fold, respectively (see Table A.3, A.4 and A.5).

`RNAnue` also detects numerous multi-mapped split reads with ambiguous mapping positions. In principle, ignoring such reads is done by most of the DDD workflows considered in this work (e.g., SPLASH, PARIS). This remains unclear in the case of `Aligater`, but it appears this is determined through the score of the chained blocks within the read. In general, neglecting these multi-mapped split reads is done by most read quantifier tools such as `HTSeq-count` (Anders et al., 2015) and `featureCounts` (Liao et al., 2014). This reduces the uncertainty of the data, considering only high-quality reads for the subsequent downstream analyses. On the other hand, this leads to an underestimation of specific biotypes and ignores a substantial part of the sequencing data. In contrast, accepting all valid multi-mapped alignments has the opposite effect and, ultimately, overestimates specific

biotypes. RNAnue handles this by inspecting the filtering information, thereby picking the most probable alignments when making a distinction. However, this only works for a small subset and mainly results in the reads being ignored altogether. As an alternative, RNAnue allows splitting the count of a multi-mapped read evenly among all matching positions. In doing so, this quantifies all biotypes evenly, and they are represented by the portion of their read count. However, this harmed the filtering information on short transcripts and was therefore neglected in the benchmarked analyses. This probably distorts the complementarity and hybridisation energies due to the identical weighting of such multi-mapped split reads.

4.3.1 Consideration of RNA splicing events

In eukaryotic organisms, protein-coding transcripts are often subject to RNA splicing, in which specific regions are cut out, and the flanking regions are spliced together. Consequently, in the analysis of DDD experiments, the split read detection requires further investigation to account for this alteration of the RNA transcripts. It is known that introns both make an important contribution to efficient gene expression (Nott et al., 2003) and occur most frequently in higher organisms. For example, in *S. cerevisiae* only about 4% of all genes contain introns, which comprise almost a third of the mRNA molecules made each hour (Ares et al., 1999). In principle, splicing introduces additional segments that span the sequencing reads and match non-consecutively on the genome. In other words, a segment of a split read is, in turn, divided into multiple segments corresponding to the number of exons in that segment. Consequently, selecting the pair of a segment that makes up the split read gets more complicated. Ideally, the segments resulting from splicing events are merged to retain segments originating from different RNA molecules. A straightforward approach to remove these divisions is to increase the minimal distance between the matching positions on the genome. Subsequently, pairs of segments within the corresponding split read are then excluded to determine the most probable combination. For example, the protein-coding genes in the human genome contain, on average, eight introns of average length of $\sim$3.4kb (Hnilicová and Staněk, 2011). However, much larger introns can also be found, which renders this approach impracticable. This means that certain splicing-induced segments remain unchanged, while others in close proximity, that, for example, stem from RNA structures, are discarded.

In Aligater, this is done by aligning subsequences of the reads located near a ligation site against different NCBI Blast databases. Reads that match at least six base pairs on both sides of the ligation site are removed. Similarly, in SPLASH, the reads that result from the transcriptome alignment are re-aligned against the genome using STAR and removed if

they entirely span the annotated junction site. This seems redundant as STAR can detect the junctions during alignment, given information about the splicing sites is provided. Otto et al. (2014) illustrated that for Illumina short reads and 454 data, the sensitivity using BWA-MEM exceeds STAR, which may have governed the use of BWA-MEM. In addition, the reads are aligned against the transcriptome, RNAnue, and in particular, segemehl is capable of working on the transcriptome but should be applied to the genome to take advantage of the complexity of the data. In doing so, RNAnue aims to match the segments with the prebuild interval B+ tree to determine the pairs most likely originating from splice junctions. More precisely, a pair of segments that match the position within exons with no intermitting bases are concatenated to a single fragment. Consequently, the segments that overlap with regions located beyond exon features are discarded. This makes it possible to remove splicing-induced splits with high certainty but also prevents the detection of introns with more pronounced functions. For example, numerous intronic sequences have been found to encode miRNAs (Ying et al., 2010) as identified in human and mouse genomes (Lin et al., 2003; Rodriguez et al., 2004). It is to be noted that this approach implemented in RNAnue requires information about the exon boundaries, thereby being impractical with incomplete gene annotations. As in the regular split read detection, the filtering scores resolve any ambiguity between pairs of segments, including those that originate from splicing events. In this way, the split segments exhibit a clearer signal regarding complementarity and hybridisation energy.

4.3.2 Increased runtime and memory consumption

Different results were observed when assessing the time and space requirements of the respective methods. This is due to the different alignment tools that are used in each method, as listed in Table 4.2. Although these are one-time costs and heavily depend on the size of the genome to be indexed, they significantly impact the total time of analysis. It is recognisable that RNAnue (or rather segemehl) requires the least amount of time to build the genome index in the analysis of the human datasets. The runtime of Aligater in building the index is marginally higher, whereas the runtime of PARIS and SPLASH is significantly increased. On the other hand, when the actual runtime of the alignment procedure is pairwise compared between RNAnue and the respective method, RNAnue outperforms Aligater but falls short of the original pipelines of SPLASH and PARIS. This is also reflected in benchmarks done by Otto et al. (2014), in which segemehl is the slowest in terms of runtime, whereas STAR is the fastest. Similarly, the maximum resident set size (max. RSS) using RNAnue (183GB) is significantly higher compared to Aligater, and the workflows of SPLASH and PARIS. In all cases, the alignment tools, due to the in-memory indices, are responsible for the peak memory consumption. In the case of segemehl, and likely also the other tools, the peak is

reached during index building. This step needs to be done only once per genome and can also be carried out independently on a large memory server. Without index building, the maximum memory consumption of `segemehl` drops to 60GB, but the application of `RNAnue` while using `segemehl` may not be feasible on a modern computer with <50GB of memory. Due to the fact that modern HPC servers commonly carry $\geq$128GB RAM, the extensive memory requirements of `segemehl`, and thus of `RNAnue`, should not be a major problem. Furthermore, this has been benchmarked on the most recent human genome assembly. Smaller genomes have a smaller memory footprint: *Escherichia coli* 0.7 GB, *Caenorhabditis elegans* 1.5 GB, *Drosophila melanogaster* 2.6 GB or *Arabidopsis thaliana* 1.8 GB (Otto et al., 2014). One of the reasons for the increased runtime and space requirements is that `RNAnue` operates on the reference genome, whereas PARIS and partly SPLASH align against the transcriptome. For example, in the analysed PARIS datasets, this corresponds to a 50-fold increase of bases to be indexed when aligning against the reference genome.

4.4 Aggregation by clustering & annotation

To assess abundances, interactions originating from the same transcript need to be clustered. This can be done based on gene annotation or in a location-based fashion. `RNAnue` uses both because the latter is more reliable, especially for non-model organisms whose genome annotation is often patchy, and the first provides more information. The clustering is based on the mapping positions of both parts of the chimeric reads and requires overlaps for merging. The resulting clusters represent interactions, which can be further merged to transcript interactions based on the annotation (see Section Clustering for details). The final outcome of the clustering can hold split reads (singletons), interactions (clusters not overlapping any annotated feature) and transcript interactions. In principle, the clustering procedure can be done in $\mathcal{O}(nlogn)$. Here, the reads need to be sorted beforehand, which determines the upper limit of the runtime. The transcriptome landscape, in particular for ncRNAs, is still incomplete. This is supported by the fact that numerous regulatory ncRNAs are discovered on a regular basis. Recently, Lorenzi et al. (2021) introduced the human RNA atlas, a collection of RNA annotations reporting thousands of previously unknown microRNAs and lncRNAs. Consequently, the usage of different patches influences the performed analysis; therefore it is recommended that the analysis be redone. In the case of `RNAnue`, an updated build could either lead to novel interactions or affect the significance of previously detected ones. For example, in this work, the most recent patch of the reference genome by the Genome Reference Consortium, GRCh38.p13, has been used. However, this build still contains regions of unknown sequence distributed throughout the genome. Most recently, the

Table 4.2 Overview of computational methods for DDD data analysis

method	aggregation by	statistical assessment	filtering
`Aligater`	annotation	Binomial test	none
MCLASH scripts	annotation	Fisher's exact test	none
PARIS scripts	annotation	none	coverage
`RNAnue`	clustering & annotation	Binomial test	complementarity & hybridization energy
SPLASH scripts	annotation	none	none

Telomere-to-Telomere(T2T) Consortium addressed the remaining 8% of the genome, adding $\sim$ 0.13 Gbp (4.5%) in comparison to GRCh38.p13. The assembly T2T-CHM13v1.1 includes an increase in the number of genes by 3,404 (5.7%) and transcripts by 5,018 (2.2%). In this regard, the number of genes and transcripts that only occur in T2T-CHM13v1.1 are 3,604 and 6,693, respectively. This implies that the additional assembled bases may refine the RNA interactome.

4.4.1 Interval B+ tree

To match the identified split reads with the known annotations, an Interval B+ tree is prefilled with known annotations, and, optionally, detected clusters. The primary structure is just a B+-tree used to organise the endpoints of the intervals. It has one empty leaf node initially. New intervals are inserted into this bucket directly. When a leaf bucket overflows, the data in the bucket are sorted in ascending order, and the middle value, termed midpt, is to be stored as an index in an index bucket at one level higher. The intervals will be stored in two buckets logically, with those lying to the left of midpt in one bucket and those on the right of midpt in another, and the pointers to these buckets will be stored with the index as it is done in a B+-tree. As for those time intervals that happen to cover or hang around midpt, they will be stored in a secondary structure attached to midpt. Using an interval B+ tree has the advantage of integrating all known features of the annotations to match other split reads. Furthermore, it can be built in at most $\mathcal{O}(nlogn)$. For example, other tools for counting certain features, such as `featureCounts`, require a user-defined feature to be considered. In particular, Nellore et al. (2016) analysed human RNA-seq samples from the Sequence Read Archive (SRA) and identified about 18.6% exon-exon junctions in at least 1000 previously unannotated samples.

4.5 Filtering removes uncertain split reads

In any RNA-Seq experiments, the sequencing reads do not span entire transcripts. As a consequence, the originating transcripts are not always uniquely determined. This may result in multi-mappings in which a read aligns with the best score to more than one location on the reference genome. In principle, these multi-mappings are frequent in RNA-seq data involving small molecules such as sRNAs or miRNAs. This is due to the short lengths of the reads and their origin in repetitive regions across the genome. For example, a large proportion of miRNAs are highly conserved, linked through clusters in the genome that are often transcribed as polycistronic RNAs and have similar expression patterns (Kabekkodu et al., 2018). The members of a miRNA cluster often share sequence similarity (Aravin et al., 2003) that contributes to uncertainty in read mapping. In the same manner, roughly 40% of human miRNAs are encoded by multiple loci that derive from gene duplications (Ros et al., 2019) and have identical seed sequences and overall homology (Berezikov, 2011). In DDD RNA-seq data, the occurrence of multi-mappings is even more pronounced because the protocol only retains double-stranded RNA, leading to reads of short length with multiple segments. In the event of a multi-mapping, the read is typically ignored, or one possible alignment is randomly selected. For example, `Bowtie2` picks a pseudo-random integer and reports the corresponding alignment. In contrast, `segemehl`, `STAR` and `BWA-MEM` report multi-mappings as secondary alignments. However, in their original analyses of SPLASH, PARIS and LIGR-Seq multi-mappings are ignored altogether. As `RNAnue` aims to maximise the number of chimeric reads, a trade-off between the number of retained reads and their respective multi-mappings has to be made. This is controlled by the parameters for the minimum length of each fragment within the chimeric read (`--minfraglen`), and the alignment coverage of the read (--minsplicecover). Mayer and Churchman (2016) suggests that when aligning reads against the human reference genome, a minimum length of 18 is best suited to uniquely align the reads. In that regard, Chor et al. (2009) studied the k-mer spectra of more than 100 species in all kingdoms of life, examining the modalities of the distributions. In principle, the empirical distributions are unimodal in non-mammalian and multimodal in mammalian genomes. This implies that there are distinct groups of common and rare k-mers rather than a continually varying distribution. This is indicative of having to set the value to ensure unique mapping to the reference. In principle, `RNAnue` reports all candidate mapping positions for every read. In such an instance, filtering information, such as the global complementarity score (`gcs`) and the global hybridisation score (`ghs`), is used. In the case of multi-mappings, `RNAnue` examines these filtering values and adopts the right combination of split reads by first looking at optimal complementarity values and hybridisation scores. There are other approaches to cope with uncertainty in RNA-seq data,

but these are not considered here. For example, Li et al. (2010) proposed a generative statistical model to account for those reads that are unaccounted for.

4.6 Prediction accuracy and runtime analysis

The detected interactions using `RNAnue` on the human datasets of LIGR-Seq, SPLASH, and PARIS were compared with the original analyses. It can be seen that there is a substantial overlap between `RNAnue` and the analyses done by `Aligater` (LIGR-Seq), SPLASH and PARIS. In addition, interactions that were detected exclusively by `RNAnue` amount to $\sim$29% (LIGR-Seq), $\sim$ 43% (SPLASH), $\sim$ 12% (PARIS). Consequently, the highest discrepancy between the detected interactions can be seen in the SPLASH datasets. This seems obvious as in the original analysis; the reads are initially aligned against a custom transcriptome. Although this includes additions of manually curated ncRNA databases, it mainly contains only well-characterised transcripts. This, in particular, holds true for lncRNAs that are difficult to annotate and for which only physical transcriptomics evidence can be used. Concretely, sequence-specific features, such as open reading frames (ORF), are not present in protein-coding genes. In any case, automated methods to annotate the transcripts are rapid but suffer from low accuracy or incompleteness, while manual methods are much slower but with higher quality. For that, a trade-off has to be made. In that regard, Uszczynska-Ratajczak et al. (2018) lists databases of lncRNAs annotations in the human genome using different methods for annotation. Here the completeness ranges between 4.4% and 71.7%. `RNAnue` not only aligns the reads against the genome, thereby including the whole sequence information but also aggregates the reads separately from any annotation. In doing so, the chimeric reads that overlap existing features are considered as they would have been discarded otherwise. This extends the annotation, effectively assigning more features. In addition, the SPLASH analyses remove the duplicates based on the mapping coordinates. As shown before, this removes fewer duplicates and more unique reads than methods which are based on sequence similarities.

4.7 Reconstruction of the RNA structure

In this work, the developed workflows deduce many intra- and intermolecular interactions. The next step is to learn more about the interaction sites to gain insights into the RNA function. RNA folds into secondary structures through base-pairing, which can further fold into very complex tertiary structures. However, in most cases, it is sufficient to decipher the secondary structure to gain insights into the RNA function while neglecting the tertiary

structure. As stated earlier, predicting the RNA structure for a given input sequence has not improved over time, falling short of 70% in prediction accuracy. However, experimental data can be incorporated into RNA structure prediction algorithms to guide the prediction directly. This has been done using structure probing data in multiple studies (Lorenz et al., 2016b; Sloma and Mathews, 2015). In principle, the data can be incorporated into RNA structure prediction algorithms by means of hard- or soft constraints. To assess the effect of these constraints on RNA structure prediction, in particular using `RNAfold`, perfect hard constraints were used. In other words, base pairs were prohibited and enforced as present in the reference structure of selected RNAs. It has been shown that using no constraints in the prediction results as expected in the typical performance in RNA structure prediction algorithms, as mentioned before. In contrast, constraining both paired and unpaired bases yields almost perfect accuracy. In principle, `RNAfold` will not form MFE secondary structures if this conflicts with the constraints. Nevertheless, the enforcement of certain pairs is occasionally ignored when it contradicts the overall structure. Interestingly, when only prohibiting base pairs, the PPV is higher than only enforcing base pairs. This is plausible as a higher number of bases are unpaired in RNA structure than paired bases, therefore restricting the folding space to a high degree. This is also reflected in the rRNAs that were used in this assessment, which typically contain multiple independent RNA folding domains. In that regard, structure probing experiments conducted on mammalian transcriptomes revealed that individual mRNAs have different propensities to form RNA secondary structure (Mortimer et al., 2014). On the other hand, the sensitivity when only enforcing the base pairs is close to when constraining both paired and unpaired bases. This seems obvious in this setting since the remaining region should not contain any true base pairs that were not predicted (false negatives). Although structure probing experiments such as SHAPE are widely used to generate secondary structure models, they only capture indirect effects. In contrast, Direct Duplex Detection experiments are more informative and directly map the RNA duplexes. In principle, this allows to capture auxiliary structures like pseudoknots. As shown in this work, numerous RNA-RNA interactions, including RNA structures, were detected using `RNAnue`. In the following, this interaction data was incorporated into `RNAfold` in a similar manner as hard constraints, as illustrated in Figure 3.11B. These results exhibit a similar trend as with perfect hard constraints in which constraining unpaired and paired bases yield the best prediction accuracy but the effect is less noticeable. There are multiple reasons for that. In `RNAnue` the global complementarity score `gcs` merely covers the interaction site itself and not the full transcripts. In that regard, the length of the complementarity is crucial as a `gcs` based on shorter length provides less information about the structure that can then be used to guide the structure prediction. However, this can be controlled using the parameter θ that indicates

the portion of the complementarity on the read. In this context, a moderate value of $\theta \geq 0.5$ has been used that could be further increased to capture a broader range of the interaction site. However, it is to be noted that the majority of the RNAs in the compiled benchmark dataset are rRNAs and tRNAs (see Figure A.1), resulting in a bias towards these classes. Consequently, more diverse reference structures are required to evaluate the prediction performance. In any case, DDD adds to the list of powerful high-throughput methods that aid in RNA structure detection. These include, among other methods, single-nucleotide structure probing experiments that determine the single-strandedness of RNA nucleotides. In contrast, DDD methods mainly capture double-stranded regions, thereby serving as a complement to chemical probing experiments (e.g., SHAPE-Seq, DMS-Seq) and other spectroscopic techniques (e.g., NMR, X-ray crystallography, cryoelectron microscopy) in RNA structure analysis. In recent years, this has led to a high abundance of RNA structure data for which deep learning algorithms can aid in the prediction of RNA structure and function. According to Sun et al. (2017), the widespread success of deep learning can be significantly attributed to the availability of extensive annotated datasets. Yu et al. (2022) discuss successful applications of deep learning in the predictions of RNA structures. In general, using deep learning models allows to neglect the specific features of the RNA structure and rather accept the entire sequence into the model. This is a significant advantage over traditional prediction algorithms. For example, prior to v2.1 of the `ViennaRNA` package, G-quadruplex structures (GQS) were not supported but gained more attention due to their emerging role in gene regulation. In addition, not all RNA secondary structure motifs are thermodynamically well-characterised. This holds true for pentaloops, for which Saon and Znosko (2022) present a specific thermodynamic model that can be incorporated into RNA structure prediction software. Although there has been some effort in the field of deep learning to RNA structure analysis, a major problem remains the limited amount of training data. This often leads to overfitting, in which the model only works well on the training data.

4.8 Visualisation and storage of RNA-RNA interactions

Huang et al. (2009) first introduced the concept of RNA-RNA interactions as joint structures that can be represented as graphs. Similarly, Zhu et al. (2007) provide an overview of some of the major biological networks for which graphs are an appropriate mathematical representation. In this work, `VisualGraphX` has been introduced to provide the means to explore large-scale graphs in an interactive manner utilising current web standards. In contrast, common graph visualisation tools are limited in their ability to display thousands of nodes and, thus, compromise in terms of speed and usability. These include regular

desktop applications (e.g., Cytoscape, GEPHI, TULIP) in which manipulation of the data items is not entirely visible and requires resubmission to the application or web service. Here, VisualGraphX delivers direct responsiveness to the user when exploring the data. As opposed to loading the full graph at once, its core idea is to precompute an initial subgraph, which results in a significant speed-up when visualising large-scale graphs. This has been tested in visualising the mTOR pathway (Sabers et al., 1995). In comparison to Cytoscape, VisualGraphX is able to display the graph in less than half the time, corresponding to a significant speed up. This difference is more pronounced when displaying more complex networks exceeding 10'000 nodes. One of the reasons for this advantage is the computation of the subgraph. On the one hand, VisualGraphX only needs to display a subset of nodes and edges that, in turn, require less computational resources. This is important as the force-directed algorithms need to calculate the positions of the nodes and edges and their repulsive forces within the canvas, which results in a high running time. In general, these algorithms have a complexity of at least $\mathcal{O}(n^2)$ in which n corresponds to the number of nodes in the graph (Fruchterman and Reingold, 1991). Also, the ansynchronous design allows VisualGraphX to compute the subgraph as soon as the user specifies the input graph file without noticing. Similarly, additional nodes are computed, which can be added to the canvas without long delays through user interactions. This is implemented as a depth-first and breadth-first search which have a complexity of $\mathcal{O}(n+e)$ in which e corresponds to the number of edges. As a consequence, the more complex the graph, the more noticeable this is. In principle, this corresponds to a tree visualisation with additional edges that need to be considered separately. However, this method requires a starting node within the graph, which is not always known beforehand. If this is the case, VisualGraphX can still visualise the entire graph but no longer has the speed advantage in the graph drawing. On the other hand, VisualGraphX can handle nodes given as URLs from another resource, such as Google Chart API, and thus further lightening the computation costs. In its current form, VisualGraphX implements the common force-directed layout to provide a general visualisation platform. However, for specific use cases, other layouts or graph types that are available in the utilised $d3$ and other libraries may be more suited. In that regard, Cytoscape provides more versatility to visualise graph-based data from various domains. In addition, VisualGraphX is only available as a plugin for the workflow system Galaxy. However, the modular implementation of VisualGraphX using the MVC design pattern allows extension of existing layouts and processing routines easily.

Currently, numerous approaches are available to unravel the complexity of RNA-RNA interactions. These developments generate large data sets that are stored in different resources, although they overlap significantly. In this work, the introduced database aims to lay the

foundation for such unified data storage. Most commonly, scientific studies provide the resulting data in a non-standardised form, such as in tab-delimited text format. Although this can be readily incorporated into subsequent bioinformatic analyses, the data structure is self-contained and limited. In contrast, databases provide structured data storage, which allows to create complex relationships between the data. In terms of RNA-RNA interaction data, several database platforms have been established that store RNA-RNA interaction data. For example, sRNATarBase (Wang et al., 2016) provides a collection of sRNA targets in bacteria. Other resources include NPInter v3.0 (Hao et al., 2016), RAID v2.0 (Yi et al., 2017) and RAIN (Junge et al., 2017). Similarly, RISE (Gong et al., 2018) incorporates RNA-RNA interaction data taken from the original analyses of the DDD methods considered in this work. Querying these resources for single identifiers (e.g., gene, pathway) or a combination thereof returns interactions for these entities. In particular, the search result page features static or interactive interaction maps, functional annotation and other metadata. However, the implemented use cases are limited to simple queries and full access to the interaction data is only given when downloading the data. This typically results in tab-delimited format, which then makes the database obsolete. Although specific interaction formats are available, these go along with certain information loss. As an alternative, a data structure that closely resembles the interaction data is needed. Graphs are the predominant representation of interaction data. In principle, graph databases explicitly store relationships and algorithms on a graph structure. This allows for efficient queries to determine complex relationships and deeper insights into the network structure. However, RNA-RNA interaction data is typically accompanied by metadata that describes the underlying experiments, parameter settings or auxiliary information of the interaction sites. For that, multi-model database systems combine these models, thereby providing a data container that best captures the nature of the data.

Chapter 5

Conclusion and future directions

In the past decade, high-throughput sequencing technologies revealed that most of the genome is transcribed into RNA. Interestingly, a large proportion of the transcriptome is non-coding, which presents an important regulatory layer in all domains of life. As of today, many different classes of regulatory non-coding RNA (ncRNAs) exist, and they interact with distinct biomolecules, including DNA, RNA, and protein. It is, therefore, of great interest to decipher these underlying networks to gain a deep understanding of cellular regulation. RNA-RNA interaction prediction algorithms alone are not capable of considering all biological facors. Thus, they suffer from low accuracy. It is more appropriate to use inference methods that make use of the wealth of high-throughput sequencing data. In this thesis, different algorithmic approaches to decipher RNA-RNA interactomes on a global scale were presented. At first, prediction algorithms were correlated with differentially expressed genes from RNA-Seq data. Although this method is less high-throughput, it requires no direct interaction signals and can provide a starting point for individual RNA-RNA interactions. This approach was further examined by using regular RNA-Seq differential expression studies, and the prediction results with known validated interactions were assessed. It was shown that this allows the identification of individual RNA-RNA interactions with high accuracy, thereby providing the methodology to assist known RNA-RNA prediction algorithms. In combination with other methods, this can aid in interpretation and experimental design to understand the underlying regulation mechanisms. In an alternative data-driven approach, the focus lay on data of so-called *Direct Duplex Detection (DDD)* experiments that employ psoralen-mediated crosslinking to directly identify RNA-RNA duplexes in cells, thereby providing a powerful tool in deciphering the RNA interactome. `RNAnue` was introduced that enhances the standard primary data analysis of these experiments by appropriate filtering, statistical assessment and annotation-independent clustering. In that regard, the prediction accuracy of these methods was explored by analysing the original data and comparing the results with known valid

interactions. RNAnue not only exhibits a higher accuracy than all the original analyses but also detects numerous transcripts falling within unannotated features. In addition, how these results can be leveraged to increase the accuracy of RNA structure prediction algorithms was examined. In the remainder of this thesis, methods to visualise these RNA-RNA interactions were examined. Here, VisualGraphX was introduced for the interactive exploration of large-scale graphs. It has been developed as an efficient Rich Internet Application (RIA) for the workflow system Galaxy designed specifically for fast graph build-up and traversal while consuming fewer resources. Finally, a prototype for storing of RNA-RNA interactions was introduced based on a multi-model database.

5.1 Accuracy and limits

The methods introduced in this work provide the means to decipher RNA-RNA interactomes to different degrees. However, these methods do not come without limitations. In the prediction of RNA-RNA interactions, the combination of RNA-seq data with the workflow utilising GLASSgo and CopraRNA achieves a high prediction accuracy but identifies only a low number of true positives. However, more tolerant settings in the target prediction heavily increase the number of predicted targets, which are only weakly supported by the data. For example, in precomputed results, CopraRNA already determines 69 targets for the sRNA micF with a p-value < 0.01. Similarly, lowering the correlation coefficient for an interaction also reports numerous random sRNA-mRNA pairs. For that, other metrics are required to increase the number of true positives. In addition, this approach is restricted to bacterial small RNAs, and to map these interactions on a global scale, comprehensive sets of homologous sequences are required.

In the case of RNAnue, the accuracy with respect to the positive predictive value (PPV) is higher than in the original analyses with a high number of true positives. However, this assessment is limited to miRNA-mRNA and snoRNA-rRNA interactions as these are well-characterised and experimentally validated. RNAnue can reliably capture these interactions, using the complementarity information. In the case of miRNA-mRNA interactions, the miRNA's seed region is essential for binding to the mRNA and is typically reflected in the complementarity, thereby exhibiting a high gcs. Similarly, C/D box snoRNAs contain long sequence regions which are highly complementary to regions on rRNAs. However, in other classes, the interaction sites are less centred and span large parts of the read which leads to a lower gcs with a high Θ (i.e., aligned portion of the read). Consequently, interactions with such properties are neglected, but applying more tolerant settings (low gcs) introduces false positives. This requires additional cut-offs directed towards the RNA of interest, such

as the number of supporting reads. In principle, the majority of the detected interactions on the basis of DDD methods are related to either rRNAs or small RNAs, where the number of detected lncRNAs is limited. In any case, the prediction performance of `RNAnue` exceeds generalised RNA-RNA prediction algorithms that are based on thermodynamics, which seem to have reached their theoretical limits.

In principle, `RNAnue` can also be used for similar datasets in organisms in which chimeric reads are present. These include, among others, CRAC (Sander et al., 2009) that subsequently resulted in CLASH (Helwak et al., 2013; Helwak and Tollervey, 2014; Kudla et al., 2011; Travis et al., 2014), iPAR-CLIP (Jungkamp et al., 2011), MARIO (Nguyen et al., 2016), GRIL-Seq (Han et al., 2016) and RIL-Seq (Melamed et al., 2020, 2016). Common to all is that they employ UV-mediated crosslinking between RNA-RNA duplexes but are additionally bound to an RNA chaperone. Consequently, the detected chimeric reads are enriched towards protein-bound RNA-RNA interactions, thereby rendering the statistical assessment as impractical. In fact, preliminary results on human AGO-CLASH datasets (Helwak et al., 2013) revealed that significant identified RNA-RNA interactions either exhibit a low number of supporting chimeric reads or have low corresponding filter scores. In particular, when matching the detected interactions with validated miRNA targets using miRTarBase (Chou et al., 2018) and TarBase v8 (Karagkouni et al., 2018), the statistical assessment is not able to define validated interactions. In other words, the validated interactions are equally identified as significant and non-significant. However, the global complementarity and hybridisation energy values that `RNAnue` provides are sufficient to assess the detected RNA-RNA interactions in CLASH datasets.

In any case, these methods generate interaction data that contributes to the global mapping of RNA-RNA interactomes. For that, the visualisation and storage of RNA-RNA interactions are integral to understanding complex regulatory relationships between RNA molecules. `VisualGraphX`, however, only provides generic graph visualisation capabilities and lacks behind the functionalities of software platforms such as `Cytoscape`. It natively implements standard graph traversal algorithms but does not provide extensive functionalities. In terms of data storage, the prototype presented in this work represents a proof of concept that still lacks basic functionality compared to known databases.

5.2 Advancements

As much as these methods introduced in this work have their limits, other features can be introduced to enhance their overall performance. This ranges from changes in the implementation to more complex feature additions.

Pre-processing of sequencing data

In its current form, `RNAnue` does not include a routine to remove PCR duplicates for which external tools such as `BBtools dedupe` have to be utilised. However, there are multiple options to include this step in the workflow. This work shows that methods based on sequence similarity discover most of the PCR duplicates in the data. For that, the `SeqAn` library implements various pairwise alignment routines that can provide the basic functionality to detect duplicated transcripts. Similarly, methods for de-duplication using the alignment information are readily implementable but require additional assessment. For example, the alignment results done with `segemehl` provide information about potential PCR duplicates in the `SAM` flags. Other approaches use of the mapping coordinates and `CIGAR` strings.

In the actual pre-processing implemented in `RNAnue`, the state transition table impacts the performance. This holds true for long sequence patterns in which the table can exhaust the available cache memory. In numerous states of the state transition table, a subsequent mismatch enforces an large shift of the pattern. These states only differ in their matching positions of the blocks and can be described in a more compact manner. Sustik and Moore (2007) propose an algorithm using such a smart transition table and tested it on randomly generated patterns and text of four letter alphabets. Here, the average shift amount falls behind the initial algorithms while using fewer states. Nevertheless, the initial algorithm it at a disadvantage for long patterns. It therefore needs to be investigated at which properties of the pattern the algorithm benefits from this modification. Other improvements which should be pursued include the implementation details or heuristics when creating the state transition table.

Primary data analysis

`RNAnue` currently accepts only the alignment results from `segemehl` as the aligner reports the split reads using custom SAM tags. This allows `segemehl` to store additional information, such as the predecessing fragments. Another reason lies in the simplicity of the parameter settings, which allows the specification of the minimum length or score of the fragments or the coverage of split read. In addition, one of the main reasons for the usage of `segemehl` was

established in a small benchmark (see section 3.2.2), showing its advantage in the detection of split reads. In principle, minor adjustments in the source code of the split read detection could extend its capabilities such that RNAnue works with other alignment tools. For example, in the SAM format specification, the SA tag is a predefined standard tag. It is supposed to store additional alignments in a chimeric alignment as used by STAR and BWA-MEM. However, using a different alignment tool has no benefit in the split read detection (see Tables A.3, A.4,A.5). Instead, other tools that can rescue previously unmapped reads seems promising. Otto et al. (2014) introduced Lack, a tool accompanying segemehl that was able to re-align 51% of previously unmapped reads. It is of particular interest to investigate on how the split read detection can benefit from this.

Mapping uncertainty

RNAnue also detects numerous multi-mapped split reads with ambiguous mapping positions. For that, a simple method is implemented that divides the count of a split and distributes it evenly across the matching positions. However, more sophisticated methods need to be considered to address these uncertainties. In one approach, the multi-mapped reads are distributed based on the uniquely mapped read ratio. This means that a split read, whose segments contain more uniquely mapped reads, is more likely to get a multi-mapping assigned to. ERANGE (Mortazavi et al., 2008) introduced this strategy that is also included in Cufflinks (Trapnell et al., 2010). However, this depends on the transcript length, in which short ncRNAs are less likely to have more uniquely mapped reads than longer transcripts. Other strategies include inspecting the read coverage upstream and downstream of the matching positions, in which a higher coverage results in a higher portion of the read count. Most commonly, quantification tools use the expectation maximisation (EM) algorithm that estimates the maximum likelihood value of transcript abundance and can handle multi-mapped reads (Deschamps-Francoeur et al., 2020). In that regard, Li et al. (2010) showed that this approach is more accurate than the simple method implemented in RNAnue. Videm et al. (2021) introduced a framework for the analysis of DDD data using EM to handle multi-mapped split reads. In another approach, Robert and Watson (2015) proposed to cluster the multi-mapped reads that map to the same locations into a multi-mapped group (MMG) and are then considered as 'novel' transcript that can be analysed further. All these different strategies can account for multi-mapped reads to some degree, but this does not solve the problem of uncertainty for those reads. Even more so, this can impact the downstream analysis, requiring additional studies to incorporate these reads.

Runtime

In the analysis of DDD data using RNAnue, with the exception of Aligater, the runtime is substantially increased. This is mainly due to the clustering procedure but is in part dependent on the primary data analysis. It is therefore desirable to integrate segemehl into RNAnue on the source or object code as this impacts the runtime of the analysis for which segemehl is a significant contributor. Currently, RNAnue needs to process the alignment results similarly before the different routines (e.g., align, detect, analysis). This can be prevented by direct coupling of segemehl. In addition, this further bundles the external tool dependencies so it can be distributed as one package.

Improvements in RNA structure prediction

This work showed that RNA structure prediction using perfect hard constraints yields almost perfect prediction results. However, this approach is not robust and small errors distort the results. Rather than restricting the folding space, soft constraints add motif-specific pseudo-energies to the free energy contributions of certain loops, thereby guiding the folding process. Most commonly, this is done using SHAPE reactivity data. For that, RNAfold implements three different algorithms to convert these data into pseudo-energies to guide the structure prediction (Lorenz et al., 2016a). In a simple approach, pseudo-energies are estimated for nucleotides involved in stacked helices, while other conformations are not considered (Deigan et al., 2009). By contrast, Zarringhalam et al. (2012) proposed to use SHAPE reactivities to associate the nucleotides with probabilities to be unpaired and subsequently derive pseudo-energy weights for the base pairs. In another approach, Washietl et al. (2012) aims to optimise a vector of pseudo-energies to determine which nucleotides should be unpaired. It needs to be assessed which approach provides the most promising results to reconstitute the initial structure dimerisation. In the following, it needs to be assessed how the interaction data from RNAnue can be utilised to guide the structure prediction using soft constraints. The data needs to be converted into SHAPE reactivities to be incorporated in the structure prediction using the aforementioned approaches. It is known that the probability of an nucleotide to form a canonical base pair is estimated to be inversely proportional to SHAPE reactivities. As RNAnue calculates a complementarity measure, it reports regions with canonical base pairs which can used to derive SHAPE reactivities. Alternatively, these can simply be weighted using complementarity. A robust approach is required to reconstitute the RNA structure reliably. However, reference databases are limited to certain types of RNAs and thus limit the assessment of non-specific RNA structures. Similarly, this approach could be extended to guide the prediction of RNA-RNA interactions. Unfortunately, the reference data for

RNA-RNA interactions is even lower, and some effort is required to assess the applicability of this approach to guide the prediction of intermolecular interactions.

Visualisaton RNA-RNA interactions

So far, `VisualGraphX` provides the means to visualise large-scale graphs in an interactive fashion. In its current state, this is limited to interactions between entities. Kerpedjiev et al. (2015) created `forna` that is a web-based tool, similar to `VisualGraphX` use a force-directed layout but makes it possible to visualise RNA secondary structures on the sequence level. This also allows the integration of multiple structures into the same canvas but is missing the possibility of displaying the secondary structures of two RNAs with dimerisation. In that regard, `RILogo` (Menzel et al., 2012) makes use of sequence logos to visualise intra- and intermolecular base-pairing between two RNA sequences. In principle, the interactions are visualised using arcs or lines within the primary RNA sequence. However, this merely draws an arc diagram in which the nodes (e.g., bases) are placed along a line and do not represent the secondary structure. Moreover, sequence logos are static and lack interactivity. The capabilities of known visualisation libraries (e.g., `d3.js`, `cytoscape.js`) in displaying RNA-RNA interactions on the sequence level have not been investigated. This includes native integration of graph visualisation within ArangoDB. In general, `VisualGraphX` is only available as a plugin for the workflow system `Galaxy` in which it is widely applicable. However, architectural changes in the most recent version of `Galaxy` (v22.01) render `VisualGraphX` as inapplicable. For this reason, a standalone version, either as a local instance or on a web server, seems desirable.

Data warehouse expansion

The prototype of the RNA-RNA interaction database introduced in this work currently harbours a limited set public databases and analysis results conducted in this thesis. Other data information, including RNA-associated interactions, can be integrated into the data model with slight modifications in the document properties. However, there is no standardised format to describe these interaction data. Although it lacks some features specific to RNA-based interactions, the JSON Graph Format (JGF) makes it possible to capture the basic graph structure and allows the specification of user-defined properties. `RNAnue` can perform simple queries on the database and export the interaction data into JGF, which can then be visualised using `VisualGraphX`. For these queries, the data is indexed using the pre-defined attributes (e.g., `_id`, `_key`, `_from`, `_to`) which allows to retrieve documents in a short time. However, for other non-indexed attributes (e.g., organism, evidence, synonyms),

a full document scan is required which becomes noticeable in the increased query time. To avoid this, other data indices are required when introducing more complex queries. For that, the built-in search engine *ArangoSearch* is integrated natively into ArangoDB and provides a range of information retrieval features which combine searches on all data models in a single query. In addition, previous queries can be cached such that subsequent requests only require a fraction of the initial query time. It must, therefore, be assessed, which queries are of interest and which indices are required to retrieve the documents in reasonable time. Also, apart from the built-in web interface, the database requires a user front-end to access or manipulate the data. ArangoDB can be integrated into machine learning pipelines. The multi-model concept allows the combination of multiple data streams into features, which in turn can be used by common machine learning frameworks (e.g., tensorflow, skikit-learn). In that regard, ArangoDB is not restricted to common graph queries (e.g., traversal, pattern matching), which are easy to implement using the built-in query language AQL. It further has native support for known graph algorithms but also integrates widely-used graph machine learning libraries. For example, the Deep Graph Library (https://www.dgl.ai/) can directly be integrated into the graph dataset. To this end, the database needs to be supplemented with data resources to provide a reliable data basis for subsequent analyses.

A List of software contributions

`RNAnue`: Workflow for detecting RNA-RNA interactions from DDD experiments.
GNU Public License https://github.com/Ibvt/RNAnue

`GLASSgo`: sRNA homolog finder.
MIT License https://github.com/lotts/GLASSgo/

`VisualGraphX`: a web-based visualisation tool for large-scale graphs
AFL License https://gitlab.com/comptrans/VisualGraphX

References

Abouelhoda, M. I., Kurtz, S., and Ohlebusch, E. (2002). The Enhanced Suffix Array and Its Applications to Genome Analysis. In Algorithms in Bioinformatics, pages 449–463. Springer, Berlin, Germany.

Afgan, E., Baker, D., Batut, B., van den Beek, M., Bouvier, D., Čech, M., Chilton, J., Clements, D., Coraor, N., Grüning, B. A., Guerler, A., Hillman-Jackson, J., Hiltemann, S., Jalili, V., Rasche, H., Soranzo, N., Goecks, J., Taylor, J., Nekrutenko, A., and Blankenberg, D. (2018). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res., 46(W1):W537–W544.

Alkan, C., Karakoç, E., Nadeau, J. H., Sahinalp, S. C., and Zhang, K. (2006). RNA–RNA Interaction Prediction and Antisense RNA Target Search. J. Comput. Biol., 13(2):267–282.

Altman, S. (1990). Nobel lecture. Enzymatic cleavage of RNA by RNA. Biosci. Rep., 10(4):317–337.

Altschul, S. F. (2014). BLAST Algorithm. In eLS. John Wiley & Sons, Ltd, Chichester, England, UK.

Altuvia, S., Zhang, A., Argaman, L., Tiwari, A., and Storz, G. (1998). The Escherichia coli OxyS regulatory RNA represses fhlA translation by blocking ribosome binding. EMBO J., 17(20):6069–6075.

Ameres, S. L., Martinez, J., and Schroeder, R. (2007). Molecular Basis for Target RNA Recognition and Cleavage by Human RISC. Cell, 130(1):101–112.

Anders, S., Pyl, P. T., and Huber, W. (2015). HTSeq—a Python framework to work with high-throughput sequencing data. Bioinformatics, 31(2):166–169.

Andronescu, M., Bereg, V., Hoos, H. H., and Condon, A. (2008a). RNA STRAND: The RNA Secondary Structure and Statistical Analysis Database. BMC Bioinf., 9(1):1–10.

Andronescu, M., Bereg, V., Hoos, H. H., and Condon, A. (2008b). RNA STRAND: The RNA Secondary Structure and Statistical Analysis Database. BMC Bioinf., 9(1):1–10.

Andronescu, M., Condon, A., Hoos, H. H., Mathews, D. H., and Murphy, K. P. (2007). Efficient parameter estimation for RNA secondary structure prediction. Bioinformatics, 23(13):i19–i28.

Andronescu, M., Zhang, Z. C., and Condon, A. (2005). Secondary Structure Prediction of Interacting RNA Molecules. J. Mol. Biol., 345(5):987–1001.

Aravin, A. A., Lagos-Quintana, M., Yalcin, A., Zavolan, M., Marks, D., Snyder, B., Gaasterland, T., Meyer, J., and Tuschl, T. (2003). The Small RNA Profile during Drosophila melanogaster Development. Dev. Cell, 5(2):337–350.

Ares, M., Grate, L., and Pauling, M. H. (1999). A handful of intron-containing genes produces the lion's share of yeast mRNA. RNA, 5(9):1138–1139.

Argaman, L. and Altuvia, S. (2000). fhlA repression by OxyS RNA: kissing complex formation at two sites results in a stable antisense-target RNA complex11Edited by M. Gottesman. J. Mol. Biol., 300(5):1101–1112.

Auber, D. (2004). TULIP - a huge graph visualization framework. In Graph Drawing Software, pages 105–126. Springer, Berlin, Germany.

Aw, J. G. A., Shen, Y., Wilm, A., Sun, M., Lim, X. N., Boon, K.-L., Tapsin, S., Chan, Y.-S., Tan, C.-P., Sim, A. Y. L., Zhang, T., Susanto, T. T., Fu, Z., Nagarajan, N., and Wan, Y. (2016). In Vivo Mapping of Eukaryotic RNA Interactomes Reveals Principles of Higher-Order Organization and Regulation. Mol. Cell, 62(4):603–617.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., Opperman, D. J., Sagmeister, T., Buhlheller, C., Pavkov-Keller, T., Rathinaswamy, M. K., Dalwadi, U., Yip, C. K., Burke, J. E., Garcia, K. C., Grishin, N. V., Adams, P. D., Read, R. J., and Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. Science, 373(6557):871–876.

Bamezai, S., Rawat, V. P. S., and Buske, C. (2012). Concise Review: The Piwi-piRNA Axis: Pivotal Beyond Transposon Silencing. Stem Cells, 30(12):2603–2611.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., Yefanov, A., Lee, H., Zhang, N., Robertson, C. L., Serova, N., Davis, S., and Soboleva, A. (2013). NCBI GEO: archive for functional genomics data sets—update. Nucleic Acids Research, 41(D1):D991–D995.

Bartolomei, M. S., Zemel, S., and Tilghman, S. M. (1991). Parental imprinting of the mouse H19 gene. Nature, 351(6322):153–155.

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. In Third international AAAI conference on weblogs and social media.

Basu, D. (1980). Randomization Analysis of Experimental Data: The Fisher Randomization Test. J. Am. Stat. Assoc., 75(371):575–582.

Batzoglou, S., Jaffe, D. B., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Berger, B., Mesirov, J. P., and Lander, E. S. (2002). ARACHNE: A Whole-Genome Shotgun Assembler. Genome Res., 12(1):177.

Baulin, E., Yacovlev, V., Khachko, D., Spirin, S., and Roytberg, M. (2016). URS DataBase: universe of RNA structures and their motifs. Database, 2016:baw085.

Beisel, C. L. and Storz, G. (2010). Base pairing small RNAs and their roles in global regulatory networks. FEMS Microbiol. Rev., 34(5):866–882.

Bellaousov, S., Reuter, J. S., Seetin, M. G., and Mathews, D. H. (2013). RNAstructure: web servers for RNA secondary structure prediction and analysis. Nucleic Acids Res., 41(W1):W471–W474.

Bengtsson, M., Ståhlberg, A., Rorsman, P., and Kubista, M. (2005). Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. Genome Res., 15(10):1388–1392.

Berezikov, E. (2011). Evolution of microRNA diversity and regulation in animals. Nat. Rev. Genet., 12:846–860.

Bernhart, S. H., Hofacker, I. L., Will, S., Gruber, A. R., and Stadler, P. F. (2008). RNAalifold: improved consensus structure prediction for RNA alignments. BMC Bioinf., 9(1):1–13.

Bernhart, S. H., Tafer, H., Mückstein, U., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2006). Partition function and base pairing probabilities of RNA heterodimers. Algorithms Mol. Biol., 1(1):1–10.

Bhattacharya, A. and Cui, Y. (2016). SomamiR 2.0: a database of cancer somatic mutations altering microRNA–ceRNA interactions. Nucleic Acids Research, 44(D1):D1005–D1010.

Blankenberg, D., Von Kuster, G., Bouvier, E., Baker, D., Afgan, E., Stoler, N., Taylor, J., and Nekrutenko, A. (2014). Dissemination of scientific software with Galaxy ToolShed. Genome Biol., 15(2):1–3.

Bohnsack, M. T., Martin, R., Granneman, S., Ruprecht, M., Schleiff, E., and Tollervey, D. (2009). Prp43 Bound at Different Sites on the Pre-rRNA Performs Distinct Functions in Ribosome Synthesis. Mol. Cell, 36(4):583–592.

Bolger, A. M., Lohse, M., and Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics, 30(15):2114–2120.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D^3 Data-Driven Documents. IEEE Trans. Visual. Comput. Graphics, 17(12):2301–2309.

Bouchard-Bourelle, P., Desjardins-Henri, C., Mathurin-St-Pierre, D., Deschamps-Francoeur, G., Fafard-Couture, É., Garant, J.-M., Elela, S. A., and Scott, M. S. (2020). snoDB: an interactive database of human snoRNA sequences, abundance and interactions. Nucleic Acids Research, 48(D1):D220–D225.

Boyer, R. S. and Moore, J. S. (1977). A Fast String Searching Algorithm. Commun. ACM, 20(10):762–772.

Bozkaya, T. and Ozsoyoglu, M. (2006). Indexing valid time intervals. In Database and Expert Systems Applications, pages 541–550. Springer, Berlin, Germany.

Brennecke, J., Aravin, A. A., Stark, A., Dus, M., Kellis, M., Sachidanandam, R., and Hannon, G. J. (2007). Discrete Small RNA-Generating Loci as Master Regulators of Transposon Activity in Drosophila. Cell, 128(6):1089–1103.

Burley, S. K., Berman, H. M., Kleywegt, G. J., Markley, J. L., Nakamura, H., and Velankar, S. (2017). Protein Data Bank (PDB): The Single Global Macromolecular Structure Archive. In Protein Crystallography, pages 627–641. Springer, New York, NY, USA.

Busch, A., Richter, A. S., and Backofen, R. (2008). IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. Bioinformatics, 24(24):2849–2856.

Bushnell, B. (2014). BBtools Software Package. https://sourceforge.net/projects/bbmap.

Butcher, S. E. and Pyle, A. M. (2011). The Molecular Interactions That Stabilize RNA Tertiary Structure: RNA Motifs, Patterns, and Networks. Acc. Chem. Res., 44(12):1302–1311.

Cao, Y., Wu, J., Liu, Q., Zhao, Y., Ying, X., Cha, L., Wang, L., and Li, W. (2010). sRNATarBase: A comprehensive database of bacterial sRNA targets verified by experiments. RNA, 16(11):2051–2057.

Carmichael, G. G., Weber, K., Niveleau, A., and Wahba, A. J. (1975). The host factor required for RNA phage Qbeta RNA replication in vitro. Intracellular location, quantitation, and purification by polyadenylate-cellulose chromatography. J. Biol. Chem., 250(10):3607–3612.

Cary, M. P. (2007). Biopax–biological pathways exchange language level 3, release candidate 3 (version 0.92) documentation.

Caudy, A. A. and Hannon, G. J. (2004). Induction and Biochemical Purification of RNA-Induced Silencing Complex From Drosophila S2 Cells. In RNA Interference, Editing, and Modification, pages 59–72. Humana Press.

Cech, T. R. (1990). Self-splicing and enzymatic activity of an intervening sequence rna from tetrahymena. Bioscience reports, 10(3):239–261.

Cesana, M., Cacchiarelli, D., Legnini, I., Santini, T., Sthandier, O., Chinappi, M., Tramontano, A., and Bozzoni, I. (2011). A Long Noncoding RNA Controls Muscle Differentiation by Functioning as a Competing Endogenous RNA. Cell, 147(2):358–369.

Cha, R. S. and Thilly, W. G. (1993). Specificity, efficiency, and fidelity of PCR. Genome Research, 3(3):S18–S29.

Chen, L., Zhu, Q.-H., and Kaufmann, K. (2020). Long non-coding RNAs in plants: emerging modulators of gene activity in development and stress responses. Planta, 252(5):92–14.

Chen, W., Yang, S., Zhou, Z., Zhao, X., Zhong, J., Reinach, P. S., and Yan, D. (2017). The Long Noncoding RNA Landscape of the Mouse Eye. Invest. Ophthalmol. Visual Sci., 58(14):6308–6317.

Cheng, J., Guo, J.-M., Xiao, B.-X., Miao, Y., Jiang, Z., Zhou, H., and Li, Q.-N. (2011). piRNA, the new non-coding RNA, is aberrantly expressed in human cancer cells. Clin. Chim. Acta, 412(17):1621–1625.

Chor, B., Horn, D., Goldman, N., Levy, Y., and Massingham, T. (2009). Genomic DNA k-mer spectra: models and modalities. Genome Biol., 10(10):1–10.

Chou, C.-H., Shrestha, S., Yang, C.-D., Chang, N.-W., Lin, Y.-L., Liao, K.-W., Huang, W.-C., Sun, T.-H., Tu, S.-J., Lee, W.-H., Chiew, M.-Y., Tai, C.-S., Wei, T.-Y., Tsai, T.-R., Huang, H.-T., Wang, C.-Y., Wu, H.-Y., Ho, S.-Y., Chen, P.-R., Chuang, C.-H., Hsieh, P.-J., Wu, Y.-S., Chen, W.-L., Li, M.-J., Wu, Y.-C., Huang, X.-Y., Ng, F. L., Buddhakosai, W., Huang, P.-C., Lan, K.-C., Huang, C.-Y., Weng, S.-L., Cheng, Y.-N., Liang, C., Hsu, W.-L., and Huang, H.-D. (2018). miRTarBase update 2018: A resource for experimentally validated microRNA-target interactions. Nucleic Acids Research, 46(D1):D296–D302.

Clemson, C. M., McNeil, J. A., Willard, H. F., and Lawrence, J. B. (1996). XIST RNA paints the inactive X chromosome at interphase: evidence for a novel RNA involved in nuclear/chromosome structure. J. Cell Biol., 132(3):259–275.

Cochrane, G., Karsch-Mizrachi, I., Takagi, T., and Sequence Database Collaboration, I. N. (2016). The International Nucleotide Sequence Database Collaboration. Nucleic Acids Res., 44(D1):D48–D50.

Codd, E. F. (1970). A relational model of data for large shared data banks. Commun. ACM, 13(6):377–387.

Cole, R. (1991). Tight bounds on the complexity of the Boyer-Moore string matching algorithm. In Second Annual ACM Symposium on Discrete Algorithms, pages 224–233.

Compeau, P. E. C., Pevzner, P. A., and Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly. Nat. Biotechnol., 29:987–991.

Crick, F. (1970). Central Dogma of Molecular Biology. Nature, 227(5258):561–563.

Crochemore, M., Czumaj, A., Gasieniec, L., Jarominek, S., Lecroq, T., Plandowski, W., and Rytter, W. (1994). Speeding up two string-matching algorithms. Algorithmica, 12(4):247–267.

Cromar, G. L., Zhao, A., Yang, A., and Parkinson, J. (2015). Hyperscape: visualization for complex biological networks. Bioinformatics, 31(20):3390–3391.

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M., and Li, H. (2021). Twelve years of SAMtools and BCFtools. GigaScience, 10(2):giab008.

de Koning, A. P. J., Gu, W., Castoe, T. A., Batzer, M. A., and Pollock, D. D. (2011). Repetitive Elements May Comprise Over Two-Thirds of the Human Genome. PLoS Genetics, 7(12).

Deigan, K. E., Li, T. W., Mathews, D. H., and Weeks, K. M. (2009). Accurate SHAPE-directed RNA structure determination. Proc. Natl. Acad. Sci. U.S.A., 106(1):97–102.

Deschamps-Francoeur, G., Simoneau, J., and Scott, M. S. (2020). Handling multi-mapped reads in RNA-seq. Comput. Struct. Biotechnol. J., 18:1569–1576.

Ding, Y., Chan, C. Y., and Lawrence, C. E. (2004). Sfold web server for statistical folding and rational design of nucleic acids. Nucleic Acids Res., 32(suppl_2):W135–W141.

Ding, Y. and Lawrence, C. E. (2001). Statistical prediction of single-stranded regions in RNA secondary structure and application to predicting effective antisense target sites and beyond. Nucleic Acids Res., 29(5):1034–1046.

Ding, Y. and Lawrence, C. E. (2003). A statistical sampling algorithm for RNA secondary structure prediction. Nucleic Acids Res., 31(24):7280–7301.

Dirks, R. M. and Pierce, N. A. (2003). A partition function algorithm for nucleic acid secondary structure including pseudoknots. J. Comput. Chem., 24(13):1664–1677.

Dirks, R. M. and Pierce, N. A. (2004). An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. J. Comput. Chem., 25(10):1295–1304.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. Bioinformatics, 29(1):15–21.

Dodt, M., Roehr, J. T., Ahmed, R., and Dieterich, C. (2012). FLEXBAR—Flexible Barcode and Adapter Processing for Next-Generation Sequencing Platforms. Biology, 1(3):895–905.

Ebbert, M. T. W., Wadsworth, M. E., Staley, L. A., Hoyt, K. L., Pickett, B., Miller, J., Duce, J., Initiative, F. t. A. D. N., Kauwe, J. S. K., and Ridge, P. G. (2016). Evaluating the necessity of PCR duplicate removal from next-generation sequencing data and a comparison of approaches. BMC Bioinformatics, 17(Suppl 7).

Eggenhofer, F., Hofacker, I. L., and Höner zu Siederdissen, C. (2016). RNAlien – Unsupervised RNA family model construction. Nucleic Acids Res., 44(17):8433–8441.

Eggenhofer, F., Tafer, H., Stadler, P. F., and Hofacker, I. L. (2011). RNApredator: fast accessibility-based prediction of sRNA targets. Nucleic Acids Res., 39(suppl_2):W149–W154.

Eilbeck, K., Lewis, S. E., Mungall, C. J., Yandell, M., Stein, L., Durbin, R., and Ashburner, M. (2005). The Sequence Ontology: a tool for the unification of genome annotations. Genome Biol., 6(5):1–12.

Elliott, D. and Ladomery, M. (2015). Molecular Biology of RNA. Oxford University Press, Oxford, England, UK.

Engström, P. G., Steijger, T., Sipos, B., Grant, G. R., Kahles, A., Alioto, T., Behr, J., Bertone, P., Bohnert, R., Campagna, D., Davis, C. A., Dobin, A., Engström, P. G., Gingeras, T. R., Goldman, N., Grant, G. R., Guigó, R., Harrow, J., Hubbard, T. J., Jean, G., Kahles, A., Kosarev, P., Li, S., Liu, J., Mason, C. E., Molodtsov, V., Ning, Z., Ponstingl, H., Prins, J. F., Rätsch, G., Ribeca, P., Seledtsov, I., Sipos, B., Solovyev, V., Steijger, T., Valle, G., Vitulo, N., Wang, K., Wu, T. D., Zeller, G., Rätsch, G., Goldman, N., Hubbard, T. J., Harrow, J., Guigó, R., Bertone, P., and The RGASP Consortium (2013). Systematic evaluation of spliced alignment programs for RNA-seq data. Nat. Methods, 10(12):1185–1191.

Fabian, M. R., Sonenberg, N., and Filipowicz, W. (2010). Regulation of mRNA Translation and Stability by microRNAs. Annu. Rev. Biochem., 79(1):351–379.

Farazi, T. A., Juranek, S. A., and Tuschl, T. (2008). The growing catalog of small RNAs and their association with distinct Argonaute/Piwi family members. Development, 135(7):1201–1214.

Flamm, C., Wielach, J., Wolfinger, M. T., Badelt, S., Lorenz, R., and Hofacker, I. L. (2022). Caveats to deep learning approaches to RNA secondary structure prediction. Front. Bioinform., 0.

Fonseca, N. A., Rung, J., Brazma, A., and Marioni, J. C. (2012). Tools for mapping high-throughput sequencing data. Bioinformatics, 28(24):3169–3177.

Franz, M., Lopes, C. T., Huck, G., Dong, Y., Sumer, O., and Bader, G. D. (2016). Cytoscape.js: a graph theory library for visualisation and analysis. Bioinformatics, 32(2):309–311.

Freyhult, E. K., Bollback, J. P., and Gardner, P. P. (2006). Exploring genomic dark matter: A critical assessment of the performance of homology search methods on noncoding RNA. Genome Res., 17(1):117–125.

Frith, M. C., Pheasant, M., and Mattick, J. S. (2005). Genomics: The amazing complexity of the human transcriptome. Eur. J. Hum. Genet., 13(8):894–897.

Fruchterman, T. M. J. and Reingold, E. M. (1991). Graph drawing by force-directed placement. Softw.: Pract. Exper., 21(11):1129–1164.

Fu, Y., Wu, P.-H., Beane, T., Zamore, P. D., and Weng, Z. (2018). Elimination of PCR duplicates in RNA-seq and small RNA-seq using unique molecular identifiers. BMC Genomics, 19(1):1–14.

Gan, H., Lin, X., Zhang, Z., Zhang, W., Liao, S., Wang, L., and Han, C. (2011). piRNA profiling during specific stages of mouse spermatogenesis. RNA, 17(7):1191.

Gardner, P. P., Daub, J., Tate, J. G., Nawrocki, E. P., Kolbe, D. L., Lindgreen, S., Wilkinson, A. C., Finn, R. D., Griffiths-Jones, S., Eddy, S. R., and Bateman, A. (2009). Rfam: updates to the RNA families database. Nucleic Acids Res., 37(1):D136–D140.

Georg, J. and Hess, W. R. (2018). Widespread Antisense Transcription in Prokaryotes. In Regulating with RNA in Bacteria and Archaea, pages 191–210. John Wiley & Sons, Ltd., Chichester, England, UK.

Gerlach, W. and Giegerich, R. (2006). GUUGle: a utility for fast exact matching under RNA complementary rules including G–U base pairing. Bioinformatics, 22(6):762–764.

Gesteland Raymond, F., Thomas, C., and Atkins John, F. (2006). The RNA world: the nature of modern RNA suggests a prebiotic RNA. Cold Spring Harbor Laboratory Press.

Gong, J., Shao, D., Xu, K., Lu, Z., Lu, Z. J., Yang, Y. T., and Zhang, Q. C. (2018). RISE: a database of RNA interactome from sequencing experiments. Nucleic Acids Res., 46(D1):D194–D201.

Gottesman, S. (2010). Roles of mRNA Stability, Translational Regulation, and Small RNAs in Stress Response Regulation. In Bacterial Stress Responses, pages 59–73. John Wiley & Sons, Ltd., Chichester, England, UK.

Granneman, S., Kudla, G., Petfalski, E., and Tollervey, D. (2009). Identification of protein binding sites on U3 snoRNA and pre-rRNA by UV cross-linking and high-throughput analysis of cDNAs. Proc. Natl. Acad. Sci. U.S.A., 106(24):9613–9618.

Grivna, S. T., Beyret, E., Wang, Z., and Lin, H. (2006). A novel class of small RNAs in mouse spermatogenic cells. Genes Dev., 20(13):1709–1714.

Gusfield, D. (1997). Linear-Time Construction of Suffix Trees, page 94–121. Cambridge University Press.

Han, J., Haihong, E., Le, G., and Du, J. (2011). Survey on NoSQL database. In 2011 6th International Conference on Pervasive Computing and Applications, pages 363–366. IEEE.

Han, K., Tjaden, B., and Lory, S. (2016). GRIL-seq provides a method for identifying direct targets of bacterial small regulatory RNA by in vivo proximity ligation. Nat. Microbiol., 2(16239):1–10.

Hangauer, M. J., Vaughn, I. W., and McManus, M. T. (2013). Pervasive Transcription of the Human Genome Produces Thousands of Previously Unidentified Long Intergenic Noncoding RNAs. PLos Genet., 9(6):e1003569.

Hao, Y., Wu, W., Li, H., Yuan, J., Luo, J., Zhao, Y., and Chen, R. (2016). NPInter v3.0: an upgraded database of noncoding RNA-associated interactions. Database, 2016:baw057.

Helwak, A., Kudla, G., Dudnakova, T., and Tollervey, D. (2013). Mapping the Human miRNA Interactome by CLASH Reveals Frequent Noncanonical Binding. Cell, 153(3):654–665.

Helwak, A. and Tollervey, D. (2014). Mapping the miRNA interactome by cross-linking ligation and sequencing of hybrids (CLASH). Nat. Protoc., 9:711–728.

Hengge-Aronis, R. (2002). Signal Transduction and Regulatory Mechanisms Involved in Control of the σS (RpoS) Subunit of RNA Polymerase. Microbiol. Mol. Biol. Rev., 66(3):373–395.

Henkel, R., Wolkenhauer, O., and Waltemath, D. (2015). Combining computational models, semantic annotations and simulation experiments in a graph database. Database, 2015.

Hermjakob, H., Montecchi-Palazzi, L., Bader, G., Wojcik, J., Salwinski, L., Ceol, A., Moore, S., Orchard, S., Sarkans, U., Von Mering, C., et al. (2004). The hupo psi's molecular interaction format—a community standard for the representation of protein interaction data. Nature biotechnology, 22(2):177–183.

Hiller, M., Zhang, Z., Backofen, R., and Stamm, S. (2007). Pre-mRNA Secondary Structures Influence Exon Recognition. PLos Genet., 3(11):e204.

Hnilicová, J. and Staněk, D. (2011). Where splicing joins chromatin. Nucleus, 2(3):182–188.

Hoffmann, S., Otto, C., Doose, G., Tanzer, A., Langenberger, D., Christ, S., Kunz, M., Holdt, L. M., Teupser, D., Hackermüller, J., and Stadler, P. F. (2014). A multi-split mapping algorithm for circular RNA, splicing, trans-splicing and fusion detection. Genome Biology, 15(2):1–11.

Hoffmann, S., Otto, C., Kurtz, S., Sharma, C. M., Khaitovich, P., Vogel, J., Stadler, P. F., and Hackermüller, J. (11-Sep-2009). Fast Mapping of Short Sequences with Mismatches, Insertions and Deletions Using Index Structures. PLOS Computational Biology, 5(9):e1000502.

Holmqvist, E. and Vogel, J. (2018). RNA-binding proteins in bacteria. Nat. Rev. Microbiol., 16(10):601–615.

Hsu, S.-D., Lin, F.-M., Wu, W.-Y., Liang, C., Huang, W.-C., Chan, W.-L., Tsai, W.-T., Chen, G.-Z., Lee, C.-J., Chiu, C.-M., Chien, C.-H., Wu, M.-C., Huang, C.-Y., Tsou, A.-P., and Huang, H.-D. (2011). miRTarBase: a database curates experimentally validated microRNA–target interactions. Nucleic Acids Research, 39(1):D163–D169.

Huang, F. W. D., Qin, J., Reidys, C. M., and Stadler, P. F. (2009). Partition function and base pairing probabilities for RNA–RNA interaction prediction. Bioinformatics, 25(20):2646–2654.

Huang, H.-Y., Lin, Y.-C.-D., Li, J., Huang, K.-Y., Shrestha, S., Hong, H.-C., Tang, Y., Chen, Y.-G., Jin, C.-N., Yu, Y., Xu, J.-T., Li, Y.-M., Cai, X.-X., Zhou, Z.-Y., Chen, X.-H., Pei, Y.-Y., Hu, L., Su, J.-J., Cui, S.-D., Wang, F., Xie, Y.-Y., Ding, S.-Y., Luo, M.-F., Chou, C.-H., Chang, N.-W., Chen, K.-W., Cheng, Y.-H., Wan, X.-H., Hsu, W.-L., Lee, T.-Y., Wei, F.-X., and Huang, H.-D. (2020). miRTarBase 2020: updates to the experimentally validated microRNA–target interaction database. Nucleic Acids Research, 48(D1):D148–D154.

Huang, V. and Li, L.-C. (2012). miRNA goes nuclear. RNA Biol., 9(3):269–273.

Huang, Z., Shi, J., Gao, Y., Cui, C., Zhang, S., Li, J., Zhou, Y., and Cui, Q. (2019). HMDD v3.0: a database for experimentally supported human microRNA–disease associations. Nucleic Acids Research, 47(D1):D1013–D1017.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novère, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J., and Of the Sbml Forum:, T. r. (2003). The systems biology markup language (SBML): a medium for. Bioinformatics, 19(4):524–531.

Hume, A. and Sunday, D. (1991). Fast string searching. Softw.: Pract. Exper., 21(11):1221–1248.

Idury, R. M. and Waterman, M. S. (2009). A New Algorithm for DNA Sequence Assembly. J. Comput. Biol., 2(2):291–306.

Ignatov, D. and Johansson, J. (2017). RNA-mediated signal perception in pathogenic bacteria. WIREs RNA, 8(6):e1429.

International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. Nature, 431(7011):931–945.

Jones, B., Stekel, D., Rowe, J., and Fernando, C. (2007). Is there a Liquid State Machine in the Bacterium Escherichia Coli? IEEE.

Jørgensen, M. G., Pettersen, J. S., and Kallipolitis, B. H. (2020). sRNA-mediated control in bacteria: An increasing diversity of regulatory mechanisms. Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms, 1863(5):194504.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596:583–589.

Junge, A., Refsgaard, J. C., Garde, C., Pan, X., Santos, A., Alkan, F., Anthon, C., von Mering, C., Workman, C. T., Jensen, L. J., and Gorodkin, J. (2017). RAIN: RNA–protein Association and Interaction Networks. Database, 2017:baw167.

Jünger, M. and Mutzel, P. (2012). Graph drawing software. Springer Science & Business Media.

Jungkamp, A.-C., Stoeckius, M., Mecenas, D., Grün, D., Mastrobuoni, G., Kempa, S., and Rajewsky, N. (2011). In Vivo and Transcriptome-wide Identification of RNA Binding Protein Target Sites. Mol. Cell, 44(5):828–840.

Kabekkodu, S. P., Shukla, V., Varghese, V. K., Souza, J. D., Chakrabarty, S., and Satyamoorthy, K. (2018). Clustered miRNAs and their role in biological functions and diseases. Biol. Rev., 93(4):1955–1986.

Kai, P., Marie, B., Franziska, M., Sharma Cynthia, M., and Jörg, V. (2010). Evidence for an autonomous 5′ target recognition domain in an Hfq-associated small RNA. Proc. Natl. Acad. Sci. U.S.A., 107(47):20435–20440.

Kalvari, I., Nawrocki, E. P., Ontiveros-Palacios, N., Argasinska, J., Lamkiewicz, K., Marz, M., Griffiths-Jones, S., Toffano-Nioche, C., Gautheret, D., Weinberg, Z., Rivas, E., Eddy, S. R., Finn, R. D., Bateman, A., and Petrov, A. I. (2021). Rfam 14: expanded coverage of metagenomic, viral and microRNA families. Nucleic Acids Res., 49(D1):D192–D200.

Kapranov, P., Cawley, S. E., Drenkow, J., Bekiranov, S., Strausberg, R. L., Fodor, S. P. A., and Gingeras, T. R. (2002). Large-Scale Transcriptional Activity in Chromosomes 21 and 22. Science, 296(5569):916–919.

Karagkouni, D., Paraskevopoulou, M. D., Chatzopoulos, S., Vlachos, I. S., Tastsoglou, S., Kanellos, I., Papadimitriou, D., Kavakiotis, I., Maniou, S., Skoufos, G., Vergoulis, T., Dalamagas, T., and Hatzigeorgiou, A. G. (2018). DIANA-TarBase v8: a decade-long collection of experimentally supported miRNA–gene interactions. Nucleic Acids Res., 46(D1):D239–D245.

Kavita, K., de Mets, F., and Gottesman, S. (2018). New aspects of RNA-based regulation by Hfq and its partner sRNAs. Curr. Opin. Microbiol., 42:53–61.

Kawano, M., Reynolds, A. A., Miranda-Rios, J., and Storz, G. (2005). Detection of 5′- and 3′-UTR-derived small RNAs and cis-encoded antisense RNAs in Escherichia coli. Nucleic Acids Res., 33(3):1040–1050.

Kerpedjiev, P., Hammer, S., and Hofacker, I. L. (2015). Forna (force-directed RNA): Simple and effective online RNA secondary structure diagrams. Bioinformatics, 31(20):3377–3379.

Kertesz, M., Wan, Y., Mazor, E., Rinn, J. L., Nutter, R. C., Chang, H. Y., and Segal, E. (2010). Genome-wide measurement of RNA secondary structure in yeast. Nature, 467:103–107.

Kery, M. B., Feldman, M., Livny, J., and Tjaden, B. (2014). TargetRNA2: identifying targets of small regulatory RNAs in bacteria. Nucleic Acids Res., 42(W1):W124–W129.

Kim, D., Paggi, J. M., Park, C., Bennett, C., and Salzberg, S. L. (2019). Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat. Biotechnol., 37(8):907–915.

Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., Salzberg, S. L., Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., and Salzberg, S. L. (2013). TopHat2: Accurate Alignment of Transcriptomes in the Presence of Insertions, Deletions and Gene Fusions. Genome Biol., 14(4):R36.

Kladwang, W., VanLang, C. C., Cordero, P., and Das, R. (2011). Understanding the Errors of SHAPE-Directed RNA Structure Modeling. Biochemistry, 50(37):8049–8056.

Knudsen, B. and Hein, J. (1999). RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics, 15(6):446–454.

Knudsen, B. and Hein, J. (2003). Pfold: RNA secondary structure prediction using stochastic context-free. Nucleic Acids Res., 31(13):3423–3428.

Köberle, C., Kaufmann, S. H. E., and Patzel, V. (2006). Selecting effective siRNAs based on guide RNA structure. Nat. Protoc., 1:1832–1839.

Kormeier, B. (2014). Data Warehouses in Bioinformatics, pages 111–130. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kornienko, A. E., Guenzl, P. M., Barlow, D. P., and Pauler, F. M. (2013). Gene regulation by the act of long non-coding RNA transcription. BMC Biol., 11(1):1–14.

Kozar, I., Cesi, G., Margue, C., Philippidou, D., and Kreis, S. (2017). Impact of BRAF kinase inhibitors on the miRNomes and transcriptomes of melanoma cells. Biochim. Biophys. Acta, Gen. Subj., 1861(11):Pt.

Kozomara, A., Birgaoanu, M., and Griffiths-Jones, S. (2019). miRBase: from microRNA sequences to function. Nucleic Acids Research, 47(D1):D155–D162.

Kozomara, A. and Griffiths-Jones, S. (2011). miRBase: integrating microRNA annotation and deep-sequencing data. Nucleic Acids Res., 39(1):D152–D157.

Krüger, J. and Rehmsmeier, M. (2006). RNAhybrid: microRNA target prediction easy, fast and flexible. Nucleic Acids Res., 34(suppl_2):W451–W454.

Kudla, G., Granneman, S., Hahn, D., Beggs, J. D., and Tollervey, D. (2011). Cross-linking, ligation, and sequencing of hybrids reveals RNA-RNA interactions in yeast. Proc. Natl. Acad. Sci. U.S.A., 108(24):10010–10015.

Kurtz, S., Choudhuri, J. V., Ohlebusch, E., Schleiermacher, C., Stoye, J., and Giegerich, R. (2001). REPuter: the manifold applications of repeat analysis on a genomic scale. Nucleic Acids Res., 29(22):4633–4642.

Lai, D. and Meyer, I. M. (2016). A comprehensive comparison of general RNA–RNA interaction prediction methods. Nucleic Acids Res., 44(7):e61.

Langmead, B. (2010). Aligning Short Sequencing Reads with Bowtie. Curr. Protoc. Bioinform., 32(1):11.7.1–11.7.14.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. Nature Methods, 9(4):357–359.

Lee, R. C., Feinbaum, R. L., and Ambros, V. (1993). The c. elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14. Cell, 75(5):843–854.

Leontis, N. B., Stombaugh, J., and Westhof, E. (2002). The non-Watson–Crick base pairs and their associated isostericity matrices. Nucleic Acids Res., 30(16):3497–3531.

Li, B., Ruotti, V., Stewart, R. M., Thomson, J. A., and Dewey, C. N. (2010). RNA-Seq gene expression estimation with read mapping uncertainty. Bioinformatics, 26(4):493.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics, 25(14):1754–1760.

Li, J.-R., Tong, C.-Y., Sung, T.-J., Kang, T.-Y., Zhou, X. J., and Liu, C.-C. (2019). CMEP: a database for circulating microRNA expression profiling. Bioinformatics, 35(17):3127–3132.

Li, Y., Qiu, C., Tu, J., Geng, B., Yang, J., Jiang, T., and Cui, Q. (2014). HMDD v2.0: a database for experimentally supported human microRNA and disease associations. Nucleic Acids Research, 42(D1):D1070–D1074.

Liao, X., Li, M., Zou, Y., Wu, F.-X., Yi-Pan, and Wang, J. (2019). Current challenges and solutions of de novo assembly. Quant. Biol., 7(2):90–109.

Liao, Y. and Shi, W. (2020). Read trimming is not required for mapping and quantification of RNA-seq reads at the gene level. NAR Genomics Bioinf., 2(3).

Liao, Y., Smyth, G. K., and Shi, W. (2014). featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics, 30(7):923–930.

Lin, S.-L., Chang, D., Wu, D.-Y., and Ying, S.-Y. (2003). A novel RNA splicing-mediated gene silencing mechanism potential for genome evolution. Biochem. Biophys. Res. Commun., 310(3):754–760.

Liu, J., Carmell, M. A., Rivas, F. V., Marsden, C. G., Thomson, J. M., Song, J.-J., Hammond, S. M., Joshua-Tor, L., and Hannon, G. J. (2004). Argonaute2 Is the Catalytic Engine of Mammalian RNAi. Science, 305(5689):1437–1441.

Liu, T., Zhang, K., Xu, S., Wang, Z., Fu, H., Tian, B., Zheng, X., and Li, W. (2017). Detecting RNA-RNA interactions in E. coli using a modified CLASH method. BMC Genomics, 18(1):1–11.

Lopes, C. T., Franz, M., Kazi, F., Donaldson, S. L., Morris, Q., and Bader, G. D. (2010). Cytoscape Web: an interactive web-based network browser. Bioinformatics, 26(18):2347–2348.

Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). ViennaRNA Package 2.0. Algorithms for Molecular Biology, 6(1):26.

Lorenz, R., Luntzer, D., Hofacker, I. L., Stadler, P. F., and Wolfinger, M. T. (2016a). SHAPE directed RNA folding. Bioinformatics, 32(1):145–147.

Lorenz, R., Wolfinger, M. T., Tanzer, A., and Hofacker, I. L. (2016b). Predicting RNA secondary structures from sequence and probing data. Methods, 103:86–98.

Lorenzi, L., Chiu, H.-S., Avila Cobos, F., Gross, S., Volders, P.-J., Cannoodt, R., Nuytens, J., Vanderheyden, K., Anckaert, J., Lefever, S., Tay, A. P., de Bony, E. J., Trypsteen, W., Gysens, F., Vromman, M., Goovaerts, T., Hansen, T. B., Kuersten, S., Nijs, N., Taghon, T., Vermaelen, K., Bracke, K. R., Saeys, Y., De Meyer, T., Deshpande, N. P., Anande, G., Chen, T.-W., Wilkins, M. R., Unnikrishnan, A., De Preter, K., Kjems, J., Koster, J., Schroth, G. P., Vandesompele, J., Sumazin, P., and Mestdagh, P. (2021). The RNA Atlas expands the catalog of human non-coding RNAs. Nat. Biotechnol., 39:1453–1465.

Lott, S. C., Schäfer, R. A., Mann, M., Backofen, R., Hess, W. R., Voß, B., and Georg, J. (2018). GLASSgo – Automated and Reliable Detection of sRNA Homologs From a Single Input Sequence. Front. Genet., 0.

Lott, S. C., Voß, B., Hess, W. R., and Steglich, C. (2015). CoVennTree: a new method for the comparative analysis of large datasets. Front. Genet., 0.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol., 15(12):1–21.

Lu, Z., Gong, J., and Zhang, Q. C. (2018). PARIS: Psoralen Analysis of RNA Interactions and Structures with High Throughput and Resolution. In RNA Detection, pages 59–84. Humana Press, New York, NY, New York, NY, USA.

Lu, Z., Zhang, Q. C., Lee, B., Flynn, R. A., Smith, M. A., Robinson, J. T., Davidovich, C., Gooding, A. R., Goodrich, K. J., Mattick, J. S., Mesirov, J. P., Cech, T. R., and Chang, H. Y. (2016). RNA Duplex Map in Living Cells Reveals Higher-Order Transcriptome Structure. Cell, 165(5):1267–1279.

Lucks, J. B., Mortimer, S. A., Trapnell, C., Luo, S., Aviran, S., Schroth, G. P., Pachter, L., Doudna, J. A., and Arkin, A. P. (2011). Multiplexed RNA structure characterization with selective 2'-hydroxyl acylation analyzed by primer extension sequencing (SHAPE-Seq). Proc. Natl. Acad. Sci. U.S.A., 108(27):11063–11068.

Lybecker, M., Bilusic, I., and Raghavan, R. (2014). Pervasive transcription: detecting functional RNAs in bacteria. Transcription, 5(4):e944039.

Lyngsø, R. B. and Pedersen, C. N. S. (2004). RNA Pseudoknot Prediction in Energy-Based Models. J. Comput. Biol., 7(3-4):409–427.

Ma, L., Bajic, V. B., and Zhang, Z. (2013). On the classification of long non-coding RNAs. RNA Biol., 10(6):924–933.

Maglott, D., Ostell, J., Pruitt, K. D., and Tatusova, T. (2011). Entrez Gene: gene-centered information at NCBI. Nucleic Acids Research, 39(suppl_1):D52–D57.

Mann, M., Wright, P. R., and Backofen, R. (2017). IntaRNA 2.0: enhanced and customizable prediction of RNA–RNA interactions. Nucleic Acids Res., 45(W1):W435–W439.

Markham, N. R. and Zuker, M. (2008). UNAFold. In Bioinformatics, pages 3–31. Humana Press.

Martin, K. and Hoffman, B. (2007). An Open Source Approach to Developing Software in a Small Organization. IEEE Software, 24(1):46–53.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal, 17(1):10–12.

Mathews, D. H., Disney, M. D., Childs, J. L., Schroeder, S. J., Zuker, M., and Turner, D. H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc. Natl. Acad. Sci. U.S.A., 101(19):7287.

Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure11Edited by I. Tinoco. J. Mol. Biol., 288(5):911–940.

Mathews, D. H. and Turner, D. H. (2002a). Dynalign: an algorithm for finding the secondary structure common to two RNA sequences11Edited by I. Tinoco. J. Mol. Biol., 317(2):191–203.

Mathews, D. H. and Turner, D. H. (2002b). Experimentally Derived Nearest-Neighbor Parameters for the Stability of RNA Three- and Four-Way Multibranch Loops. Biochemistry, 41(3):869–880.

Mathews David, H., Disney Matthew, D., Childs Jessica, L., Schroeder Susan, J., Michael, Z., and Turner Douglas, H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc. Natl. Acad. Sci. U.S.A., 101(19):7287–7292.

Mauger, D. M. and Weeks, K. M. (2010). Toward global RNA structure analysis. Nat. Biotechnol., 28(11):1178–1179.

Mayer, A. and Churchman, L. S. (2016). Genome-wide profiling of RNA polymerase transcription at nucleotide resolution in human cells with native elongating transcript sequencing. Nat. Protoc., 11(4):813–833.

McClure, R., Balasubramanian, D., Sun, Y., Bobrovskyy, M., Sumby, P., Genco, C. A., Vanderpool, C. K., and Tjaden, B. (2013). Computational analysis of bacterial RNA-Seq data. Nucleic Acids Res., 41(14):e140.

Meister, G., Landthaler, M., Patkaniowska, A., Dorsett, Y., Teng, G., and Tuschl, T. (2004). Human Argonaute2 Mediates RNA Cleavage Targeted by miRNAs and siRNAs. Mol. Cell, 15(2):185–197.

Melamed, S., Adams, P. P., Zhang, A., Zhang, H., and Storz, G. (2020). RNA-RNA Interactomes of ProQ and Hfq Reveal Overlapping and Competing Roles. Mol. Cell, 77(2):411–425.e7.

Melamed, S., Faigenbaum-Romm, R., Peer, A., Reiss, N., Shechter, O., Bar, A., Altuvia, Y., Argaman, L., and Margalit, H. (2018). Mapping the small RNA interactome in bacteria using RIL-seq. Nat. Protoc., 13(1):1–33.

Melamed, S., Peer, A., Faigenbaum-Romm, R., Gatt, Y. E., Reiss, N., Bar, A., Altuvia, Y., Argaman, L., and Margalit, H. (2016). Global Mapping of Small RNA-Target Interactions in Bacteria. Mol. Cell, 63(5):884–897.

Menzel, P., Seemann, S. E., and Gorodkin, J. (2012). RILogo: visualizing RNA–RNA interactions. Bioinformatics, 28(19):2523–2526.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nat. Methods, 5:621–628.

Mortimer, S. A., Kidwell, M. A., and Doudna, J. A. (2014). Insights into RNA structure and function from genome-wide studies. Nat. Rev. Genet., 15(7):469–479.

Muddashetty, R. S., Khanam, T., Kondrashov, A., Bundman, M., Iacoangeli, A., Kremerskothen, J., Duning, K., Barnekow, A., Hüttenhofer, A., Tiedge, H., and Brosius, J. (2002). Poly(A)-binding Protein is Associated with Neuronal BC1 and BC200 Ribonucleoprotein Particles. J. Mol. Biol., 321(3):433–445.

Murray-Rust, P., Rzepa, H. S., and Wright, M. (2001). Development of chemical markup language (cml) as a system for handling complex chemical content. New journal of chemistry, 25(4):618–634.

Mückstein, U., Tafer, H., Bernhart, S. H., Hernandez-Rosales, M., Vogel, J., Stadler, P. F., and Hofacker, I. L. (2008). Translational Control by RNA-RNA Interaction: Improved Computation of RNA-RNA Binding Thermodynamics. In Bioinformatics Research and Development, pages 114–127. Springer, Berlin, Germany.

Nawrocki, E. P. and Eddy, S. R. (2013). Infernal 1.1: 100-fold faster RNA homology searches. Bioinformatics, 29(22):2933–2935.

Nellore, A., Jaffe, A. E., Fortin, J.-P., Alquicira-Hernández, J., Collado-Torres, L., Wang, S., Iii, R. A. P., Karbhari, N., Hansen, K. D., Langmead, B., and Leek, J. T. (2016). Human splicing diversity and the extent of unannotated splice junctions across human RNA-seq samples on the Sequence Read Archive. Genome Biol., 17(1):1–14.

Nguyen, T. C., Cao, X., Yu, P., Xiao, S., Lu, J., Biase, F. H., Sridhar, B., Huang, N., Zhang, K., and Zhong, S. (2016). Mapping RNA–RNA interactome and RNA structure in vivo by MARIO. Nat. Commun., 7(12023):1–12.

Nott, A., Meislin, S. H., and Moore, M. J. (2003). A quantitative analysis of intron effects on mammalian gene expression. RNA, 9(5):607–617.

Nouaille, S., Mondeil, S., Finoux, A.-L., Moulis, C., Girbal, L., and Cocaign-Bousquet, M. (2017). The stability of an mRNA is influenced by its concentration: a potential physical mechanism to regulate gene expression. Nucleic Acids Res., 45(20):11711.

Okamura, K., Ishizuka, A., Siomi, H., and Siomi, M. C. (2004). Distinct roles for Argonaute proteins in small RNA-directed RNA cleavage pathways. Genes Dev., 18(14):1655–1666.

Otto, C., Stadler, P. F., and Hoffmann, S. (2014). Lacking alignments? The next-generation sequencing mapper segemehl revisited. Bioinformatics, 30(13):1837–1843.

Pain, A., Ott, A., Amine, H., Rochat, T., Bouloc, P., and Gautheret, D. (2015). An assessment of bacterial small RNA target prediction programs. RNA Biol., 12(5):509–513.

Paraskevopoulou, M. D. and Hatzigeorgiou, A. G. (2016). Analyzing MiRNA–LncRNA Interactions. In Long Non-Coding RNAs, pages 271–286. Humana Press, New York, NY, New York, NY, USA.

Parekh, S., Ziegenhain, C., Vieth, B., Enard, W., and Hellmann, I. (2016). The impact of amplification on differential expression analyses by RNA-seq - Scientific Reports. Scientific Reports, 6(25533):1–11.

Pavlopoulos, G. A., Secrier, M., Moschopoulos, C. N., Soldatos, T. G., Kossida, S., Aerts, J., Schneider, R., and Bagos, P. G. (2011). Using graph theory to analyze biological networks. BioData Mining, 4(1):10–27.

Peano, C., Pietrelli, A., Consolandi, C., Rossi, E., Petiti, L., Tagliabue, L., De Bellis, G., and Landini, P. (2013). An efficient rRNA removal method for RNA sequencing in GC-rich bacteria. Microb. Inf. Exp., 3(1):1–11.

Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. U.S.A., 85(8):2444–2448.

Peltzer, A., Jäger, G., Herbig, A., Seitz, A., Kniep, C., Krause, J., and Nieselt, K. (2016). EAGER: efficient ancient genome reconstruction. Genome Biology, 17(1):1–14.

Pervouchine, D. D. (2004). IRIS: intermolecular RNA interaction search. Genome Inform., 15(2):92–101.

Peselis, A. and Serganov, A. (2014). Structure and function of pseudoknots involved in gene expression control. Wiley Interdiscip. Rev.: RNA, 5(6):803.

Popenda, M., Szachniuk, M., Blazewicz, M., Wasik, S., Burke, E. K., Blazewicz, J., and Adamiak, R. W. (2010). RNA FRABASE 2.0: an advanced web-accessible database with the capacity to search the three-dimensional fragments within RNA structures. BMC Bioinf., 11:231.

Pruitt, K. D., Tatusova, T., and Maglott, D. R. (2005). NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res., 33(1):D501–D504.

Puton, T., Kozlowski, L. P., Rother, K. M., and Bujnicki, J. M. (2013). CompaRNA: a server for continuous benchmarking of automated methods for RNA secondary structure prediction. Nucleic Acids Res., 41(7):4307–4323.

Raden, M., Ali, S. M., Alkhnbashi, O. S., Busch, A., Costa, F., Davis, J. A., Eggenhofer, F., Gelhausen, R., Georg, J., Heyne, S., Hiller, M., Kundu, K., Kleinkauf, R., Lott, S. C., Mohamed, M. M., Mattheis, A., Miladi, M., Richter, A. S., Will, S., Wolff, J., Wright, P. R., and Backofen, R. (2018). Freiburg RNA tools: a central online resource for RNA-focused research and teaching. Nucleic Acids Res., 46(W1):W25–W29.

Rajewsky, N. (2006a). L(ou)sy miRNA targets? Nature Structural & Molecular Biology, 13(9):754–755.

Rajewsky, N. (2006b). microRNA target predictions in animals. Nat. Genet., 38(6).

Ramani, V., Qiu, R., and Shendure, J. (2015). High-throughput determination of RNA structure by proximity ligation. Nat. Biotechnol., 33(9):980–984.

Rehmsmeier, M., Steffen, P., Höchsmann, M., and Giegerich, R. (2004). Fast and effective prediction of microRNA/target duplexes. RNA, 10(10):1507–1517.

Reinert, K., Dadi, T. H., Ehrhardt, M., Hauswedell, H., Mehringer, S., Rahn, R., Kim, J., Pockrandt, C., Winkler, J., Siragusa, E., Urgese, G., and Weese, D. (2017). The SeqAn C++ template library for efficient sequence analysis: A resource for programmers. J. Biotechnol., 261:157–168.

Reuter, J. S. and Mathews, D. H. (2010). RNAstructure: software for RNA secondary structure prediction and analysis. BMC Bioinf., 11(1):1–9.

Rhoades, M. W., Reinhart, B. J., Lim, L. P., Burge, C. B., Bartel, B., and Bartel, D. P. (2002). Prediction of Plant MicroRNA Targets. Cell, 110(4):513–520.

Richter, A. and Backofen, R. (2012). Accessibility and conservation: General features of bacterial small RNA–mRNA interactions? RNA Biol., 9(7):954–965.

Rintala-Maki, N. D. and Sutherland, L. C. (2009). Identification and characterisation of a novel antisense non-coding RNA from the RBM5 gene locus. Gene, 445(1):7–16.

RNAcentral Consortium (2021). RNAcentral 2021: secondary structure integration, improved sequence search and new member databases. Nucleic Acids Research, 49(D1):D212–D220.

Robert, C. and Watson, M. (2015). Errors in RNA-Seq quantification affect genes of relevance to human disease. Genome Biol., 16(1).

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics, 26(1):139–140.

Rodriguez, A., Griffiths-Jones, S., Ashurst, J. L., and Bradley, A. (2004). Identification of Mammalian microRNA Host Genes and Transcription Units. Genome Res., 14(10a):1902–1910.

Ros, X. B.-D., Kasprzak, W. K., Bhandari, Y., Fan, L., Cavanaugh, Q., Jiang, M., Dai, L., Yang, A., Shao, T.-J., Shapiro, B. A., Wang, Y.-X., and Gu, S. (2019). Structural Differences between Pri-miRNA Paralogs Promote Alternative Drosha Cleavage and Expand Target Repertoires. Cell Rep., 26(2):447–459.e4.

Rubin, D. L., Shah, N. H., and Noy, N. F. (2008). Biomedical ontologies: a functional perspective. Briefings Bioinf., 9(1):75–90.

Sabers, C. J., Martin, M. M., Brunn, G. J., Williams, J. M., Dumont, F. J., Wiederrecht, G., and Abraham, R. T. (1995). Isolation of a Protein Target of the FKBP12-Rapamycin Complex in Mammalian Cells ($_*$). J. Biol. Chem., 270(2):815–822.

Sander, G., Grzegorz, K., Elisabeth, P., and David, T. (2009). Identification of protein binding sites on U3 snoRNA and pre-rRNA by UV cross-linking and high-throughput analysis of cDNAs. Proc. Natl. Acad. Sci. U.S.A., 106(24):9613–9618.

Sanford, J. R., Wang, X., Mort, M., VanDuyn, N., Cooper, D. N., Mooney, S. D., Edenberg, H. J., and Liu, Y. (2009). Splicing factor SFRS1 recognizes a functionally diverse landscape of RNA transcripts. Genome Research, 19(3):381–394.

Sanger, F., Coulson, A. R., Friedmann, T., Air, G. M., Barrell, B. G., Brown, N. L., Fiddes, J. C., Hutchison, C. A., Slocombe, P. M., and Smith, M. (1978). The nucleotide sequence of bacteriophage φX174. J. Mol. Biol., 125(2):225–246.

Saon, Md. S. and Znosko, B. M. (2022). Thermodynamic characterization of naturally occurring RNA pentaloops. RNA, 28(6):832–841.

Sasaki, T., Shiohama, A., Minoshima, S., and Shimizu, N. (2003). Identification of eight members of the argonaute family in the human genome. Genomics, 82(3):323–330.

Sato, K., Kato, Y., Hamada, M., Akutsu, T., and Asai, K. (2011). IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. Bioinformatics, 27(13):i85–i93.

Schäfer, R. A., Lott, S. C., Georg, J., Grüning, B., Hess, W., and Voß, B. (2020a). GLASSgo Setup & Usage. DaRUS.

Schäfer, R. A., Lott, S. C., Georg, J., Grüning, B. A., Hess, W. R., and Voß, B. (2020b). GLASSgo in Galaxy: high-throughput, reproducible and easy-to-integrate prediction of sRNA homologs. Bioinformatics, 36(15):4357–4359.

Schäfer, R. A. and Voß, B. (2021). RNAnue: efficient data analysis for RNA–RNA interactomics. Nucleic Acids Res., 49(10):5493–5501.

Schäfer, R. A. and Voß, B. (2016). VisualGraphX: interactive graph visualization within . Bioinformatics, 32(22):3525–3527.

Schmieder, R. and Edwards, R. (2011). Quality control and preprocessing of metagenomic datasets. Bioinformatics, 27(6):863–864.

Schönberger, B., Schaal, C., Schäfer, R., and Voß, B. (2018). RNA interactomics: recent advances and remaining challenges. F1000Research, 7:1824.

Seemann, S. E., Gorodkin, J., and Backofen, R. (2008). Unifying evolutionary and thermodynamic information for RNA folding of multiple alignments. Nucleic Acids Res., 36(20):6355–6362.

Seemann, S. E., Richter, A. S., Gesell, T., Backofen, R., and Gorodkin, J. (2011). PETcofold: predicting conserved interactions and structures of two multiple alignments of RNA sequences. Bioinformatics, 27(2):211.

Sena, J. A., Galotto, G., Devitt, N. P., Connick, M. C., Jacobi, J. L., Umale, P. E., Vidali, L., and Bell, C. J. (2018). Unique Molecular Identifiers reveal a novel sequencing artefact with implications for RNA-Seq based gene expression analysis - Scientific Reports. Scientific Reports, 8(13121):1–13.

Senmatsu, S., Asada, R., Oda, A., Hoffman, C. S., Ohta, K., and Hirota, K. (2021). lncRNA transcription induces meiotic recombination through chromatin remodelling in fission yeast. Commun. Biol., 4(295):1–10.

Seo, S. W., Kim, D., Latif, H., O'Brien, E. J., Szubin, R., and Palsson, B. O. (2014). Deciphering Fur transcriptional regulatory network highlights its complex role beyond iron metabolism in Escherichia coli. Nat. Commun., 5:4910.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. Genome Res., 13(11):2498–2504.

Sharma, E., Sterne-Weiler, T., O'Hanlon, D., and Blencowe, B. J. (2016). Global Mapping of Human RNA-RNA Interactions. Molecular Cell, 62(4):618–626.

Shimoni, Y., Friedlander, G., Hetzroni, G., Niv, G., Altuvia, S., Biham, O., and Margalit, H. (2007). Regulation of gene expression by small non-coding RNAs: a quantitative view. Mol. Syst. Biol., 3(1):138.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., and Higgins, D. G. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol. Syst. Biol., 7(1):539.

Sloma, M. F. and Mathews, D. H. (2015). Improving RNA Secondary Structure Prediction with Structure Mapping Data. In Methods in Enzymology, volume 553, pages 91–114. Academic Press, Cambridge, MA, USA.

Smith, C., Heyne, S., Richter, A. S., Will, S., and Backofen, R. (2010). Freiburg RNA Tools: a web server integrating IntaRNA, ExpaRNA and LocARNA. Nucleic Acids Res., 38(suppl_2):W373–W377.

States, D. J., Gish, W., and Altschul, S. F. (1991). Improved sensitivity of nucleic acid database searches using application-specific scoring matrices. Methods, 3(1):66–70.

Steffen, P., Voß, B., Rehmsmeier, M., Reeder, J., and Giegerich, R. (2006). RNAshapes: an integrated RNA analysis package based on abstract shapes. Bioinformatics, 22(4):500–503.

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., Iyer, R., Schatz, M. C., Sinha, S., and Robinson, G. E. (2015). Big Data: Astronomical or Genomical? PLoS Biol., 13(7):e1002195.

Storz, G., Opdyke, J. A., and Zhang, A. (2004). Controlling mRNA stability and translation with small, noncoding RNAs. Curr. Opin. Microbiol., 7(2):140–144.

Storz, G. and Papenfort, K. (2018). Global Regulation by CsrA and Its RNA Antagonists. Microbiology Spectrum.

Strobel, E. J., Yu, A. M., and Lucks, J. B. (2018). High-throughput determination of RNA structures. Nat. Rev. Genet., 19:615–634.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 843–852. IEEE.

Sun, X., Zhulin, I., and Wartell, R. M. (2002). Predicted structure and phyletic distribution of the RNA-binding protein Hfq. Nucleic Acids Res., 30(17):3662.

Sustik, M. A. and Moore, J. S. (2007). String Searching over Small Alphabets. Technical Report TR-07-62, Department of Computer Sciences, University of Texas at Austin.

Sükösd, Z., Swenson, M. S., Kjems, J., and Heitsch, C. E. (2013). Evaluating the accuracy of SHAPE-directed RNA secondary structure predictions. Nucleic Acids Res., 41(5):2807–2816.

Swarts, D. C., Makarova, K., Wang, Y., Nakanishi, K., Ketting, R. F., Koonin, E. V., Patel, D. J., and van der Oost, J. (2014). The evolutionary journey of Argonaute proteins. Nat. Struct. Mol. Biol., 21(9):743–753.

Tafer, H. and Hofacker, I. L. (2008). RNAplex: a fast tool for RNA–RNA interaction search. Bioinformatics, 24(22):2657–2663.

Tarazona, S., Furió-Tarí, P., Turrà, D., Pietro, A. D., Nueda, M. J., Ferrer, A., and Conesa, A. (2015). Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package. Nucleic Acids Res., 43(21):e140.

The UniProt Consortium, Bateman, A., Martin, M.-J., Orchard, S., Magrane, M., Agivetova, R., Ahmad, S., Alpi, E., Bowler-Barnett, E. H., Britto, R., Bursteinas, B., Bye-A.-Jee, H., Coetzee, R., Cukura, A., Da Silva, A., Denny, P., Dogan, T., Ebenezer, T., Fan, J., Castro, L. G., Garmiri, P., Georghiou, G., Gonzales, L., Hatton-Ellis, E., Hussein, A., Ignatchenko, A., Insana, G., Ishtiaq, R., Jokinen, P., Joshi, V., Jyothi, D., Lock, A., Lopez, R., Luciani, A., Luo, J., Lussi, Y., MacDougall, A., Madeira, F., Mahmoudy, M., Menchi, M., Mishra, A., Moulang, K., Nightingale, A., Oliveira, C. S., Pundir, S., Qi, G., Raj, S., Rice, D., Lopez, M. R., Saidi, R., Sampson, J., Sawford, T., Speretta, E., Turner, E., Tyagi, N., Vasudev, P., Volynkin, V., Warner, K., Watkins, X., Zaru, R., Zellner, H., Bridge, A., Poux, S., Redaschi, N., Aimo, L., Argoud-Puy, G., Auchincloss, A., Axelsen, K., Bansal, P., Baratin, D., Blatter, M.-C., Bolleman, J., Boutet, E., Breuza, L., Casals-Casas, C., de Castro, E., Echioukh, K. C., Coudert, E., Cuche, B., Doche, M., Dornevil, D., Estreicher,

A., Famiglietti, M. L., Feuermann, M., Gasteiger, E., Gehant, S., Gerritsen, V., Gos, A., Gruaz-Gumowski, N., Hinz, U., Hulo, C., Hyka-Nouspikel, N., Jungo, F., Keller, G., Kerhornou, A., Lara, V., Le Mercier, P., Lieberherr, D., Lombardot, T., Martin, X., Masson, P., Morgat, A., Neto, T. B., Paesano, S., Pedruzzi, I., Pilbout, S., Pourcel, L., Pozzato, M., Pruess, M., Rivoire, C., Sigrist, C., Sonesson, K., Stutz, A., Sundaram, S., Tognolli, M., Verbregue, L., Wu, C. H., Arighi, C. N., Arminski, L., Chen, C., Chen, Y., Garavelli, J. S., Huang, H., Laiho, K., McGarvey, P., Natale, D. A., Ross, K., Vinayaka, C. R., Wang, Q., Wang, Y., Yeh, L.-S., Zhang, J., Ruch, P., and Teodoro, D. (2021). UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res., 49(D1):D480–D489.

Thomson, T. and Lin, H. (2009). The Biogenesis and Function of PIWI Proteins and piRNAs: Progress and Prospect. Annu. Rev. Cell Dev. Biol., 25(1):355–376.

Till, P., Mach, R. L., and Mach-Aigner, A. R. (2018). A current view on long noncoding RNAs in yeast and filamentous fungi. Appl. Microbiol. Biotechnol., 102(17):7319.

Tjaden, B., Goodwin, S. S., Opdyke, J. A., Guillier, M., Fu, D. X., Gottesman, S., and Storz, G. (2006). Target prediction for small, noncoding RNAs in bacteria. Nucleic Acids Res., 34(9):2791–2802.

Tomczak, K., Czerwińska, P., and Wiznerowicz, M. (2015). The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. Contemporary Oncology, 19(1A):A68.

Tong, Z., Cui, Q., Wang, J., and Zhou, Y. (2019). TransmiR v2.0: an updated transcription factor-microRNA regulation database. Nucleic Acids Research, 47(D1):D253–D258.

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat. Biotechnol., 28:511–515.

Travis, A. J., Moody, J., Helwak, A., Tollervey, D., and Kudla, G. (2014). Hyb: A bioinformatics pipeline for the analysis of CLASH (crosslinking, ligation and sequencing of hybrids) data. Methods, 65(3):263–273.

Tripathi, V., Ellis, J. D., Shen, Z., Song, D. Y., Pan, Q., Watt, A. T., Freier, S. M., Bennett, C. F., Sharma, A., Bubulya, P. A., Blencowe, B. J., Prasanth, S. G., and Prasanth, K. V. (2010). The Nuclear-Retained Noncoding RNA MALAT1 Regulates Alternative Splicing by Modulating SR Splicing Factor Phosphorylation. Mol. Cell, 39(6):925–938.

Tsuiji, H., Yoshimoto, R., Hasegawa, Y., Furuno, M., Yoshida, M., and Nakagawa, S. (2011). Competition between a noncoding exon and introns: Gomafu contains tandem UACUAAC repeats and associates with splicing factor-1. Genes Cells, 16(5):479–490.

Ukkonen, E. (1995). On-line construction of suffix trees. Algorithmica, 14(3):249–260.

Ule, J., Jensen, K., Mele, A., and Darnell, R. B. (2005). CLIP: A method for identifying protein–RNA interaction sites in living cells. Methods, 37(4):376–386.

Underwood, J. G., Uzilov, A. V., Katzman, S., Onodera, C. S., Mainzer, J. E., Mathews, D. H., Lowe, T. M., Salama, S. R., and Haussler, D. (2010). FragSeq: transcriptome-wide RNA structure probing using high-throughput sequencing. Nat. Methods, 7:995–1001.

Uszczynska-Ratajczak, B., Lagarde, J., Frankish, A., Guigó, R., and Johnson, R. (2018). Towards a complete map of the human long non-coding RNA transcriptome. Nat. Rev. Genet., 19(9):535.

van Bakel, H., Nislow, C., Blencowe, B. J., and Hughes, T. R. (2010). Most Dark Matter Transcripts Are Associated With Known Genes. PLoS Biol., 8(5):e1000371.

van Nimwegen, E. (2006). Scaling Laws in the Functional Content of Genomes. In Power Laws, Scale-Free Networks and Genome Biology, pages 236–253. Springer, Boston, MA, Boston, MA, USA.

Vasimuddin, M., Misra, S., Li, H., and Aluru, S. (2019). Efficient architecture-aware acceleration of bwa-mem for multicore systems. In 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pages 314–324.

Videm, P., Kumar, A., Zharkov, O., Grüning, B. A., and Backofen, R. (2021). ChiRA: an integrated framework for chimeric read analysis from RNA-RNA interactome and RNA structurome data. GigaScience, 10(2):giaa158.

Viegas, F. B., Wattenberg, M., van Ham, F., Kriss, J., and McKeon, M. (2007). ManyEyes: a Site for Visualization at Internet Scale. IEEE Trans. Visual. Comput. Graphics, 13(6):1121–1128.

Volders, P.-J., Anckaert, J., Verheggen, K., Nuytens, J., Martens, L., Mestdagh, P., and Vandesompele, J. (2019). LNCipedia 5: towards a reference set of human long non-coding RNAs. Nucleic Acids Res., 47(D1):D135–D139.

Wang, J., Liu, T., Zhao, B., Lu, Q., Wang, Z., Cao, Y., and Li, W. (2016). sRNATarBase 3.0: an updated database for sRNA-target interactions in bacteria. Nucleic Acids Res., 44(D1):D248–D253.

Wang, P., Zhi, H., Zhang, Y., Liu, Y., Zhang, J., Gao, Y., Guo, M., Ning, S., and Li, X. (2015). miRSponge: a manually curated database for experimentally supported miRNA sponges and ceRNAs. Database, 2015.

Washietl, S., Hofacker, I. L., Stadler, P. F., and Kellis, M. (2012). RNA folding with soft constraints: reconciliation of probing data and thermodynamic secondary structure prediction. Nucleic Acids Res., 40(10):4261–4272.

Waters, S. A., McAteer, S. P., Kudla, G., Pang, I., Deshpande, N. P., Amos, T. G., Leong, K. W., Wilkins, M. R., Strugnell, R., Gally, D. L., Tollervey, D., and Tree, J. J. (2017). Small RNA interactome of pathogenic E. coli revealed through crosslinking of RNase E. EMBO J., 36(3):374–387.

Watson, J. D. and Crick, F. H. C. (1953). Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. Nature, 171(4356):737–738.

Weidmann, C. A., Mustoe, A. M., and Weeks, K. M. (2016). Direct Duplex Detection: An Emerging Tool in the RNA Structure Analysis Toolbox. Trends Biochem. Sci., 41(9):734–736.

Weiner, P. (1973). Linear pattern matching algorithms. In 14th Annual Symposium on Switching and Automata Theory (swat 1973), pages 1–11. IEEE.

Will, S., Reiche, K., Hofacker, I. L., Stadler, P. F., and Backofen, R. (2007). Inferring Noncoding RNA Families and Classes by Means of Genome-Scale Structure-Based Clustering. PLoS Comput. Biol., 3(4):e65.

Wong, T. N., Sosnick, T. R., and Pan, T. (2007). Folding of noncoding RNAs during transcription facilitated by pausing-induced nonnative structures. Proc. Natl. Acad. Sci. U.S.A., 104(46):17995–18000.

Wright, P. R. and Georg, J. (2018). Workflow for a Computational Analysis of an sRNA Candidate in Bacteria. In Bacterial Regulatory RNA, pages 3–30. Springer, New York, NY, USA.

Wright, P. R., Georg, J., Mann, M., Sorescu, D. A., Richter, A. S., Lott, S., Kleinkauf, R., Hess, W. R., and Backofen, R. (2014). CopraRNA and IntaRNA: predicting small RNA targets, networks and interaction domains. Nucleic Acids Res., 42(W1):W119–W123.

Wright, P. R., Richter, A. S., Papenfort, K., Mann, M., Vogel, J., Hess, W. R., Backofen, R., and Georg, J. (2013a). Comparative genomics boosts target prediction for bacterial small RNAs. Proc. Natl. Acad. Sci. U.S.A., 110(37):E3487–E3496.

Wright, P. R., Richter, A. S., Papenfort, K., Mann, M., Vogel, J., Hess, W. R., Backofen, R., and Georg, J. (2013b). Comparative genomics boosts target prediction for bacterial small RNAs | Proceedings of the National Academy of Sciences. Proc. Natl. Acad. Sci. U.S.A.

Yates, A. D., Achuthan, P., Akanni, W., Allen, J., Allen, J., Alvarez-Jarreta, J., Amode, M. R., Armean, I. M., Azov, A. G., Bennett, R., Bhai, J., Billis, K., Boddu, S., Marugán, J. C., Cummins, C., Davidson, C., Dodiya, K., Fatima, R., Gall, A., Giron, C. G., Gil, L., Grego, T., Haggerty, L., Haskell, E., Hourlier, T., Izuogu, O. G., Janacek, S. H., Juettemann, T., Kay, M., Lavidas, I., Le, T., Lemos, D., Martinez, J. G., Maurel, T., McDowall, M., McMahon, A., Mohanan, S., Moore, B., Nuhn, M., Oheh, D. N., Parker, A., Parton, A., Patricio, M., Sakthivel, M. P., Abdul Salam, A. I., Schmitt, B. M., Schuilenburg, H., Sheppard, D., Sycheva, M., Szuba, M., Taylor, K., Thormann, A., Threadgold, G., Vullo, A., Walts, B., Winterbottom, A., Zadissa, A., Chakiachvili, M., Flint, B., Frankish, A., Hunt, S. E., IIsley, G., Kostadima, M., Langridge, N., Loveland, J. E., Martin, F. J., Morales, J., Mudge, J. M., Muffato, M., Perry, E., Ruffier, M., Trevanion, S. J., Cunningham, F., Howe, K. L., Zerbino, D. R., and Flicek, P. (2020). Ensembl 2020. Nucleic Acids Research, 48(D1):D682–D688.

Yi, Y., Zhao, Y., Li, C., Zhang, L., Huang, H., Li, Y., Liu, L., Hou, P., Cui, T., Tan, P., Hu, Y., Zhang, T., Huang, Y., Li, X., Yu, J., and Wang, D. (2017). RAID v2.0: an updated resource of RNA-associated interactions across organisms. Nucleic Acids Res., 45(Database):D115.

Ying, S.-Y., Chang, C. P., and Lin, S.-L. (2010). Intron-Mediated RNA Interference, Intronic MicroRNAs, and Applications. In RNA Therapeutics, pages 203–235. Humana Press.

Yu, H., Qi, Y., and Ding, Y. (2022). Deep Learning in RNA Structure Studies. Front. Mol. Biosci., 0.

Zambrano, R. A. I., Hernandez-Perez, C., and Takahashi, M. K. (2022). RNA Structure Prediction, Analysis, and Design: An Introduction to Web-Based Tools. In Riboregulator Design and Analysis, pages 253–269. Humana, New York, NY, New York, NY, USA.

Zarringhalam, K., Meyer, M. M., Dotu, I., Chuang, J. H., and Clote, P. (2012). Integrating Chemical Footprinting Data into RNA Secondary Structure Prediction. PLoS One, 7(10):e45160.

Zeng, C., Fukunaga, T., and Hamada, M. (2018). Identification and analysis of ribosome-associated lncRNAs using ribosome profiling data. BMC Genomics, 19(1):1–14.

Zhang, J., Kobert, K., Flouri, T., and Stamatakis, A. (2014). PEAR: a fast and accurate Illumina Paired-End reAd mergeR. Bioinformatics, 30(5):614–620.

Zhang, Z., Bajic, V. B., Yu, J., Cheung, K.-H., and Townsend, J. P. (2011). Data Integration in Bioinformatics: Current Efforts and Challenges. In Bioinformatics - Trends and Methodologies. IntechOpen, Croatia.

Zhu, X., Gerstein, M., and Snyder, M. (2007). Getting connected: analysis and principles of biological networks. Genes Dev., 21(9):1010–1024.

Zuker, M. (1989). use of dynamic programming algorithms in RNA secondary structure prediction. Mathematical methods for DNA sequences / editor, Michael S. Waterman.

Zuker, M. (1994). Prediction of RNA Secondary Structure by Energy Minimization. In Computer Analysis of Sequence Data, pages 267–294. Springer, New York, NY, USA.

Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Res., 31(13):3406–3415.

Zuker, M., Mathews, D. H., and Turner, D. H. (1999). Algorithms and Thermodynamics for RNA Secondary Structure Prediction: A Practical Guide. In RNA Biochemistry and Biotechnology, pages 11–43. Springer, Dordrecht, The Netherlands.

Zyprych-Walczak, J., Szabelska, A., Handschuh, L., Górczak, K., Klamecka, K., Figlerowicz, M., and Siatkowski, I. (2015). The Impact of Normalization Methods on RNA-Seq Data Analysis. Biomed Res. Int., 2015.

Appendix A

A.1 Pre-processing

Tools for RNA-seq preprocessing were assessed using yeast DDD dataset from SPLASH (Aw et al., 2016). The corresponding sequencing reads were kindly provided by the authors. Paired-end read were merged and subjected to the different preprocessing tools and subsequently trimmed for the following listed adapters.

TruSeq2 Adapters

```
>TruSeq_IndexedAdapter
GATCGGAAGAGCACACGTCTGAACTCCAGTCAC
>TruSeq_smallRNAAdapter
TGGAATTCTCGGGTGCCAAGG
```

Tolerant settings

Trimmomatic

```
trimmomatic SE \
ILLUMINACLIP:../TruSeq2-SE.fa:2:30:10 \
MINLEN:15 \
AVGQUAL:20\
SLIDINGWINDOW:4:20
```

CUTADAPT

```
cutadapt \
-a file:TruSeq2-SE.fa \
-m 15 -q 20 -j 30 -e 2
```

FLEXBAR

```
flexbar \
-a file:TruSeq2-SE.fa \
-qt 20 -qw 4 -m 15 -ae 0.2
```

Strict settings

Trimmomatic

```
trimmomatic SE \
ILLUMINACLIP:../TruSeq2-SE.fa:0:30:10 \
MINLEN:15 \
AVGQUAL:20\
SLIDINGWINDOW:4:20
```

CUTADAPT

```
cutadapt \
-a file:TruSeq2-SE.fa \
-m 15 -q 20 -j 30 -e 0
```

FLEXBAR

```
flexbar \
-a file:TruSeq2-SE.fa \
-qt 20 -qw 4 -m 15 -ae 0.0
```

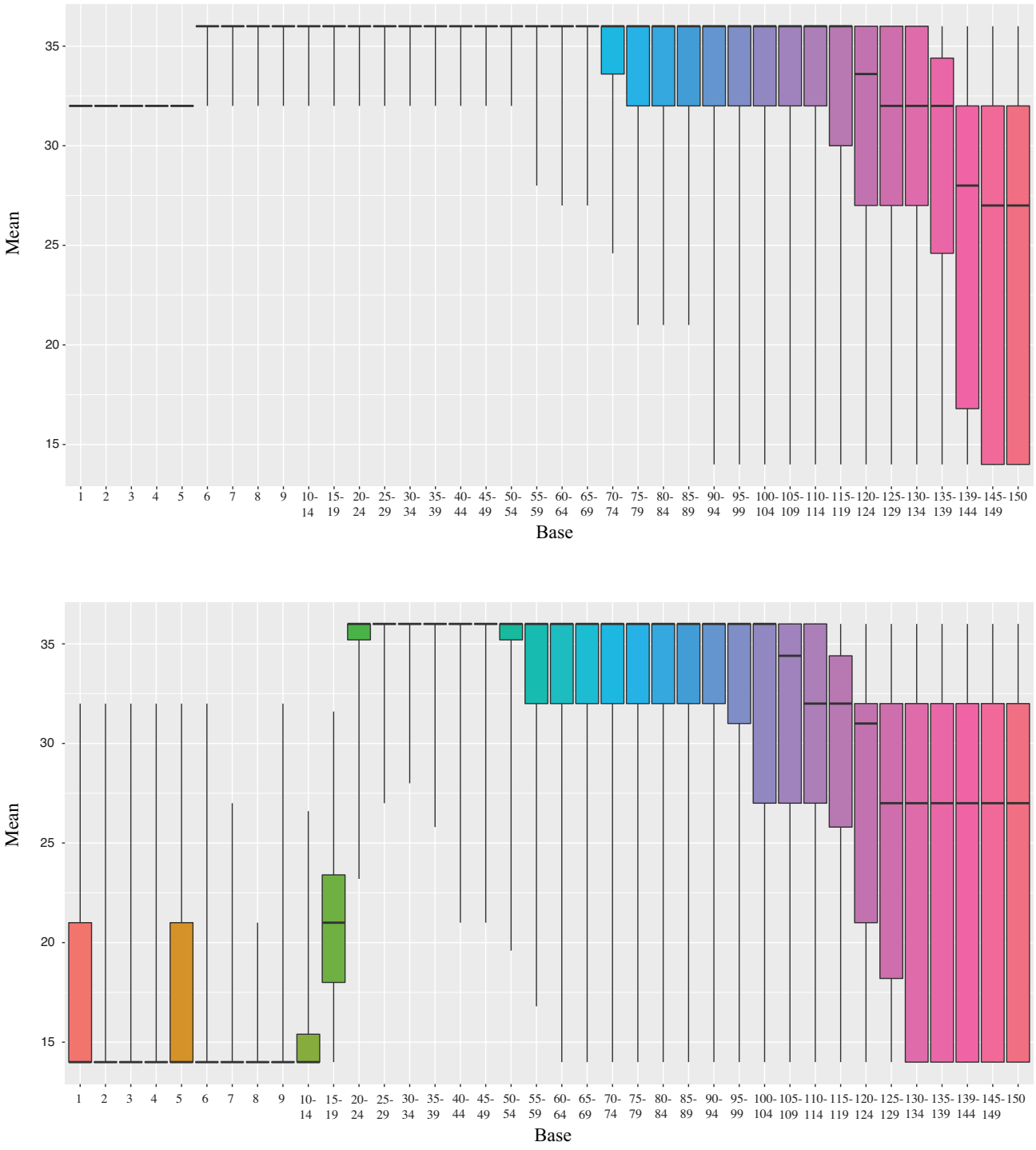

Fig. A.1 Quality scores across all bases for the SPLASH yeast dataset that has been used for benchmarking. In doing so, the sequencing reads in quadruplets were concatenated and quality checked using FASTQC v.0.11.8

A.2 Reference RNA secondary structures

#RNAs	RNA type
725	tmRNA
723	16S rRNA
707	tRNA
470	Ribonuclease P RNA
450	Synthetic RNA
394	Signal Recognition Particle RNA
205	23S Ribosomal RNA
161	5S Ribosomal RNA
152	Group I Intron
146	Hammerhead Ribozyme
64	Other Ribosomal RNA
53	Other Ribozyme
42	Group II Intron
41	Group II Intron

Table A.1 Most common RNA types in RNA Strand v2.0 (Andronescu et al., 2008b)

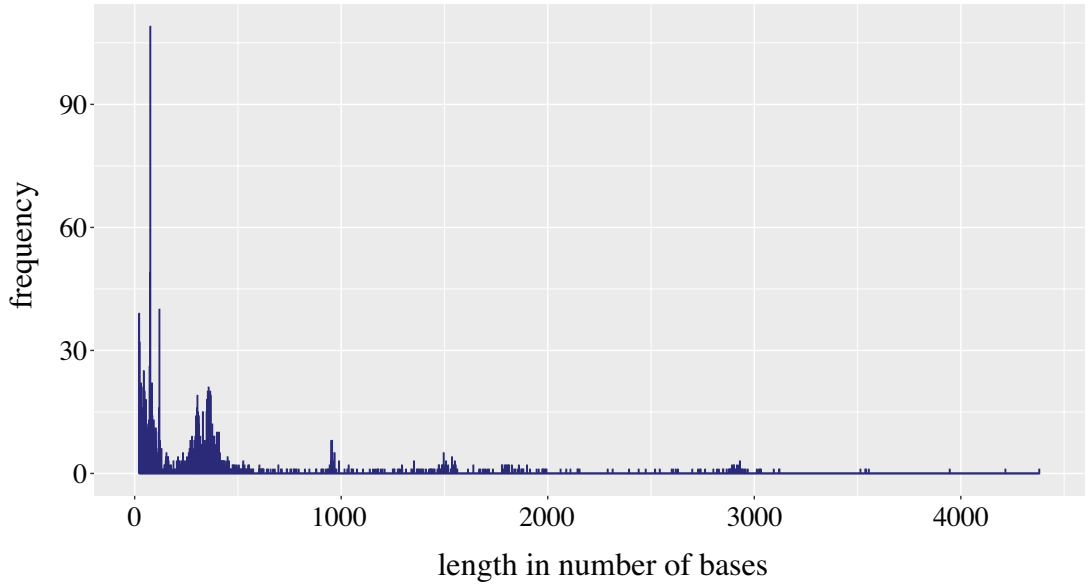

Fig. A.2 Length distribution of RNAs in `CompaRNA` (Puton et al., 2013)

A.3 Statistics of the used datasets

library	raw reads	pre-processed	aligned	annotated (ncRNAs)
LB #001	29,826,236	29,387,027	27,618,714	23,706,771 (1,895,379)
LB #002	27,462,841	26,949,408	23,134,276	20,173,275 (950,698)
LB #003	26,203,021	25,801,073	22,764,098	19,665,069 (1,635,395)
LB+0.5% α MG #001	25,797,475	25,414,294	22,308,033	19,585,788 (1,324,431)
LB+0.5% α MG #002	24,247,745	23,496,678	21,941,302	19,246,785 (1,166,224)
LB+0.5% α MG #003	27,733,019	27,079,092	23,468,567	20,368,452 (1,533,085)
MOPS Rich #001	25,798,907	25,299,315	24,673,951	21,767,243 (1,522,074)
MOPS Rich #002	78,069,457	75,497,738	56,192,479	49,761,334 (5,143,217)
MOPS Rich #003	22,238,206	21,420,428	15,055,674	13,299,074 (1,411,699)
MOPS Rich +0.5% αMG #001	30,378,681	29,930,294	29,260,984	25,662,034 (2,137,021)
MOPS Rich +0.5% α MG #002	18,789,280	17,950,964	12,335,637	11,081,417 (1,548,588)
MOPS Rich +0.5% α MG #003	17,033,107	16,057,047	9,485,174	8,555,399 (1,219,040)
MOPS Minimal #001	41,232,420	31,128,837	26,740,274	23,335,873 (2,534,102)
MOPS Minimal #002	28,703,454	23,174,521	17,615,864	15,440,363 (1,900,376)
MOPS Minimal #003	32,533,308	27,696,171	24,066,868	21,056,840 (1,684,045)
MOPS Minimal +0.5% αMG #001	41,303,284	39,563,872	33,381,765	28,538,915 (3,478,780)
MOPS Minimal +0.5% α MG #002	43,455,761	41,565,726	28,630,167	25355946 (3,807,905)
MOPS Minimal +0.5% α MG #003	26,116,589	24,055,582	18,469,081	16,000,974 (2,263,383)
MOPS Minimal #001	34,085,732	33,154,526	30,301,299	26,597,613 (3,667,483)
MOPS Minimal #002	32,646,666	30,959,073	27,522,438	24,677,355 (4,120,277)
MOPS Minimal #003	32,765,674	31,508,744	29,234,164	25,924,770 (3,889,970)
MOPS Minimal +0.5% 2DG #001	43,739,917	42,107,216	39,815,623	34,648,506 (3,307,074)
MOPS Minimal +0.5% 2DG #002	31,866,797	31,248,224	29,421,389	25,965,296 (3,171,511)
MOPS Minimal +0.5% 2DG #003	35,517,710	34,167,852	31,581,788	27,732,650 (4,035,225)
	777'545'287	734,613,702 $\sim$ 94%	625,019,609 $\sim$ 80%	548,147,742 $\sim$ 70.5% (59,346,982 $\sim$ 7.6%)

Table A.2 Read statistics of the datasets used in the RNA-RNA interaction prediction (Seo et al., 2014)

library	raw reads	preprocessed (RNAnue)	aligned (original)	aligned (RNAnue)	split reads (original)	split reads (RNAnue)
AMTLig #001	74,674,228	63,474,426	56,783,161	57,605,996	4,164,451	23,911,038
AMTLig #002	86,793,027	68,445,406	65,888,979	60,968,861	4,263,369	22,779,875
AMTnoLig #001	74,416,665	65,348,021	59,822,709	60,542,139	369,470	10,648,213
AMTnoLig #002	91,605,615	64,289,079	68,285,451	60,175,129	430,006	8,244,939
noAMTLig #001	63,587,469	57,139,231	43,888,457	52,781,013,	4,368,620	21,341,868
noAMTLig #002	195,508,413	160,128,777	133,722,918	150,454,247	6,881,064	44,652,426
noAMTnoLig #001	68,352,734	58,916,386	50,204,443	53,682,666	649,939	9,092,502
noAMTnoLig #002	161,551,979	141,066,633	118,712,918	126,601,111	1,621,126	19,668,903
total	816,490,130	678,807,959	597,279,036	622,811,162	22,748,045	160,339,764

Table A.3 Statistics of the human LIGR-seq datasets

library	original reads	preprocessed (RNAnue)	aligned (original)	aligned (RNAnue)	split reads (original)	split reads (RNANue)
GM12892 Total RNA #001	53,747,987	51,854,852	53,308,466	49,693,932	1,467,951	1,022,343
GM12892 Total RNA #002	29,859,136	28,361,542	29,864,687	25,961,754	1,288,867	9,904,119
GM12892 Total RNA #003	45,400,893	44,077,187	45,665,506	42,605,214	1,331,665	1,059,066
GM12892 Total RNA #004	60,332,939	57,989,628	59,564,405	55,772,718	1,080,378	875,963
GM12892 PolyA #001	183,913,864	176,490,908	175,802,188	176,490,912	6,576,177	26,932,377
GM12892 PolyA #002	115,234,808	112,253,309	110,845,899	108,120,799	5,413,415	20,804,850
GM12892 PolyA #003	3,963,881	3,856,675	3,813,341	3,700,022	245,320	756,564
GM12892 PolyA #004	53,371,012	53,091,920	50,588,519	65,112,315	2,585,258	3,451,211
GM12892 snoRNA #001	149,071,830	143,853,382	148,386,187	137,542,822	1,964,934	1,761,992
total	694,896,350	671,829,403	677,839,198	665,000,488	21,953,965	66,568,485

Table A.4 Statistics of the human SPLASH data.

library	original reads	preprocessed (original)	preprocessed (RNAnue)	aligned (original)	aligned (RNAnue)	split reads (original)	split reads (RNAnue)
HEK293T #001	47,730,545	23,926,978	22,761,483	7,469,186	11,707,076	1,871,076	11,646,703
HEK293T #002	52,440,215	26,453,476	24,669,852	8,300,327	13,775,540	2,247,017	13,735,745
HEK293T #003	183,151,141	65,768,587	62,114,664	18,625,763	30,643,409	5,497,231	30,488,247
total	283,321,901	116,149,041	109,545,999	34,395,276	56,126,173	9,165,324	55,870,695

Table A.5 Statistics of the human PARIS datasets.

A.4 Execution of external pipelines

`Aligater` **(LIGR-Seq)**

`Aligater` can only align reads to the transcriptome and requires a specific format for the transcriptome headers. For that reason the transcriptomes provided by the authors at https://github.com/timbitz/Aligater were used and unified with the latest release of miRBase (v22) and miRTarbase (v7). The single-end LIGR-Seq reads were aligned using `Aligater align` with default parameters. The resulting alignments were then subjected to `Aligater detect` to retrieve the split reads. Here, the chaining penalty was set to -24. Following this, the chimeric reads were filtered by applying `Aligater post` using the `BLAST` database. In additon, the split reads were reclassified using 'Aligater reclass' and finally a statistical analysis was performed on the reads by calling `Aligater stats`. Interactions with a p-value below 0.05 were retained.

SPLASH

For SPLASH, reads were aligned to the transcriptome provided at https://csb5.github.io/splash/ using `Bwa-mem`. The transcriptome was merged with sequences from the latest release of miRBase (v22) and miRTarBase (v7). The minimum score was set to 20 (option -T). Finally, reads were deduplicated using `samtools rmdup`. Afterwards, the custom script 'find_chimeras.py' was used to detect the split reads that were filtered using `filter_chimeras.py`. In contrast to the original paper, we also kept chimeric reads whose segments are 50 bases or less apart. We took the surviving chimeric reads and extracted the corresponding alignments and remapped these against the genome (GRCh38.p13) using `STAR` with the following parameters:

```
STAR \
--readFilesType BAM SE \
--runMode inputAlignmentsFromBAM \
--readFilesCommand samtools view \
--outWigType bedGraph \
--twopassMode Basic \
--alignSplicedMateMapLminOverLmate 0.1 \
--outSJfilterOverhangMin 10 6 6 6 \
--outSJfilterCountUniqueMin 6 1 1 1 \
--outSJfilterCountTotalMin 6 1 1 1 \
--outSJfilterDistToOtherSJmin 5 0 5 0 \
```

```
--winAnchorMultimapNmax 9000 \
--seedPerWindowNmax 1000 \
--outSAMstrandField None \
--outSAMmultNmax 1 \
--outMultimapperOrder Random \
--outSAMattributes All \
--outSAMprimaryFlag AllBestScore \
--outFilterMultimapScoreRange 0 \
--outFilterMultimapNmax 9000 \
--outFilterMismatchNmax 2 \
--outFilterIntronMotifs None \
--outFilterMatchNminOverLread 0.1 \
--outFilterScoreMinOverLread 0.1 \
--alignEndsType Local
```

`pickJunctionReads.awk` was then used to remove split reads that entirely span annotated junctions. This was done using the annotations from ENCODE (GRCh38.p13).

PARIS

In the PARIS workflow `readCollapse.pl`, which was provided by the authors at https://github.com/qczhang/icSHAPE/, was used to remove PCR duplicates. Afterwards, `trimmomatic` with default parameter settings was used to remove adapter sequences. The reads were aligned with `STAR` v.2.7.5a, allowing at most 100 different positions to map to (option outFilterMultimapNmax) and all genomic gaps ($\geq$ 1nt) were considered as introns (option alignIntronMin). Penalties for non-canonical junctions and AT/AC, GT/AT junctions were decreased to -4 (options `scoreGapNoncan` and `scoreGapATAC`). In the chimeric read alignment, the segment length had to exceed 18nt (option `chimSegmentMin`) and entirely span the chimeric junction (option chimJunctionOverhangMin). The accompanying script `samPairingCalling.pl` taken from https://github.com/qczhang/paris/ groups them into duplex groups and annotates them using `annotateTrans.test.pl`. Finally, `filterRG.pl` filters read groups with minimal support.