Statistical models to capture protein-RNA interaction footprints from truncation-based CLIP-seq data

Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) vorgelegt von

Sabrina Krakau

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Berlin, 2019

Erstgutachterin:
Prof. Dr. Annalisa Marsico, Freie Universität Berlin, Deutschland

Zweitgutachter:
Prof. Dr. Rolf Backofen, Albert-Ludwigs-Universität Freiburg, Deutschland

Datum der Disputation: 01.11.2019

Abstract

Protein-RNA interactions play an important role in all post-transcriptional regulatory processes. High throughput detection of protein-RNA interactions has been facilitated by the emerging CLIP-seq (crosslinking and immunoprecipitation combined with high-throughput sequencing) techniques. Enrichments in mapped reads as well as base transitions or deletions at crosslink sites can be used to infer binding regions. Single-nucleotide resolution techniques (iCLIP and eCLIP) have been achieved by capturing high fractions of cDNAs which are truncated at protein-RNA crosslink sites. Increasing numbers of datasets and derivatives of these protocols have been published in recent years, requiring tailored computational analyses. Existing methods unfortunately do not explicitly model the specifics of truncation patterns and possible biases caused by background binding or crosslinking sequence preferences.

We present PureCLIP, a hidden Markov model based approach, which simultaneously performs peak calling and individual crosslink site detection. It is capable of incorporating external data to correct for non-specific background signals and, for the first time, for the crosslinking biases. We devised a comprehensive evaluation based on three strategies. Firstly, we developed a workflow to simulate iCLIP data, which starts from real RNA-seq data and known binding regions and then mimics the experimental steps of the iCLIP protocol, including the generation of background signals. Secondly, we used experimental iCLIP and eCLIP datasets, using the proteins' known predominant binding regions. And thirdly, we assessed the agreement of called sites between replicates, assuming target-specific signals are reproducible between replicates.

On both simulated and real data, PureCLIP is consistently more precise in calling crosslink sites than other state-of-the-art methods. In particular when incorporating input control data and crosslink associated motifs (CL-motifs) PureCLIP is up to 13% more precise than other methods and we show that it has an up to 20% higher agreement across replicates. Moreover, our method can optionally merge called crosslink sites to binding regions based on their distance and we show that the resulting regions reflect the known binding regions with high-resolution.

Additionally, we demonstrate that our method achieves a high precision robustly over a range of different settings and performs well for proteins with different binding characteristics. Lastly, we extended the method to include individual CLIP replicates and show that this can boost the precision even further. PureCLIP and its documentation are publicly available at `https://github.com/skrakau/PureCLIP`.

Preface

Publication and contributions

This thesis is build on a project whose ideas grew out of many fruitful discussions with Annalisa Marsico (MPI for Molecular Genetics), and with Hugues Richard (Sorbonne Universités, UPMC) who later contributed to the model design. The resulting method was published in *Genome Biology* (Krakau, Richard, and Marsico, 2017 [79]) and is available at `https://doi.org/10.1186/s13059-017-1364-2`. A number of additional evaluations and improvements of the method were performed and implemented since then, including an extension to incorporate replicates.

In accordance with the standard scientific protocol, throughout this thesis I will use the personal pronoun *we* to indicate the reader and the writer, or my scientific collaborators and myself.

Acknowledgements

First and foremost, I would like to thank my supervisor Annalisa Marsico for her advice and support during my PhD, and for the trust and freedom she granted me to develop my ideas. I also want to thank Hugues Richard for his advise and many valuable comments. I greatly appreciate the many lively and fruitful discussions with Annalisa and Hugues, which allowed the project to grow.

I further wish to thank the IMPRS coordinators Kirsten Kelleher and Fabian Feutlinske for their assistance along the way towards the PhD and for organizing great retreats and summer schools. My thanks also go to my office mates and colleagues of the RNA Bioinformatics group, Stefan Budach, Lisa Barros de Andrade e Sousa, Roman Schulte-Sasse, Brian Caffrey and Evgenia Ntini, for many interesting scientific and non-scientific conversations and discussions.

I would like to thank Leon Kuchenbecker, Brian Caffrey and Lena Moser for proofreading and commenting on the thesis manuscript.

I wish to thank my parents, my brother Freddy and Natasha for their continuous support. Finally, my biggest gratitude goes to Leon Kuchenbecker, who had to listen many times to emerging thoughts about hidden Markov models and who continuously supported me during the last years, in particular during the time of writing this thesis.

Contents

Part I.

Introduction and Preliminaries

1. Introduction

In molecular biology, the primary goal is to decipher the roles and interactions of molecules that drive cellular functions, the key players being DNAs, RNAs and proteins. This includes regulatory mechanisms used by cells to implement their specific gene expression levels, defining their metabolic states and functions, and that allow them to differentiate into tissue specific lineages or developmental stages.

Among the molecular interactions, the interactions between proteins and RNAs play an important role as they regulate a large number of metabolic processes within the cell. To be able to investigate such processes, we need to be able to analyse the exact landscape of interactions, i.e. identify the exact binding regions, optimally in a quantitative fashion. Since 2003, CLIP-seq technologies [140] allow the in vivo, transcriptome-wide detection of binding regions for a protein of interest with high resolution. Many advances optimizing different experimental steps were made over the last years, improving the sensitivity and resolution.

The various protein-RNA binding mechanisms are still largely unexplored. The primary scope of this thesis is the development of novel computational methods that allow for an accurate analysis of CLIP-seq data at a high resolution. With this, we aim to gain insights into the underlying binding mechanisms as well as the experiment-specific signal characteristics and biases. Moreover, with our work we want to empower other researchers to draw more reliable conclusions from CLIP experiments, to understand the mechanisms of the different regulatory processes and finally, to explore disease-causing disruptions or alterations of protein-RNA interactions.

1.1. Thesis outline

In this thesis we describe our work ranging from a statistical model to capture interaction footprints, to a truncation-based CLIP data simulation workflow and a comprehensive evaluation strategy including experimental data. The thesis is divided into three parts.

The first part comprises four chapters, where we will first give an introduction into the general biological background and protein-RNA interactions (Chapter 2). In Chapter 3 we will then explain the available experimental technologies used to capture protein-RNA interactions in detail, describe the computational challenges concerning the analysis and give an overview of existing methods. Furthermore, in Chapter 4 we will describe the fundamental statistical concepts that are relevant in the context of the methods presented in the subsequent chapters.

In the second part will focus on developed methods, evaluation strategies and results.

In Chapter 5 we will present our method to capture protein-RNA interactions. For that we will first describe in detail how we model interaction footprints, detecting both regions enriched in pulled-down RNA fragments and individual crosslink sites, using a hidden Markov model (HMM). Additionally, we will present a non-homogeneous version of this HMM that can include covariates to correct for various biases. Lastly, we will present an extended version to include replicates. In Chapter 6 we will then describe the experimental and simulated data used for evaluating the method's performance in comparison to other state-of-the-art methods. For the latter, we developed a simulation framework mimicking the experimental steps of the iCLIP protocol. In Chapter 7 we will present some intermediate results from the model training for an example eCLIP dataset. In Chapter 8 we will then proceed to present the main evaluation, where we compare the precision of our method to that of other methods, both at the crosslink site and at the binding region level. For this purpose, we use 1) simulated data, 2) experimental iCLIP and eCLIP data for proteins with known binding regions and 3) we assess the agreement of called sites between replicates. In Chapter 9 we will complement these evaluations with additional assessments, i.e. we will show the memory and runtime requirements, the performance for different settings and when including RNA-seq data instead of input control data. We will further show results for a protein with binding characteristics different from those of the proteins used before. Lastly, we present the performance gain of our method when including replicates.

In the third and last part we will then discuss the advantages and disadvantages of our model in comparison to other strategies and review a few general insights concerning CLIP data. We conclude the thesis with an outlook, where we will describe potential future improvements and applications of our method.

2. Biological background

In this chapter we will give an introduction into the field of molecular genetics, discuss in more detail the role of protein-RNA interactions and provide a brief overview of the experimental techniques that are most important in this context.

2.1. Introduction to genetics

For living organisms the instructions for their development, reproduction and all biological functions are stored in *deoxyribonucleic acid (DNA)* molecules. This information, which makes each species and each individual unique, is passed from one generation to the next. DNA is a long biopolymer consisting of nucleotides, each containing one of the four bases adenine (A), cytosine (C), guanine (G) and thymine (T). The nucleotides are covalently bound via their phosphate and deoxyribose groups, which build the DNA backbone. Since the phosphate group is always bound to the third carbon atom of one deoxyribose and the fifth of the next, each strand has a direction and its ends are denoted as the 3' and 5' end accordingly. The sequence of the nucleotides encodes the genetic information. Typically, two strands form a double stranded DNA helix, where As always pair with Ts and Cs with Gs via hydrogen bonds, as discovered by Watson & Crick in 1953 [148]. As a result, one strand is the exact reverse complement of the other. This enables DNA replication prior cell division, where the strands are separated and serve as templates for the synthesis of new reverse complement strands.

In eukaryotes, the genetic information is located in the cell's nucleus and organized in multiple molecules called *chromosomes*. The human genome contains 23 pairs of chromosomes, each set comprising about 3 billion base pairs and about 40,000 genes [111, 151], i.e. regions encoding proteins or other regulatory nucleic acids. Generally, within one individual each cell contains the same genetic information with the relatively rare exception of somatic variation, i.e. substitutions, deletions or insertions which can originate, for example, from errors during the DNA replication process. Another exception are certain immune cells, T- and B-cells, which acquire somatic variation through a targeted recombination process as well as gametes which contain only a single copy of each chromosome.

To apply the genetic instructions, the cells read out the information and use it to regulate the synthesis of proteins. A key molecule in this process is the *ribonucleic acid (RNA)*. Information is transferred from DNA to RNA and from RNA to protein molecules. This flow of information is part of what became known as the *central dogma of molecular biology*, shown schematically in Figure 2.1, which most importantly states that information cannot be transferred back from protein molecules to nucleic acids.

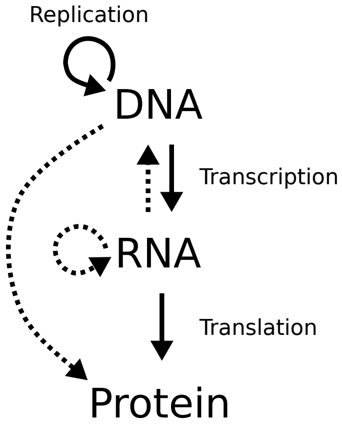


Figure 2.1.: The central dogma of molecular biology as (re-)stated by Francis Crick in 1970 [34]. Solid arrows indicate the general transfer of sequence information. Dashed arrows indicate special cases of transfer, for example reverse transcription (RNA → DNA) or RNA replication (RNA → RNA). A transfer from DNA directly to protein is theoretically possible, but has not been observed in living cells [142].

The shown version of the dogma, in contrast to the version stated by James Watson in 1965 [74], also allows the transfer from RNA to DNA as a special case and is valid until today.

2.1.1. Transcription and regulation

The process which transfers the genetic sequence information from DNA to RNA is called transcription and carried out by a protein complex called *RNA polymerase*. The RNA polymerase moves along the DNA and uses one strand as a template to synthesize single-stranded RNA in 5' → 3' direction (see Figure 2.2). Thus the resulting RNA encodes the exact same information as the corresponding DNA region.

Although the genetic information is identical in each cell of one organism, the cellular functions differ significantly between different points in time, different tissues and under different external conditions, for example stress. This is enabled by complex regulatory processes, both at a transcriptional and at a post-transcriptional layer.

At the transcriptional level, for example, specific proteins called *transcription factors* (TFs) can bind to specific regions in the DNA and thereby promote or repress the activity of the RNA polymerase. Additionally, distal genomic elements called *enhancers* can interact with the promoter of genes via genomic loops and increase transcription. Another layer of regulation is added by so called *epigenetic* modifications, which influence gene expression without altering the underlying DNA sequence [5, 15]. The most important epigenetic mechanisms are DNA methylations and modifications at histones, proteins important for organizing the DNA inside the nucleus. Epigenetic modifications can be inherited during cell divisions and to the next generation, but are

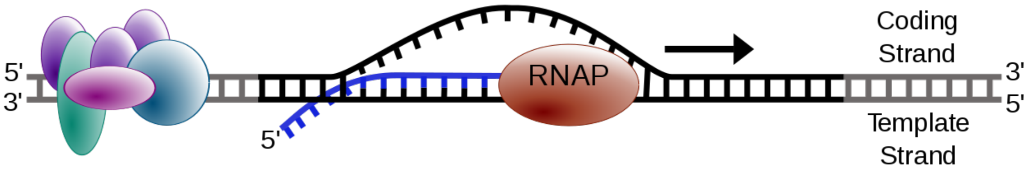


Figure 2.2.: Transcription The RNA polymerase (RNAP) synthesizes RNA in 5' → 3' direction based on a DNA template. From Wikimedia Commons (Public Domain).

far less stable than the genetic information [5].

2.2. RNA

RNA molecules are the key player in the transfer of genetic information and its regulation. Like DNA, RNA is a biopolymer consisting of nucleotides, but instead of deoxyribose RNA contains ribose and instead of the nucleotide base thymine (T) it contains uridine (U). Moreover, in contrast to DNA, RNA is a single-stranded molecule, often exhibiting intramolecular base pairs. RNAs that are translated into proteins are called *messenger RNAs* (mRNAs). Besides mRNA, there exist *non-coding RNAs* (ncRNAs) that do not encode proteins and fulfill diverse functions. We will give an introduction into the diverse world of RNAs in the following sections.

2.2.1. Post-transcriptional modifications

Before being translated, nascent mRNAs called pre-mRNAs undergo several processing and modification steps, of which some already take place co-transcriptionally. First, pre-mRNAs are processed into mature mRNAs through splicing, 5' capping and 3'end polyadenylation. RNA splicing is the process where certain regions, called *introns*, are cleaved out and the remaining regions, called *exons*, are ligated. Alternative splicing events, i.e. the use of different splice sites or different exon combinations, generate different mRNA isoforms from one single gene. From an evolutionary viewpoint this mechanism is an efficient way to increase the number of resulting proteins without equivalently increasing the size of the genome. The entire collection of present transcripts in a cell or given cell population is called the *transcriptome*.

In addition to the classical post-processing events such as splicing, the individual RNA nucleotides can be further chemically modified by specific enzymes. Altogether, more than 100 different modification types were already identified [20], although many of them are rare. The most frequent modifications are different types of RNA methylations, A-to-I RNA editing, where adenines are deaminated to inosines (Is), and pseudouridylations, where uridines are isomerized to pseudouridines (Ψs) . Such post-transcriptional modifications can either directly alter the resulting protein sequence or impact regulation, for example, by altering RNA structure or protein binding sites. Together these modifications build the *epitranscriptome*, which increases not only the complexity of the transcriptome, but also allow for dynamic changes between different developmental stages or under different environmental conditions.

ncRNAs are processed and modified as well, however, exhibiting their own specific biogenesis pathways. Polyadenylated mRNAs are finally exported to the cytoplasm, where the sequence information is translated into proteins.

2.2.2. Non-coding RNAs

In humans, less than 2% of the genome encodes for proteins [31], while the remaining 98% contains other regulatory elements or genes that are transcribed into ncRNAs. However, a large fraction is thought to be functionless, often referred to as *junk DNA*. Such junk DNA can also be transcribed into *junk RNAs*, which is thought of as transcriptional noise [77]. In 2012 it has been estimated by the ENCODE Consortium that more than 80% of the human genome has some sort of biochemical function [31]. This statement has caused an ongoing controversial debate, mainly about the definition of “function” and “junk” [51], where some counter that the functional fraction has an upper limit of 25% [56, 57]. Similarly, the distinction between junk RNA and functional ncRNAs remains challenging [110]. Although the number of known ncRNAs has increased constantly over the past years, the exact number of those that are also functional is difficult to estimate and remains to be determined [110, 113].

A large number of different functional ncRNA classes is known, ranging from ribosomal RNAs (rRNAs) and transfer RNAs (tRNAs) that are involved in the translation of mRNAs into proteins, to micro RNAs (miRNAs), small nuclear RNAs (snRNA), long ncRNAs (lncRNAs) and circular RNAs (circRNAs), each coming with their own functional roles and predominant cellular localizations. Similar to mRNAs, ncRNAs are processed and modified, however, undergoing distinct regulatory principles. ncRNAs are involved in many different cellular processes, play important roles in gene regulation and, consequently, are associated with many different diseases, including cancer as well as developmental and neurological disorders [45, 137]. For a detailed review about different classes of ncRNAs and their pathways see, for instance, [23, 45].

2.2.3. RNA structure

During transcription, RNA is synthesized as a single-stranded molecule which allows for intramolecular base pairing between complementary parts of the sequence forming secondary structures. Even though this secondary structure is more or less predictable based on complementary subsequences [46, 87, 118, 160], for many RNAs the structure is dynamically changing [27]. Importantly, it not only depends on the sequence itself, but also on potential chemical modifications as well as interactions with proteins, DNAs and other RNAs [46].

2.2.4. Post-transcriptional regulation

Post-transcriptional regulation is carried out at a multitude of different levels. Broadly speaking via different regulatory processes such as RNA splicing, modification, transport, export, translation and degradation, each modulated by protein-RNA and potentially RNA-RNA interactions. The former will be discussed in detail in the next section.

The latter includes interactions with a wide range of ncRNAs, which in turn undergo regulatory processes. A well studied example are miRNAs, which are $\sim$ 22 nt long, bind to mRNAs and thereby regulate their translation [45, 97, 137]. miRNAs can be located within intragenic or intronic regions of other genes and often have their own promoters and thus transcriptional regulation. They undergo complex biogenesis pathways, starting from long primary transcripts (pri-miRNAs) that contain hairpin structures and are cleaved by Drosha and Dicer, finally resulting in short miRNAs. This biogenesis in turn is regulated by other ncRNAs and RNA binding proteins. Mature miRNAs are then loaded into the RNA-induced silencing complex (RISC) and bind to complementary regions within the 3'-UTR of their target mRNAs, where they inhibit translation and cause mRNA decay [137]. Although many different regulatory processes were revealed in recent years, the vast majority of such processes is still poorly understood and remains to be explored.

2.3. Protein-RNA interactions

Interactions between RNAs and RNA binding proteins (RBPs) play an essential role in both transcriptional and post-transcriptional gene regulation. RBPs bind on several sites of coding and non-coding RNAs and regulate RNA functions via metabolic processes as described in the previous section. They usually fulfill many diverse functions and many of them bind to DNA as well. Together, RNAs and RBPs form *ribonucleoprotein (RNP)* complexes. Beside RBPs acting on RNAs, RNAs can also act on proteins and influence their function. For example, lncRNAs can serve as guides to recruit transcription factors, as scaffolds to create large RNPs, as decoys for RBPs or as signals, e.g. by inducing conformational changes of signaling molecules [50, 108]. RNPs are involved in many different important cellular processes. Well known examples of larger RNP complexes are the spliceosome, which is responsible for splicing the pre-mRNA, and the ribosome, responsible for the translation of mRNAs into proteins. Consequently, disruptions of protein-RNA interactions, by mutations either of the RBP or the bound RNA, can cause severe diseases ranging from cancer and auto-immune defects to neurological disorders [22, 93].

Conventional RBPs contain one or more well-defined *RNA binding domains* (RBDs), such as RNA recognition motifs (RRMs), heterogeneous nuclear RNP K-homology domains (KHs) and zinc fingers (ZFs) [94]. Most of these RBDs bind short, 3 to 5 nt long regions of the RNA [8, 40]. In humans, more than 1,500 different proteins were predicted to be RNA binding based on canonical RBDs and existing literature [52]. However, recent years revealed that proteins can bind RNAs also without featuring such classical domains, for example through intrinsically disordered regions (IDRs) [21], shape complementary regions or RNA deposition through protein-protein interactions (for a detailed review see Hentze et. al [66]). In particular large RNP complexes, such as the ribosome or spliceosome, often lack canonical RBDs [12, 112]. Moreover, large-scale in vivo *RNA interactome capture* (RIC) approaches, making use of quantitative mass spectrometry to identify proteins bound to RNAs, detected more than 1000 different human RBPs [11, 21]. Interestingly, only half of these proteins contain classical RBDs and a large fraction was not known to be RNA related before [66]. This illustrates that a large number of RBPs is still likely to be discovered and their functions to be explored.

When shifting the focus from the involved proteins or protein domains to the RNAs, one can ask for the exact binding targets for a protein of interest. This can provide valuable information about the protein's role in regulatory processes and the underlying binding mechanisms. It is known that the binding affinities of RBPs depend on both the RNA sequence and structure, while some RBPs have a stronger preference for either of the two [40, 65]. In vitro and in vivo experimental strategies have been developed to identify and characterize the target RNAs of RBPs. In vitro techniques enable the study of binding affinities while eliminating other regulatory factors. In protocols such as SELEX [138], RNAcompete [116] and Bind-N-Seq [82], the target RNAs are identified by exposing an RBP to a large pool of short, random RNAs and subsequently measuring the bound RNAs. Bind-N-Seq, the most recent one, uses high throughput sequencing. Based on this method, a recent study [40] examined the RNA binding affinities for 78 human RBPs. The results showed that a large number of proteins bind to a relatively small set of short, low-complex sequence motifs, while obtaining RNA specificity through preferences for different structures or flanking sequences. Many RBPs were shown to bind to specific bipartite motifs, reflecting the binding of multiple RBDs [2, 40]. Furthermore, the authors speculate that long stretches of mono- or dinucleotides within transcripts might facilitate sliding of certain RBPs along the RNA [40]. As this study mainly focused on proteins containing classical RBDs, the binding affinities of newly detected RBPs with novel types of RBDs remain to be examined.

In living cells, protein-RNA interactions do not solely depend on the sequence-structure binding preferences, but also on numerous other factors. For example, other RBPs can regulate the interactions via cooperative or competitive binding [134]. Moreover, the RNA's structure highly depends on the condition within the cell, e.g. on chemical modifications and on interactions with other biomolecules (see Section 2.2.3). In vivo methods, such as *crosslinking and immunoprecipitation (CLIP)* technologies, enable the detection of protein-RNA interactions as they occur in living cells and will be described in detail in Section 3.2.

Both in vitro and in vivo methods are important to gain new insights into complex and still poorly understood regulatory layers. Many open challenges remain: detecting new RBPs and their RBDs, understanding unknown binding mechanisms and finally deriving information about the RBP's function. Fortunately, more and more high-throughput methods are being developed allowing us to accurately investigate these questions. Deepening our understanding of the detailed regulatory processes will also enable the development of new therapeutics, for example by blocking or enhancing specific protein-RNA binding sites.

2.4. Experimental techniques in molecular biology

In the following we will briefly discuss the main biotechnological methods laying the foundation for research in the field of molecular genetics, as well as for the experimental and computational methods described in this thesis.

Next generation sequencing

A common task in molecular biology is to detect the exact sequence of nucleic acid molecules of interest, for example of the whole genome or of transcribed RNA molecules. The first sequencing technology was introduced by Sanger [122] based on the synthesis of complementary strands by DNA polymerases and the incorporation of chain-terminating nucleotides for one specific base. This results in molecules of different lengths all ending with the corresponding base. The lengths can be detected and then used to recover the sequence. For this purpose, DNA molecules are first fragmented, amplified and then sequenced from one end, generating short stretches of known nucleotides called *reads*. For more than one decade now, several new technologies were developed where fragments are sequenced in a massively parallel manner, allowing a faster analysis with decreasing costs compared to the previously used Sanger technology. These technologies are known as *Next-generation sequencing (NGS)* or *high-throughput sequencing (HTS)*. One of the most widely used technologies is Illumina sequencing [13], which features relatively low error rates and is relevant in the context of this thesis. Standard Illumina platforms support read lengths between 50 and 300 nt, as well as the sequencing of fragments from both ends, called paired-end sequencing. More details and an overview of the current field can be found, for instance, in the review by Heather & Chain [64].

Reverse transcription (RT)

Since sequencing technologies are designed for DNA molecules, when investigating RNA molecules these are first artificially transcribed into *complementary DNA (cDNA)* molecules by using RNA-dependent DNA polymerases, called reverse transcriptases, from retroviruses.

PCR amplification

Polymerase chain reaction (PCR) is a technique to amplify DNA fragments in order to prepare libraries large enough for sequencing. The DNA molecules are exponentially amplified using DNA polymerases and cycles of repeated heating and cooling. First, during the denaturation step heat causes a melting of the double-stranded DNA into single-stranded DNA. Next, lower temperatures allow the annealing of primers to the templates molecules. Such primers are short, synthetic stretches of DNA that are complementary to a target region at the 3' end of the template DNA. Finally, DNA polymerases are used to extend the complementary strand in 5' to 3' direction and the resulting double-strand DNAs serve as templates for the next cycle.

Immunoprecipitation

Immunoprecipitation (IP) is a technique to isolate or *pull-down* a specific protein out of a solution by using an antibody binding specifically to this protein. During this process, the antibody itself is immobilized to a solid support, e.g. agarose or magnetic beads, to allow a subsequent washing and finally the elution of the purified protein.

NGS applications

NGS technologies are applied to an increasingly diverse range of biological questions. In order to study particular cellular processes, technologies capturing only certain molecules of interest were developed. For example, with DNase-seq [132] only open chromatin regions are sequenced. With ChIP-seq [72] only genomic regions that are bound by a certain protein are sequenced. Another example are protocols based on chemical probing such as SHAPE-seq [91], which causes cDNA truncations within unpaired regions of RNAs and can be used to infer structural information. Furthermore, CHART protocols can be used to capture proteins and DNA bound to an RNA of interest [127]. In the context of protein-RNA interactions, RIP-seq [159] and CLIP-seq [140] protocols can be used to detect RNAs bound to a protein of interest. This class of protocols will be described in-depth in the following chapter. Generally, an increasing number of such NGS protocols is being developed, addressing specific biological questions and requiring tailored computational analyses.

3. Capturing protein-RNA interactions

In the previous chapter we have discussed the important role of interactions between proteins and RNAs for regulatory processes. These interactions can be investigated from different perspectives, either by pulling down RNAs with the goal to detect bound proteins or, and this will be the focus in this chapter, by pulling down one protein of interest with the goal to detect the bound RNAs. In order to fully understand regulatory processes mediated by RBPs, it is crucial to accurately determine the exact binding regions for a protein of interest.

In the following we will give an overview of available experimental technologies, with a particular focus on the iCLIP and eCLIP protocols and discuss the characteristics of the resulting data. We will discuss the challenges in interpreting this data, discuss existing analysis methods and motivate a revised model.

3.1. RIP-seq

The first experimental method developed to capture protein-RNA interactions is known as RNA immunoprecipitation (RIP), with an early version already published 1979 [84] and a more systematical approach established in the early 2000s [106, 135, 136]. Native protein-RNA complexes are pulled down by immunoprecipitation (IP) with antibodies specific for the protein of interest. RIP is then combined with RT-PCR, microarray experiments (RIP-chip) [136] or high-throughput sequencing (RIP-seq) [159] to detect the bound RNA transcripts. The protein-RNA interactions are preserved using either optimized washing conditions or by inducing covalent crosslinks at the interaction sites with formaldehyde [106]. Note, that the latter causes crosslinks not only at sites with protein-RNA interactions, but also at protein-protein and protein-DNA interaction sites. As a consequence, both strategies capture also indirectly bound RNAs, e.g. from large ribonucleoprotein complexes [150].

3.2. CLIP-seq

In 2003, a technology using crosslinking and immunoprecipitation combined with high-throughput sequencing (CLIP-seq) [140] was invented, allowing a genome-wide binding site detection. In contrast to RIP, RNAs are fragmented prior to the IP, resulting in a far higher resolution. Further, CLIP methods use UV light, which causes the formation of covalent crosslinks only at sites with direct protein-RNA interactions, but not at protein-protein interaction sites. The crosslinks are relatively strong and allow a stringent washing to remove indirectly bound or sticky RNAs. Note that CLIP protocols are always combined with sequencing, therefore the terms CLIP and CLIP-seq are used synonymously. The most commonly used protocols in this field are HITS-CLIP (CLIP coupled with high throughput sequencing) [85], photoactivatable ribonucleoside-enhanced CLIP (PAR-CLIP) [62], individual-nucleotide CLIP (iCLIP) [78] and enhanced CLIP (eCLIP) [143]. In all protocols the covalent crosslinks subsequently increase the probability for base transitions, deletions and truncations during the reverse transcription, which can serve as *diagnostic events* to localize interaction sites (see Figure 3.1, step (9)).

An important limitation of HITS-CLIP and PAR-CLIP is that due to the ligation of an adapter at the 5' end of the RNA fragments, they only capture cDNAs which are entirely read by the reverse transcriptase, i.e. which are not truncated (see Figure 3.1, left column). For HITS-CLIP it has been shown that deletions, affecting 8-20% of the captured cDNAs, can be used as diagnostic events to infer crosslink sites. In contrast, transitions are more dispersed within the reads and can not be reliably distinguished from SNPs, RNA editing sites or sequencing errors [157]. For PAR-CLIP, cells are additionally treated with UV-reactive nucleoside analogs such as 4-thioridine (4SU) or 6-thioguanosine (6sG), which are incorporated into the RNA and crosslink with higher efficiency. This additionally leads to T-to-C transitions during reverse transcription, which are used as diagnostic events. Although the nucleoside analogs increase the resolution, the results of the experiment depend on a successful incorporation of these analogs and studies showed that these can additionally cause unwanted cellular responses [19].

The fraction of truncated and thus in HITS-CLIP and PAR-CLIP lost fragments is typically over 80% [133]. Second generation CLIP protocols, such as iCLIP [78] and eCLIP [143], were developed to capture both read-through and truncated cDNAs, allowing the detection of crosslink sites with a much higher resolution. In this thesis we aim for single-nucleotide resolution and are thus mainly interested in CLIP-seq protocols that are capturing truncated cDNAs, which will therefore be described in more detail in the following section.

3.2.1. Truncation-based CLIP protocols

Although a wide range of derivatives exist to date, such as 4SU-iCLIP [68], FAST-iCLIP [47], irCLIP [156], Fr-iCLIP [17], seCLIP [145] and FLASH [4], in the context of this thesis we focus on the currently most widely used variants iCLIP and eCLIP.

iCLIP was developed in 2010 and uses a cleavable adapter in combination with an additional circularization step, which allows all cDNA fragments to be amplified and sequenced (see Figure 3.1, middle column). In 2016 eCLIP was published [143], which also captures truncated cDNAs, but introduced several modifications to the protocol (see Figure 3.1, right column). Since the vast majority of captured cDNAs in these protocols are truncated, a feature exploited for the computational analysis, we refer to them as *truncation-based* CLIP protocols.

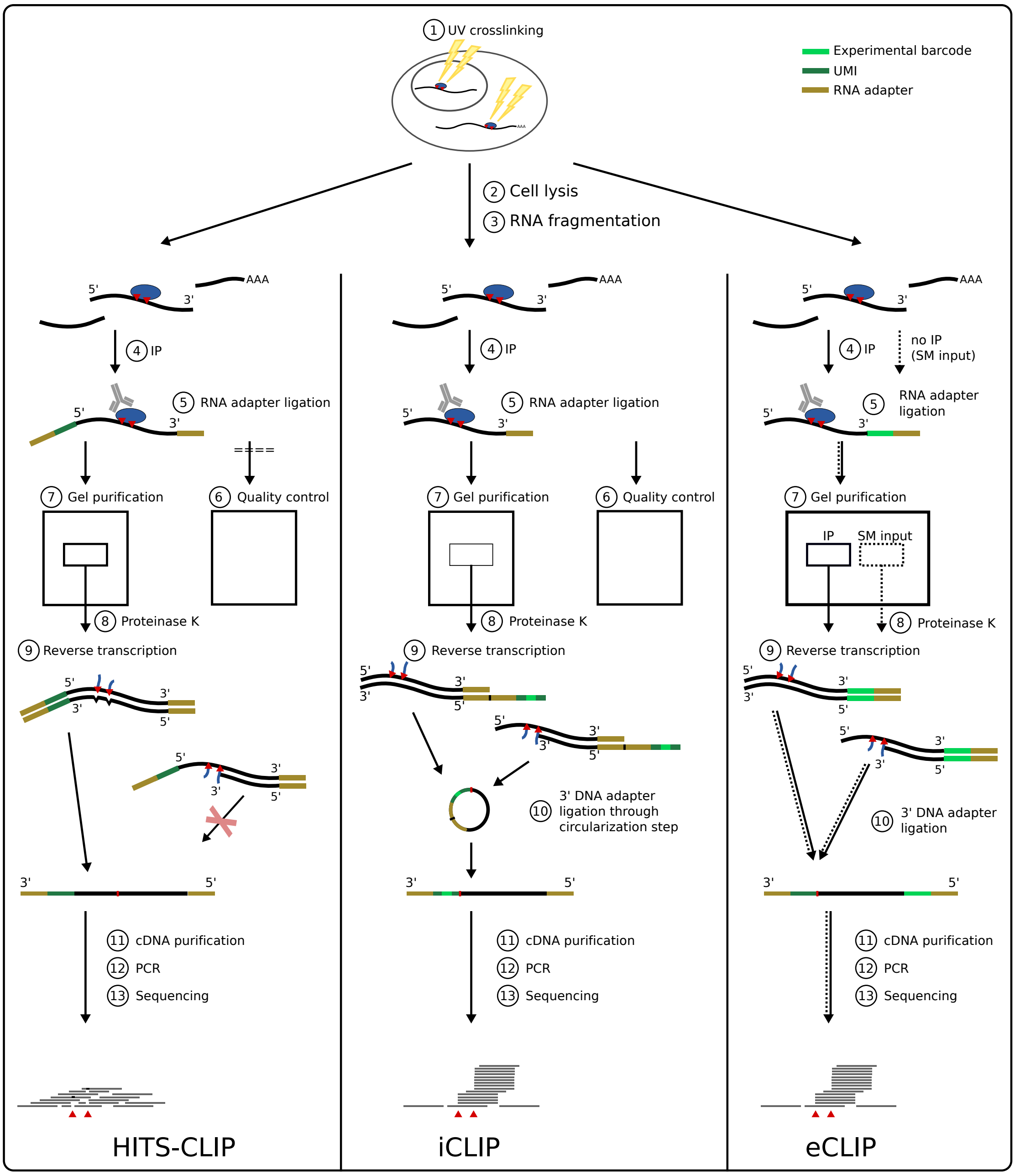


Figure 3.1.: Comparison of CLIP-seq protocols and their main steps. The HITS-CLIP protocol, where truncated cDNAs are lost, in comparison to the truncation-based iCLIP and eCLIP protocols.

Experimental protocol

The common main steps of iCLIP and eCLIP protocols, and to some extend also of the derived variants, are the following [83, 150]:

1. UV crosslinking: in vivo UV-C treatment (254 nm wavelength) causes the formation of covalent crosslinks at sites with direct protein-RNA interactions.

2. Cell lysis: stringent buffer is used to break down cell membranes as well as most protein-protein and protein-RNA interactions, which prepares the RNA for the following fragmentation.

3. RNA fragmentation: RNase conditions are chosen such that an optimal RNA fragment length distribution is achieved. Too long fragments should be avoided to increase the resolution and to minimize the number of cDNA truncations at off-target crosslinks. Too short fragments should be avoided as well, to allow a unique mapping against the genome. Further, in case of target binding sites upstream of RNA cleavage sites [61] (see Section 3.3) a narrow range of short RNAs is more likely to cause biases. Thus, optimally, this step yields a large range of fragment sizes [61], typically between 30 and 200 bp [83] or 300 bp [126].

4. Immunoprecipitation (IP): protein-RNA complexes are pulled down using antibodies specific for the protein of interest. Note, that for many RBPs suitable antibodies are lacking. In such cases, peptide tags can be inserted into the RBP loci with CRISPR/Cas9 [115], enabling a pull-down of the protein-RNA complexes using tag specific antibodies [144].

5. RNA adapter ligation: 3' end adapters required for reverse transcription are ligated. Importantly, 5' end adapters are not yet ligated to account for subsequent cDNA truncations.

6. Quality control (optional): controlling specificity of pulled-down protein-RNA complexes using SDS-PAGE with high RNAse conditions. The desired band is slightly above the expected band of the target protein, additional bands indicate contamination with background proteins.

7. Further purification: SDS-PAGE and transfer to nitrocellulose membrane are used to purify the target protein-RNA complex and remove remaining free RNA fragments or adapters. A region is excised that corresponds to the molecular weight of the protein-RNA complexes of interest ($\leq$220 nt of RNA [143]).

8. Protein removal: Proteinase K is used to digest proteins, leaving a small peptide at the crosslink site.

9. Reverse transcription (RT): cDNA is synthesized from the 3' end of the RNA towards the 5' end. The reverse transcriptase is likely to be interrupted by the remaining peptide at the crosslink site, which causes the truncation of cDNAs.

10. DNA adapter and random barcode ligation: 3' adapters required for PCR are added using different strategies. In iCLIP, a cleavable adapter is introduced via the RT primer. In combination with an extra circularization and linearization step, the adapter is brought to the 3' end (see [78]). In contrast, in eCLIP the adapters are directly ligated to the 3' cDNA ends. In both protocols, additionally random barcodes are introduced allowing for downstream PCR duplicate removal (see Section 3.3.1).

11. cDNA purification: removal of remaining free adapters and cDNA size selection (using gel purification or/and silane beads).

12. PCR amplification of cDNAs.

13. High-throughput sequencing: iCLIP libraries are single-end sequenced. The read starts originate from the 3' cDNA ends and thus contain information about potential crosslink sites. eCLIP libraries are paired-end sequenced and the read start of the second read contains the information about potential crosslink sites. For both protocols Illumina technologies are used to sequence reads of typically 50 nt length, including barcodes [68, 143].

Variations in all of these steps can have a major impact on the resulting data, and should be appreciated for downstream data analysis. For a more detailed description of recent advances in CLIP technologies see Lee & Ule [83].

Importantly, in the eCLIP protocol additionally a *size-matched (SM)* input control experiment is generated, which is produced using 2% of the input lysate prior IP, run on the SDS-PAGE, transferred to the membrane and excised in a size-matched manner. All other steps of the protocol are done in the same way as for the target experiment.

More and more iCLIP datasets are getting published and recently, various improvements to the protocol were proposed to alleviate previous limitations [61, 126]. Besides, to date, 223 eCLIP datasets for 150 different proteins have been published by the ENCODE consortium [31, 109, 129].

Diagnostic events

For truncation based CLIP protocols read starts can be used as diagnostic events to infer crosslink sites. It has been shown for iCLIP data that the vast majority of cDNAs is truncated [61], which illustrates the importance of this signal for inferring the exact protein-RNA interaction sites. Importantly, such truncations can occur a target-specific crosslink sites as well as at off-target crosslink sites. The exact frequency of such diagnostic events is difficult to estimate, since it varies between different experimental conditions and highly depends on the individual binding characteristics of the proteins. Deletions and mutations also occur in iCLIP and eCLIP protocols, but with far lower probabilities and importantly are not that well distinguishable from other causes [61], e.g. SNPs and sequencing or PCR errors.

3.2.2. CLIP experiments for subcellular compartments

In standard CLIP protocols, whole cells are lysed and used for the experiment. However, most proteins carry out specific regulatory functions in certain subcellular compartments. While already described in the past for non-truncation based CLIP methods [121], in 2017 a new protocol called Fr-iCLIP was published [17], performing iCLIP experiments for individual fractions, i.e. the cytoplasm, nucleoplasm and chromatin fraction. Such methods allow to capture the binding regions and characteristics specifically for different subcellular compartments.

3.3. Biasing factors

In order to infer target-specific RBP binding regions from truncation-based CLIP data, it is crucial to account for different sources of biases. When comparing CLIP experiments of different target RBPs, the overall signal can be of different strength and quality, depending on the protein abundance and the general RNA binding affinity of the RBP. Besides, when comparing binding sites of one RBP over the whole transcriptome, the strengths of the CLIP signals highly depend on the RNA abundance, both for target but also for background noise signals. PCR and mapping artefacts, for example within repetitive regions [158], additionally contribute biases.

Furthermore, background noise such as signal coming from sticky RNA fragments that were not washed away or from the binding of background binding proteins [48] as well as crosslinking biases [61, 128, 133] constitute a major challenge for the analysis of CLIP data. The most important biasing factors will be discussed in more detail in the following sections.

Another type of bias is caused by predominant RNAse cleavage sites located downstream of protein binding sites [61], leading to depleted or shifted signals due to the inefficient mapping of short reads. However, such types of biases are rather specific for each protein and protocol and should be addressed experimentally.

3.3.1. PCR artefacts

During the cycles of PCR amplification (see Section2.4) not all fragments are amplified equally, which causes a quantitative PCR bias. Additionally, sequence errors occur, which propagate during subsequent PCR cycles.

In comparison to RNA-seq, CLIP libraries are typically rather sparse and require more amplification cycles prior to sequencing. As a consequence, PCR artefacts are likely more intense with more adverse effects, in particular for truncation-based CLIP data where read starts are used as diagnostic events.

To overcome this problem, recent CLIP technologies [78] make use of random barcodes, also called *unique molecular identifiers* (UMIs). UMIs are ligated prior to PCR, and thus can be used to distinguish PCR duplicates from real biological cDNA duplicates, preserving the quantitative information. Furthermore, they can be used to detect and correct potential PCR sequence errors [130].

3.3.2. Non-specific background binding signal

One of the main problems for the analysis of CLIP data is signal from non-specific background binding, which can have different causes:

1. Co-purification of other proteins binding the same RNA fragments as the target protein. This can lead to *off-target* cDNA truncations causing false positives when assigning binding sites based on read starts.

2. Insufficient removal of non-specific proteins or sticky RNAs during washing often leads to strong background noise, in particular when highly abundant RNAs are bound by highly abundant proteins.

Certain transcripts, for example MALAT1, are bound by many proteins and are highly abundant in different tissues or conditions inducing a signal in most CLIP datasets. A previous study analysing published PAR-CLIP datasets showed that if no control dataset is used for correction, up to 45% of the called binding sites overlap with background binding sites, defined using control CLIP experiments [48]. Predominant background binding regions that are common to several CLIP datasets have been systematically identified [119] and can be used to validate the specificity of called binding sites. However, discarding such regions might prevent the detection of real binding sites. Instead, control experiments should be used for normalization to reduce the number of false positives within such regions.

Figure 3.2b shows an example for PUM2 eCLIP signals within a MALAT1 region, which strongly resemble the signals within the input control dataset and thus are most likely caused by non-specific background binding. For comparison, Figure 3.2a shows an eCLIP signal which is specific for the target dataset.

3.3.3. Transcript abundances

Besides having an effect on non-specific signals, different transcript abundances also affect target-specific CLIP read counts, making a transcriptome-wide comparison difficult. Importantly, depending on where the protein preferentially binds, also varying abundances between nascent (currently transcribed), immature (not fully spliced yet) and mature transcripts can have an effect. Thus, varying abundances between introns and exons will likely be reflected in the CLIP signals, independently of the RBPs affinity to the respective binding regions. Moreover, for nascent transcripts also varying abundances within introns and exons occur [6, 24].

3.3.4. Crosslinking biases

Another major bias is caused by different crosslinking efficiencies of different RNA sequence and structure contexts as well as of different amino acids [125]. Several studies showed a strong (UV) *crosslinking bias (CL-bias)* towards uridine-rich sequences [61, 128, 133]. Such crosslink associated motifs are referred to as *CL-motifs* [61]. Furthermore, it is known that proteins binding to double-stranded RNAs have a rather

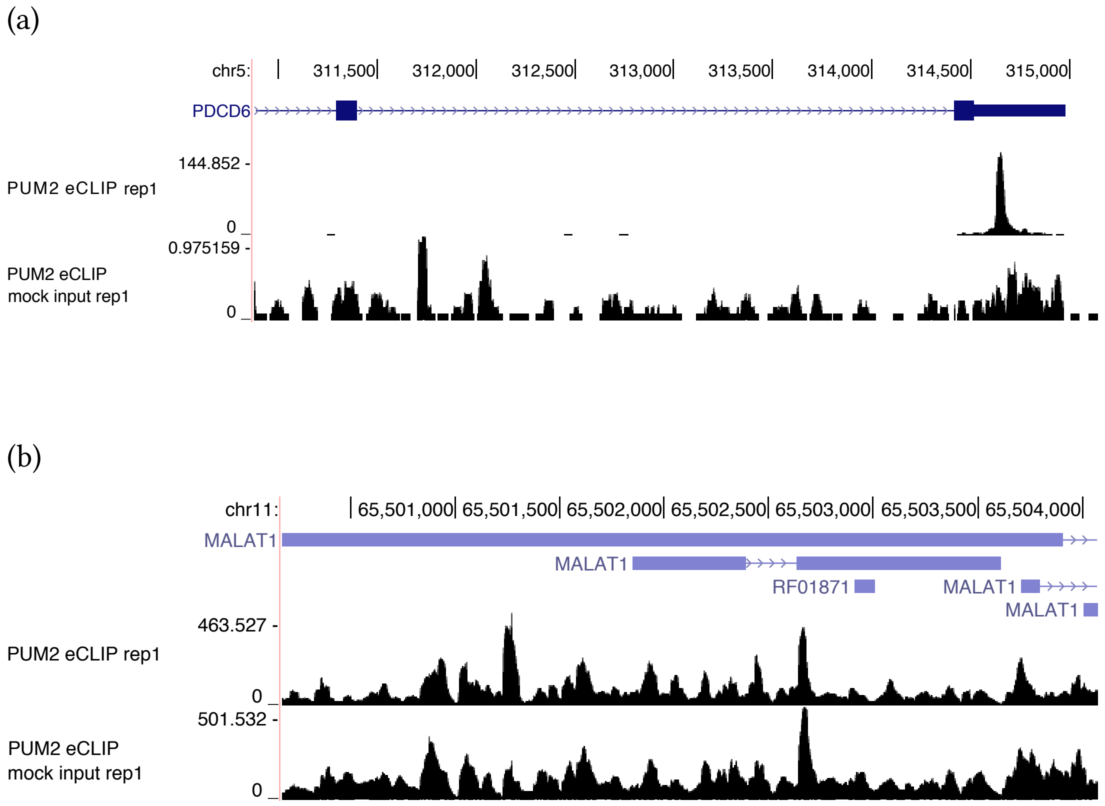


Figure 3.2.: UCSC Genome Browser [70] view showing position-wise read coverages for PUM2 eCLIP data and corresponding size-matched input data [143] (the browser tracks were directly created from the already preprocessed data ENCSR661ICQ and ENCSR439GXW provided by the ENCODE project). The numbers on the y-axes denote the read counts normalized by the total number of reads in the dataset. For conciseness, only tracks for reads mapped against the forward strand are displayed. Shown are **a)** eCLIP signals within a region at the 3' end of PDCD6 and **b)** within a MALAT1 region. Note that PUM2 is known to bind within 3'-UTRs.

poor crosslinking efficiency, because they mainly interact with the phosphate-sugar backbone of the RNAs and not with individual nucleotide bases [128]. As a consequence, single-stranded uridine-rich motifs show the highest crosslinking efficiency.

For proteins binding preferentially to motifs with high crosslinking efficiencies, CLIP data is likely already of higher quality and *CL-biases* are less of an issue. In contrast, if the target protein binds to motifs with low crosslinking efficiencies, background binding - in particular of co-purified proteins bound to the same RNA fragments at sites with higher crosslinking efficiencies [83] - can cause serious biases.

3.3.5. Reverse transcription offsets

Up until recently it was assumed that in case of cDNA truncations the reverse transcriptase terminates in such a way that the last cDNA base is located one nucleotide upstream of the actual crosslink site [78]. A recent study revealed for two examined proteins that besides the truncation rate also the truncation position depends on the used reverse transcription enzyme and buffer conditions [146]. More precisely, in case of the AffinityScript reverse transcriptase, commonly used in eCLIP, cDNAs were truncated upstream of the crosslink sites, while in case of the SuperScript transcriptase, commonly used in iCLIP, the last cDNA bases were at the crosslink sites. Such differences should be investigated further and considered for the individual computational analysis.

3.4. Control experiments

Additional SDS-PAGE quality visualization steps, investigating the purity of the pulled-down protein-RNA complexes, can be performed during the experiment in combination with high RNAse conditions (see Section 3.2.1, protocol step 6), non-specific IPs, knock-out cells without the protein or non-crosslinked cells [83].

Beside controlling the overall quality of the experiment, control experiments can be also coupled with high-throughput sequencing to generate control data for downstream computational normalization. Different strategies exist that aim to represent the non-specific background signal within the target experiment. In the past, often paired control CLIP experiments were generated using non-specific antibodies to control for potential contaminations with background proteins. Ideally, such control data does not contain any or at least much lower signals compared to the target data. This sparsity additionally causes high amplification rates [143], which makes it unsuitable to use for normalization. The same holds consequently for control CLIP experiments on cells where the target protein is knocked out or where the crosslinking step is omitted.

The eCLIP protocol is designed to generate a size-matched (SM) input control where only the IP step is omitted (see Section 3.2.1). Since the input sample is run through the SDS-PAGE as well, it represents RNA fragments crosslinked to a mixture of background proteins with a similar molecular weight as the target protein. Thus, it reflects a combination of different biases: background binding, crosslinking preferences, different transcript abundances and mappabilities. However, it is worth noting that in the target data non-specific background signal might arise due to co-purified proteins bound to the same RNA fragments as the target protein [83] causing off-target cDNA truncations (see Section 3.3.2), which might not be represented by input control data. Nevertheless, concerning such off-target truncations, SM input control is probably more suitable for normalization than applying non-specific IPs which add an additional, potentially biased, subsampling layer. Furthermore, since a large fraction of background noise is simply caused by highly abundant, sticky RNAs bound by highly abundant or many different proteins [150], input data is a highly valuable method for normalization (see Figure 3.2b).

Another possibility is to use corresponding RNA-seq data for normalization in order to account for different transcript abundances. However, how exactly its signal is best used for normalization, e.g. at which resolution, remains to be explored, as RNA-seq and CLIP-seq signals are quite different. Further, predominant subcellular compartments of the RBP should be taken into account [24]. For example, if the protein mainly binds transcripts which are still chromatin associated, standard whole cell poly(A) RNA-seq data or whole cell input control data is not the best choice for normalization.

3.5. Comparison of iCLIP and eCLIP

In comparison to other NGS protocols, CLIP libraries tend to have a rather low complexity due to the sparse nature of protein-RNA binding landscapes. Because of this and a generally low amount of pulled-down RNAs, a relatively high PCR amplification rate is needed prior to sequencing, in particular in combination together with inefficient adapter ligations. In eCLIP several steps were optimized, including the adapter ligation, improving the efficiency of the protocol. In comparison to iCLIP, eCLIP was shown to produce cDNA libraries with an increased complexity, which decreases the required PCR amplification up to 1,000-fold [143]. One possible explanation is that the eCLIP protocol is more efficient in capturing RNA fragments bound by the target protein, e.g. also at low affinity binding sites. However, the increased complexity could also be caused by non-specific background noise and should be handled with care, in particular since eCLIP omits the quality control visualization step [83].

This quality visualization step (see Section 3.2.1, step 6) is an advantage of the iCLIP protocol when it comes to specificity, because it allows quality control already during the experiment and provides further important information when published together with the data. In contrast, the eCLIP protocol omits this step in order to increase the efficiency. On the other hand, eCLIP generates size-matched input control data, allowing for a downstream computational normalization for non-specific background noise. Furthermore, in the eCLIP protocol the experimental barcodes are introduced already during the 3’ RNA adapter ligation, which allows for a high efficiency by multiplexing experiments. Taken together, one can conclude that the strength of iCLIP lies more in specificity, while eCLIP is optimized for efficiency.

3.6. Computational methods to detect protein-RNA interactions from CLIP data

In order to understand regulatory processes and how they are mediated by RBPs, it is crucial to accurately measure the interactions between proteins and RNAs. To derive this information from CLIP data, adequate computational methods are required. Ideally, we would like to obtain the exact binding affinities of the target protein for different binding regions, such that they are comparable across the whole transcriptome or even across experiments. In the following sections we will briefly describe the key challenges when interpreting CLIP data, give an overview of existing methods and motivate the development of a novel approach.

3.6.1. Key challenges

The signals in the data are influenced by different biasing factors and in order to detect target-specific interactions we need to address three main challenges:

- **Definition of protein-RNA interaction footprints**

  To detect protein-RNA interactions from CLIP data, we first need to specify the generated footprints that we expect to observe in the data and that we want to capture. For example, most conventional methods simply aim to capture high peaks of bin-wise read counts. Alternatively, one could aim for individual positions with high read start counts. In any way, defining, encoding and possibly combining the signals of interest from the raw data is a crucial step in the method design, as it strongly impacts the results.

- **Detection of target-specific signals**

  We need to distinguish target-specific CLIP signals from background noise. Therefore we need to accurately model the signal distribution over the whole transcriptome or at least broader regions. Often this is done by modeling the signal, e.g. read counts, using a mixture model, assuming one background and one target component.

- **Bias correction**

  Additionally, we need to be aware of the various sources of biases which have been shown to heavily affect both iCLIP [61] and eCLIP data [143] and that can lead to differences in signal strengths as well as cause non-specific background signals (see Figure 3.2b). Thus, to avoid calling false positives, it is crucial to explicitly account for non-specific background binding, different transcript abundances and crosslinking preferences (see Section 3.3).

Only accurate computational methods tailored to the specifics of the used CLIP protocol enable us to draw reliable conclusions from the biological experiments.

3.6.2. Existing methods

Several tools have been developed for the computational analysis of HITS-CLIP and PAR-CLIP data [33, 124, 141], but only few tools have been developed for the specific analysis of truncation-based CLIP data, such as iCLIP and eCLIP data. In the following we first describe protocol-independent peak-calling methods, then discuss tools that are designed for HITS-CLIP or PAR-CLIP data and, finally, describe the most important tools that can be applied to truncation-based CLIP data. Note that some tools comprise a set of functions to pre- and post-process CLIP data, however, we focus on the functionalities for the core analysis here.

CLIP type independent methods

Piranha performs strand-specific peak-calling [141]. It models the underlying bin-wise read counts using a zero-truncated negative binomial distribution, assuming the majority of reads originates from background signal, and computes a genome-wide significance threshold above which peaks are reported. Thus, it does not explicitly model background and target signals, but it supports the additional incorporation of bin-wise covariates to correct for non-specific background signals.

CLIPper is also a strand-specific peak-calling method [88] and was used by the ENCODE consortium for the analysis of the published eCLIP datasets [143]. It incorporates gene annotations from the reference genome and uses a three-pass filter to reduce the number of false positives. 1) It applies a permutation test, where the observed reads are randomly placed within the gene. Given the distributions, for each position and its observed read coverage a false-discovery rate (FDR) is computed. Only positions with an FDR below a given threshold are kept. 2) The position-wise read coverage is interpolated across the transcript using a cubic spline fitting. From this the peaks, centers and widths are defined. 3) A Poisson distribution is used to model the peak-wise read count distribution across the whole transcriptome and to assess whether a peak is significant given a p value threshold.

Methods designed specifically for HITS-CLIP or PAR-CLIP data

PARalyzer (PAR-CLIP data analyzer) [33] is a tool that exploits PAR-CLIP specific T-to-C conversions. It uses a kernel density estimation with a Gaussian kernel function to estimate the local crosslinking signal of T-to-C conversions and compares it to the background signal of T-to-A/G/T conversions. Sites with a minimum read coverage and crosslinking signal above the background signal are considered interaction sites. In this way, obtained regions can then be further extended by a user-defined length.

CIMS is a method from the CLIP Tool Kit (CTK) designed to detect crosslink-induced mutation sites (CIMS) [102, 124]. For a selected type of mutation, i.e. substitution, deletion or insertion, a permutation test is used to assess the significance of the observed mutations at one position given the overlapping reads. For this purpose, each observed mutation is placed randomly in one of the overlapping reads, while keeping the distance to the 5' end of the read to account for position-specific sequencing error rates. Then, for each candidate crosslink site an empirical FDR is computed.

Methods able to detect crosslink sites based on truncation signals

PIPE-CLIP is an online pipeline for the analysis of HITS-CLIP, PAR-CLIP and iCLIP data [25]. It separately calls enriched read clusters and individual crosslink sites, which are subsequently merged. It models cluster-wise read counts using a zero-truncated negative binomial distribution. To detect crosslink sites, for each position it uses a binomial distribution to model the probability of the observed read start counts given the position-wise read coverage. Interestingly, the success probability of the binomial distribution is set to the average genome coverage. Although based on a powerful idea, one drawback of this method is that it is designed as an online tool and cannot be easily integrated into other types of workflows.

CITS is another method from the CTK [124, 149], which calls individual crosslink sites from iCLIP data, similarly to CIMS for HITS-CLIP data. It clusters reads based on their starts and uses a permutation test to detect positions within such clusters with a significantly elevated fraction of read starts. Called crosslink sites within a defined distance (by default: 25 nt) are then further clustered to binding regions.

iCount was developed particularly for and along with the iCLIP protocol [24, 36]. Similarly to CITS, it uses a permutation test to detect sites that are significantly enriched in read starts within a defined region. The difference is that iCount uses genomic annotations to group read start sites for this purpose, either gene-wise (by default), transcript-wise or transcript-wise while additionally separating between different genomic features (CDS, introns, 3'-UTR, 5'-UTR, ncRNA or for intergenic regions). When scoring individual sites, iCount does not only consider the read starts at the corresponding position, but also considers the read start counts a few nucleotides upstream and downstream by using a moving sum (by default: ± 3 nt). This strategy boosts the detection of crosslink sites with additional crosslinks in their direct neighbourhood. Crosslink sites are then called based on a defined FDR threshold. In a second step, the sites are then clustered based on their distance (by default: 20 nt).

3.6.3. Motivation for a revised computational model

The described existing peak-calling and individual crosslink site detection methods each have their strengths and weaknesses. However, prior to this work no method for the analysis of truncation-based CLIP data existed that performs peak-calling and individual crosslink site detection simultaneously while correcting for experimentally introduced biases.

General peak-calling strategies such as Piranha and CLIPper have the limitation that they potentially miss low affinity binding sites or sites within in lowly abundant RNAs but with a clear pattern of diagnostic events. At the same time, they are more sensitive to peaks caused by the binding of background proteins within highly abundant RNAs.

Piranha can also be used to detect enrichments in read start counts at single-nucleotide resolution by adjusting the bin-size. A major drawback in this context is that the counts are not modeled in relation to the counts in their neighbourhood. This issue is partly addressed by PIPE-CLIP, which models the read start counts in relation to the position-wise read coverage. However, as already mentioned, for iCLIP data the position-wise coverage is influenced by truncations, which likely biases the results.

In contrast, methods such as CITS and iCount detect sites with a significant fraction of read starts using region-wise permutation tests to compute FDRs for each position based on its observed read start count. With this they indirectly normalize for the number of reads within such regions, i.e. within a gene, transcript or read cluster. Their assumption under the null hypothesis is that the reads within a region start with the same probability at each position within this region. However, the distribution of read starts is highly biased by the sequence, structure and local transcript abundances, which is not taken into account in these models. As a result, these methods are sensitive to such artefacts, especially within highly abundant RNAs. Furthermore, while CITS normalizes read start counts for clusters of read starts, iCount normalizes read start counts for the whole gene or transcript. This has the disadvantage that local changes in transcript abundance are not accounted for (see Section 3.3.3). iCount's feature-wise normalization addresses differences between exons and introns, but since often multiple overlapping annotations exist, CLIP signals assigned to wrong annotations are likely cause artefacts.

Methods that are designed to capture diagnostic events in PAR-CLIP and HITS-CLIP data, such as PARalyzer and CIMS, cannot be easily extended for the analysis of truncation-based CLIP data. The reason for this is that in order to assess the significance of observed mutations, both methods make use of the reads covering the corresponding position. However, in truncation-based CLIP data the position-wise read coverage is highly influenced by neighbouring crosslinking events, which would impair the results if one would simply add read starts as diagnostic events.

To address the limitations inherent in existing approaches, we have developed PureCLIP, a method to accurately model and capture protein-RNA interaction footprints from truncation-based CLIP data. PureCLIP calls individual crosslink sites considering both regions enriched in protein-bound fragments and the iCLIP/eCLIP specific truncation patterns. As we have seen how strong signals within eCLIP data can originate from non-specific background noise (see Figure 3.2b), we have designed our method to specifically correct for such biases. For this purpose, PureCLIP is based on a non-homogeneous hidden Markov model which allows for the incorporation of additional factors into the model, such as non-specific background signal from input control experiments and, for the first time, CL-motifs. With this we aim to reduce the number of false positives within bias prone regions, while increasing PureCLIP's sensitivity outside such regions. Lastly, PureCLIP can also incorporate individual replicates, which further helps to reduce the number of false positives caused by artefacts present only in one replicate. We comprehensively evaluated the performance of PureCLIP in detecting individual crosslink sites as well as binding regions and will demonstrate its superiority over existing methods, on both simulated and experimental iCLIP and eCLIP data.

4. Preliminaries

In this chapter we define the notations that we use throughout this thesis and introduce the basic statistical concepts providing the basis for the developed methods.

4.1. Notations

Random variables are usually denoted in upper case letters, while their realizations are denoted in lower case. Symbols in boldface denote vectors, matrices or sequences, while scalars and parameter sets are set in regular typeface. If not stated differently, observations are denoted with y. A sequence of observations is written as $\boldsymbol{y} = y_1, \ldots, y_T$, where T denotes the number of observations. The subsequence from position i to j is written as $\boldsymbol{y}_{i:j} = y_i, \ldots, y_j$.

Also note that the variables α and β are used in different contexts, i.e. in the context of the forward-backward algorithm (see Sections 4.2.5 and 5.6.2) and as regression coefficients for GLMs (see Sections 4.2.7 and 5.4) to avoid complicated notations. The current meanings will become apparent in the respective contexts.

Lastly, in the context of updating a parameter θ, we denote with θ' the previously learned and with θ'' the updated value.

4.2. Fundamental statistical concepts

4.2.1. The binomial probability distribution

The binomial distribution is a discrete probability distribution used to model the number of successes in n independent experiments (trials) with binary outcome, e.g. failure or success. The probability to observe exactly k successes in n trials is:

$$P(k; n, p) = P(Y = k) = \binom{n}{k} p^n (1-p)^{n-k}, \tag{4.1}$$

where p is the probability to get a success for each trial. The term $\binom{n}{k}$ is the binomial coefficient and denotes the number of ways to distribute k successes over n trials. The expected value of a binomially distributed random variable $Y \sim B(n, p)$ is $E[Y] = np$.

4.2.2. The gamma probability distribution

The gamma distribution is a continuous probability distribution defined by two parameters to model non-negative, right-skewed values. Given a shape parameter λ and a mean parameter μ, the probability density function is:

$$P(y;\mu,\lambda) = P(Y = y) = \frac{y^{\lambda-1}e^{-\frac{\lambda y}{\mu}}}{\left(\frac{\mu}{\lambda}\right)^{\lambda}\Gamma(\lambda)}, \qquad \text{for } y \geq 0 \text{ and } \lambda, \mu > 0, \tag{4.2}$$

where $\Gamma(\lambda)$ is the ordinary gamma function. Alternatively, the gamma distribution is often described with the shape parameter λ and a scale parameter $\upsilon = \frac{\mu}{\lambda}$. In the special case that the shape parameter is $\lambda = 1$, the gamma distribution is equal to the exponential distribution. In contrast, for large λ the gamma distribution converges to a normal distribution.

4.2.3. Maximum likelihood estimation

Given a statistical model, for example a certain type of probability distribution, and the observations $\boldsymbol{y} = y_1, \ldots, y_T$, a common task is to infer the model parameters θ. The *maximum likelihood estimator* θ^{MLE} of θ is the set of values that maximize the probability for the observed data $P(\boldsymbol{Y} = \boldsymbol{y};\theta)$, or more formally:

$$\theta^{MLE} = \arg\max_{\theta} P(\boldsymbol{Y} = \boldsymbol{y};\theta). \tag{4.3}$$

If the observations are independent, this can be written as:

$$\theta^{MLE} = \arg\max_{\theta} \prod_{t=1}^{T} P(Y_t = y_t;\theta). \tag{4.4}$$

In this context $P(\boldsymbol{Y} = \boldsymbol{y};\theta)$ is a function of θ and called the likelihood function of the parameters θ, also written as $\mathcal{L}(\theta \mid \boldsymbol{y})$. This method of parameter estimation is called *maximum likelihood estimation* (MLE).

In practice, often the logarithm of the likelihood is used, which can be written as:

$$\theta^{MLE} = \arg\max_{\theta} \sum_{t=1}^{T} log\ P(Y_t = y_t;\theta). \tag{4.5}$$

The log likelihood has the same maxima as the likelihood function, but increases numerical stability and often simplifies derivation as well as obtaining closed form solutions. Assuming $\boldsymbol{Y}$ are independent, identically distributed random variables, then θ^{MLE} is asymptotically normally distributed.

4.2.4. Truncated probability distributions

In certain scenarios, the distribution of the observed data is truncated at one or both ends, because values above or below a certain threshold can not be measured. A typical example is the distribution of raindrop sizes, which are measured with a specific instrument, not capturing raindrops below a certain size [76]. If we now assume that

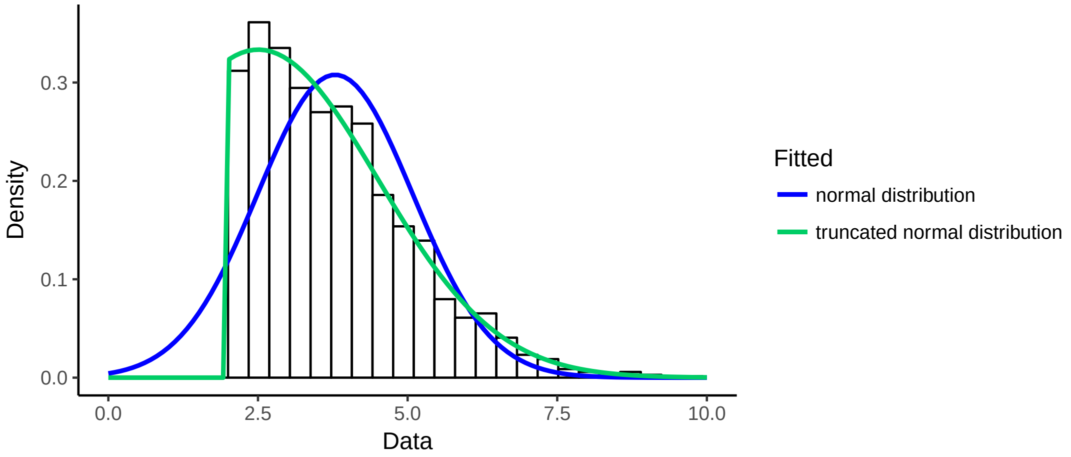


Figure 4.1.: Truncated data and the effect on parameter estimation. The data (bars) is sampled from a normal distribution which is left-truncated at a value of 2. The blue line represents a non-truncated normal distribution that was fitted to the data using MLE, while the green line represents a fitted left-truncated normal distribution with the truncation point 2.

the data follows a certain distribution, the parameter estimation, e.g. MLE, can be severely biased due to the missing values. Figure 4.1 illustrates the possible impact on the parameter estimation.

To address this problem, such data can be modeled using truncated probability distributions. For example, given the non-truncated continuous probability distribution $f(y)$, the left-truncated probability distribution with the truncation point τ can be derived as:

$$f_{trunc}(y) = P(Y = y; \tau) = \begin{cases} 0 & \text{if } y \leq \tau \\ \frac{f(y)}{1-F(\tau)} & \text{if } y > \tau, \end{cases} \tag{4.6}$$

where

$$F(\tau) = P(Y \leq \tau) = \int_{-\infty}^{\tau} f(x)\,\mathrm{d}x \tag{4.7}$$

is the cumulative distribution function up to the truncation point and the normalization term $1 - F(\tau)$ ensures that $\int_{\tau}^{\infty} f_{trunc}(y)\,\mathrm{d}y = 1$. In a similar way, right-truncated as well as truncated discrete distributions can be defined.

In other scenarios certain values, typically zeros, are inflated due to an additional component. Such data can be modeled with corresponding zero-inflated distributions, however, if the zero fraction is of no further interest this is often modeled using truncated distributions as well.

4.2.5. Hidden Markov models

Hidden Markov models (HMMs) assume a spatial dependency between neighboring positions and are commonly used to segment sequences of observations. After their invention in the late 1960s, they were initially mainly used for signal processing problems such as speech recognition [114]. From the 1980s on they were also applied for the analysis of biological sequences, for example, for gene finding [92] and CpG island detection [42], and became a well established tool in the field of bioinformatics [42]. Another well known example in this context is the application to ChIP-seq data, more precisely to profiles of different histone modifications, with the goal to segment the genome into different functional genomic elements [44, 96]. HMMs can be used as unsupervised learning methods, where the data is unlabeled and the model parameters are learned solely based on the data.

HMMs assume a mixture model, where the given observations are caused (or emitted) by discrete hidden states $\boldsymbol{z} = z_1, \dots, z_\ell$. Importantly, in contrast to other mixture models, HMMs include positional information. More precisely, the sequence of hidden states $\boldsymbol{s} = s_1, \dots, s_T$ causing the observations $\boldsymbol{y} = y_1, \dots, y_T$ is assumed to follow a Markov chain, where the probability to be in state $S_t = z_i$ at position t depends on the predecessor states $\boldsymbol{s}_{1:t-1}$. In the following we only consider HMMs with a first-order Markov chain, where the probability to be in state $S_t = z_i$ at position t only depends on the state at position $t - 1$:

$$P(S_t = z_i \mid \boldsymbol{S}_{1:t-1} = \boldsymbol{s}_{1:t-1}) = P(S_t = z_i \mid S_{t-1} = s_{t-1}), \qquad \forall i \in \{1, \dots, \ell\}. \tag{4.8}$$

This memoryless property is called the *Markov property*.

In order to characterize an HMM, first the number of hidden states ℓ needs to be fixed. Next, a transition matrix $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{\ell \times \ell}$ is required, defining the pairwise transition probabilities between the states:

$$a_{ij} = P(S_t = z_j \mid S_{t-1} = z_i), \qquad 1 \leq i, j \leq \ell, \tag{4.9}$$

where

$$\sum_{j=1}^{\ell} a_{ij} = 1, \qquad \forall i \in \{1, \dots, \ell\}. \tag{4.10}$$

Additionally, an initial state distribution $\boldsymbol{\pi}$ defines:

$$\pi_i = P(S_1 = z_i), \qquad 1 \leq i \leq \ell. \tag{4.11}$$

And finally, each state z_i comes with a certain emission probability distribution, here dependent on the parameter set ϑ_i:

$$e_i(y_t) = P(Y_t = y_t \mid S_t = z_i, \vartheta_i), \qquad \forall i \in \{1, \dots, \ell\}. \tag{4.12}$$

Figure 4.2 represents a commonly used example for HMMs, namely the dishonest casino, which occasionally switches between a *fair* and a *biased* dice. The goal is to infer from the sequence of observed dice scores the most likely sequence of hidden

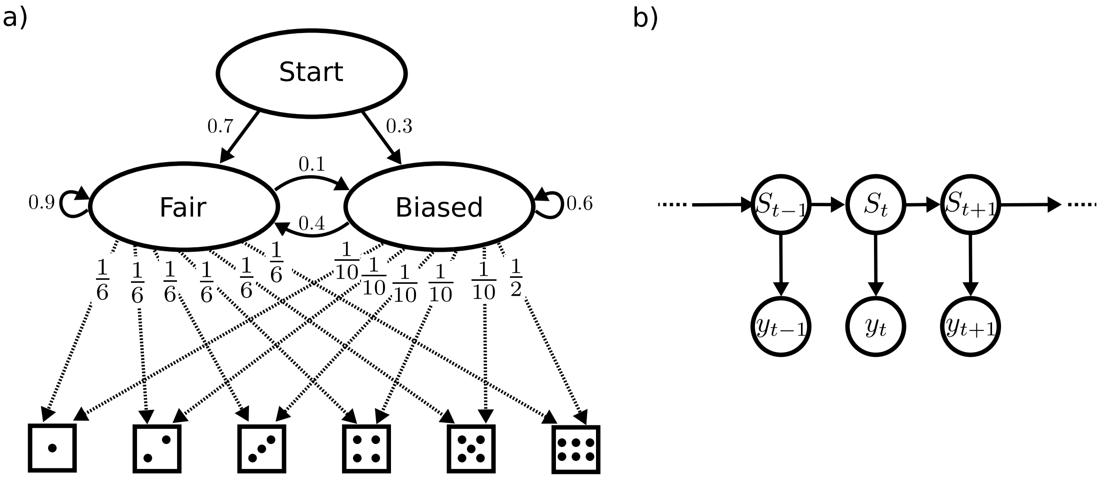


Figure 4.2.: Hidden Markov model example representing a dishonest casino. Most of the time the casino uses a fair dice, but occasionally it switches to a biased dice that generates a six with a probability of $\frac{1}{2}$. **a)** Hidden states (*fair*, *biased*) with their initial probabilities, transition probabilities and emission probabilities for different dice outcomes. **b)** Graphical representation of the model where one column corresponds to one position in the sequence, comprising a hidden state S_t and a given observation y_t.

states (*fair* or *biased*). Note that in general the emitted observations can be discrete or continuous.

While the number of states ℓ is specified in advance, the model parameters $\theta = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\vartheta}\}$ usually need to be learned. Taken together, the joint probability of the observed data $\boldsymbol{y}$ and a sequence of hidden states $\boldsymbol{s}$ is:

$$P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s}) = \pi_{s_1} e_{s_1}(y_1) \prod_{t=2}^{T} a_{s_{t-1}s_t} e_{s_t}(y_t). \tag{4.13}$$

Another important concept are the state posterior probabilities, i.e. the probability to be in state z_i at position t given the observed data and the model parameters, which satisfy the constraint

$$\sum_{i=1}^{\ell} P(S_t = z_i \mid \boldsymbol{Y} = \boldsymbol{y}, \theta) = 1. \tag{4.14}$$

The computation of these probabilities will be described in the following paragraphs, where we address the three basic problems of HMMs [114]: 1) how to compute the probability of the observed data under a given model, 2) how to learn the model parameters and 3) how to infer the most likely (sequence of) hidden states.

Probability for a given sequence of observations under the model

Assume the model parameters θ are known, then the probability for a given sequence of observations can be written as:

$$P(\boldsymbol{Y} = \boldsymbol{y} \mid \theta) = \sum_{\boldsymbol{s} \in \{1,\ldots,\ell\}^T} P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s} \mid \theta), \tag{4.15}$$

where $\{1, \ldots, \ell\}^T$ denotes all ℓ^T possible state sequences. However, this naive enumeration of all possible state sequences is computationally unfeasible. Fortunately, due to the Markov property of the HMM it can be reformulated as a dynamic programming approach referred to as the *forward-backward algorithm* [114]. Here, we denote as the *forward probability* the probability to observe the subsequence $\boldsymbol{y}_{1:t}$ and being in state z_i at position t:

$$\alpha_i(t) = P(\boldsymbol{Y}_{1:t} = \boldsymbol{y}_{1:t}, S_t = z_i \mid \theta). \tag{4.16}$$

It can be computed as follows:

$$\alpha_i(1) = \pi_i e_i(y_1), \tag{4.17}$$

$$\alpha_i(t) = e_i(y_t) \sum_{j=1}^{\ell} \alpha_j(t-1) a_{ji}. \tag{4.18}$$

Thus, the probability of a given sequence of observations under the model is:

$$P(\boldsymbol{Y} = \boldsymbol{y} \mid \theta) = \sum_{i=1}^{\ell} \alpha_i(T). \tag{4.19}$$

In theory, this could be used as a likelihood function for MLE as described in Section 4.2.3, however, since there exists no analytical solution to determine the parameters, an iterative approach is described in the next paragraph.

Although only required for learning the parameters in the next step, similarly to the forward probability a backward probability, i.e. the probability of the subsequence $\boldsymbol{y}_{t+1:T}$:

$$\beta_i(t) = P(\boldsymbol{Y}_{t+1:T} = \boldsymbol{y}_{t+1:T} \mid S_t = z_i, \theta) \tag{4.20}$$

can be computed as follows:

$$\beta_i(T) = 1, \tag{4.21}$$

$$\beta_i(t) = \sum_{j=1}^{\ell} a_{ij} e_j(y_{t+1}) \beta_j(t+1). \tag{4.22}$$

Learning the HMM parameters

The goal is to learn the initial, transition and emission probability parameters $\theta = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\vartheta}\}$ that maximize the likelihood of the HMM given the observed data. However, since the observed data is emitted by a mixture model and the corresponding hidden states are unknown, the optimal parameter estimates could be determined using numerical optimization techniques based on Equation 4.19, but no closed-form solution exists. Expectation-maximization (EM) algorithms provide an elegant alternative solution for such problems by using an iterative procedure. Within each iteration, initially fixed model parameters are used to compute new posterior probabilities of the hidden states given the observations, which are then in turn used as weights to find the parameters that maximize the conditional expectation of the observed data. It can be shown that maximizing the expected log-likelihood for the given observations also maximizes the likelihood [86]. In the following we will describe the Baum-Welch algorithm, which is a special form of the EM algorithm designed for HMMs with discrete state space.

In the context of this algorithm, we need to compute the position-wise posterior probabilities, which can be computed efficiently using the forward and backward probabilities:

$$\gamma_{ti} = P(S_t = z_i \mid \boldsymbol{Y} = \boldsymbol{y}, \theta) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{\ell} \alpha_j(t)\beta_j(t)}. \tag{4.23}$$

Furthermore, we can compute the probability to be in state z_i and z_j at position $t-1$ and t, respectively, given the observed data and the model parameters:

$$\begin{aligned} \xi_{tij} =& P(S_{t-1} = z_i, S_t = z_j \mid \boldsymbol{Y} = \boldsymbol{y}, \theta) \\ =& \frac{\alpha_i(t-1)a_{ij}\beta_j(t)e_j(y_t)}{\sum_{m=1}^{\ell}\sum_{n=1}^{\ell} \alpha_m(t-1)a_{mn}\beta_n(t)e_n(y_t)}, \qquad 1 \leq i, j \leq \ell. \end{aligned} \tag{4.24}$$

The actual algorithm then iterates over the two steps described in the following.

1) Expectation step: Given a fixed set of model parameters θ', the posterior probabilities are updated, which in turn are used as weights to compute the expected log-likelihood $Q(\theta \mid \theta')$ given the observed data:

$$Q(\theta \mid \theta') = \sum_{\boldsymbol{s} \in \{1,\ldots,\ell\}^T} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \cdot \log P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s}; \theta). \tag{4.25}$$

Using Equation 4.13, for HMMs the function becomes:

$$\begin{aligned} Q(\theta \mid \theta') =& \sum_{\boldsymbol{s} \in \{1,\ldots,\ell\}^T} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \\ & \cdot \log \left(\pi_{s_1} e_{s_1}(y_1) \prod_{t=2}^{T} a_{s_{t-1}s_t} e_{s_t}(y_t) \right) \\ =& \sum_{\boldsymbol{s} \in \{1,\ldots,\ell\}^T} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \\ & \cdot \left(\log \pi_{s_1} + \sum_{t=2}^{T} \log a_{s_{t-1}s_t} + \sum_{t=1}^{T} \log e_{s_t}(y_t) \right). \end{aligned} \tag{4.26}$$

After some rearrangements (see Appendix A.1) we obtain:

$$\begin{aligned}Q(\theta \mid \theta') =& \sum_{j=1}^{\ell} \gamma'_{1j} \log \pi_j \\ &+ \sum_{t=2}^{T} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \xi'_{tij} \log a_{ij} \\ &+ \sum_{t=1}^{T} \sum_{j=1}^{\ell} \gamma'_{tj} \log e_j(y_t).\end{aligned} \tag{4.27}$$

Recall that γ'_{tj} and ξ'_{tij} can be computed efficiently using the previously described forward-backward algorithm (see Equations 4.23 and 4.24). However, $Q(\theta \mid \theta')$ is not yet computed, but instead serves as a function for the next step.

2) Maximization step: The parameters that maximize the expected log-likelihood of the data are estimated:

$$\theta'' = \arg\max_{\theta} Q(\theta \mid \theta'). \tag{4.28}$$

Note that the individual parameters can be estimated independently from each other. The initial probabilities $\boldsymbol{\pi}$ as

$$\pi''_j = \gamma'_{1j}, \qquad \forall j \in \{1, \ldots, \ell\}, \tag{4.29}$$

the transition probabilities $\boldsymbol{A}$ as

$$a''_{ij} = \frac{\sum_{t=1}^{T} \xi'_{tij}}{\sum_{t=1}^{T} \sum_{m=1}^{\ell} \xi'_{tim}}, \qquad 1 \leq i, j \leq \ell, \tag{4.30}$$

and the state-wise emission probability parameters as:

$$\begin{aligned}\vartheta''_i =& \arg\max_{\vartheta} \sum_{t=1}^{T} \gamma'_{tj} \log e_{z_i}(y_t) \\ =& \arg\max_{\vartheta} \sum_{t=1}^{T} \gamma'_{tj} \log P(Y_t = y_t \mid S_t = z_i, \vartheta), \qquad \forall i \in \{1, \ldots, \ell\}.\end{aligned} \tag{4.31}$$

Note that the latter term corresponds to a weighted maximum likelihood estimation (MLE).

It is important to note that the Baum-Welch algorithm only finds local optima and does not guarantee a globally optimal solution. For this reason, good initial parameter estimates are crucial. The iterative procedure is terminated when the convergence criterion is met, i.e. usually when the difference between the current and the previous likelihood of the model or the estimated parameters falls below a predefined threshold.

Inference of hidden states

Finally, given the observed data and the learned model parameters, we can address the question which hidden states did most likely cause the observed data. When the goal is to detect the sequence $\boldsymbol{s}^*$ of hidden states, a dynamic programming approach referred to as the Viterbi algorithm [114] can be applied to compute

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}\in\{1,\ldots,\ell\}^T}{\arg\max} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}, \theta). \tag{4.32}$$

In some applications, however, one is not interested in the most likely sequence of hidden states, but rather in the most likely hidden state for each position considering all possible paths. For such cases *posterior decoding* is used, which simply assigns the state with the highest posterior probability to each position:

$$z_i^* = \underset{i\in\{1,\ldots,\ell\}}{\arg\max}\ P(S_t = z_i \mid \boldsymbol{Y} = \boldsymbol{y}, \theta). \tag{4.33}$$

General remarks on HMMs

Since traditional HMMs assume that the observations are emitted by hidden states and do not require any label information, they belong to the *unsupervised* learning methods. The number of states ℓ is fixed in advance, while all other parameters, i.e. the initial, transition and emission probabilities are learned. For initialization, equally distributed probabilities can be used and/or carefully selected values to prevent the algorithm from converging to local, but non-global optima.

However, a large number of extensions and generalizations of classical HMMs exist, for example, supervised HMMs using labeled data for parameter learning [95], factorial HMMs containing multiple independent state variables [71], tree structured HMMs with coupled state variables [73] and Bayesian HMMs making use of informative priors for learning the model structure, parameters and hidden states [58, 71]. For a detailed description of possible generalizations of HMMs and the corresponding challenges regarding inference, parameter learning and model selection see, for instance, Ghahramani 2001 [53].

Another generalization are *non-homogeneous* HMMs, where the model parameters can differ across positions.

4.2.6. Non-homogeneous hidden Markov models

The basic concept of HMMs assumes that the individual observations are dependent on their hidden states – which follow a Markov chain – and independent otherwise. This motivates homogeneous transition and emission probabilities. However, in some applications the observations are additionally biased by confounding factors. In our toy example of the casino (see Figure 4.2), this could be for instance the casino's current bank balance, e.g. the lower the balance the higher the probability that the casino uses the 'biased' dice. Such external information can be included in the model as

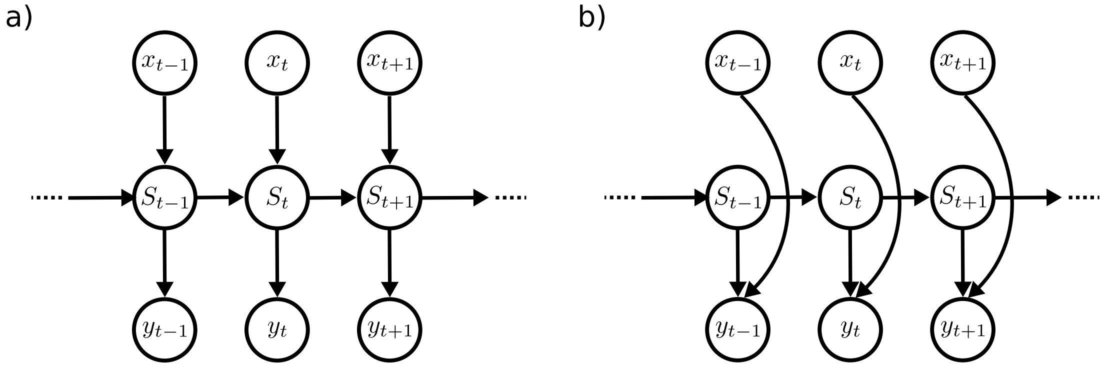


Figure 4.3.: Non-homogeneous hidden Markov models. a) A confounding factor X_t is biasing the actual state probabilities. Consequently, covariates would be modeled to influence the state transition probabilities. **b)** A confounding factor X_t is biasing the actual observations. Accordingly, covariates would be modeled to influence emission probabilities.

covariates $\boldsymbol{X}$. Different structures exist for such non-homogeneous HMMs, modeling an influence either on the state transition probabilities (see Fig. 4.3a)

$$a_{ij}(x_t) = P(S_t = z_j \mid S_{t-1} = z_i, x_t), \qquad 1 \leq i, j \leq \ell \tag{4.34}$$

or on the emission probabilities (see Fig. 4.3b)

$$e_{z_i}(y_t, x_t) = P(Y_t = y_t \mid S_t = z_i, \vartheta_i(x_t)), \qquad \forall i \in \{1, \dots, \ell\}, \tag{4.35}$$

where x_t is a covariate at position t.

The correlation between the model parameters a_{ij} or ϑ_i and the covariates x_t can be learned, for example, using (generalized) linear models. Note that if the emission probability $e_{z_i}(y_t, x_t)$ depends on multiple parameters ϑ_i, usually some of the ϑ_i are modelled dependent on x_t, while others are assumed to be constant.

4.2.7. Generalized linear models

Let us leave the context of HMMs for a moment and let $\boldsymbol{y} = (y_1, \dots, y_T)$ be a vector of observed data and $\boldsymbol{X}$ be a matrix containing explanatory (or predictor) variables, where one row represents $\boldsymbol{x}_t = (x_{t1}, \dots, x_{tp})$. A (ordinary) linear regression model can be used to describe the relationship between the observation y_t and the given explanatory variables $\boldsymbol{x}_t$ as follows:

$$y_t = \beta_0 + \beta_1 x_{t1} + \cdots + \beta_p x_{tp} + \epsilon_t, \qquad \forall t \in \{1, \dots, T\}, \tag{4.36}$$

where ϵ_t denotes the error term and is assumed to follow a normal distribution $\mathcal{N}(0, \sigma^2)$. $\boldsymbol{\beta}$ is a vector of regression coefficients that have to be learned. In other words, the expected values of $\boldsymbol{Y}$ can be predicted with $\boldsymbol{X\beta}$. However, if ϵ_t is not normally distributed, ordinary linear models are not applicable.

Generalized linear models (GLMs) can be used to extend linear models for such data [105]. This requires:

1. A link function $\boldsymbol{X\beta} = g(\mu)$ describing the relationship between the expected value $E(Y_t) = \mu_t$ and the linear predictor.
2. An exponential family probability distribution $f_\vartheta(y_t) = P(Y_t = y_t; \mu_t, \vartheta)$ describing the distribution of the error term ϵ_t.

Consequently, the ordinary linear regression model is a special case of GLMs with normally distributed error terms and the link function $g(\mu) = \mu$.

The regression coefficients $\boldsymbol{\beta}$ of a GLM are typically estimated using MLE (see Section 4.2.3), such that:

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta},\vartheta}{\arg\max} \sum_{t=1}^{T} \log P(Y_t = y_t; \boldsymbol{\beta}, \vartheta). \tag{4.37}$$

In case of ordinary linear regression, the regression coefficients $\boldsymbol{\beta}$ can be estimated using ordinary least squares, which minimizes the sum of squares of the differences between the observed data and the predicted values. This simultaneously maximizes the likelihood function. However, the MLEs of most GLMs do not have such a closed form and the parameters need to be estimated using numerical optimization strategies.

4.2.8. Numerical optimization strategies

To find the maximum (or minimum) of an objective function that can not be found analytically, different strategies exist that are based on function evaluations for different parameters. Often, this parameter space is multi-dimensional. The naive way would be to perform a grid search. While this in theory allows to find the global optimum, it also requires a large number of function evaluations. Numerical optimization strategies are designed to decrease the computational complexity, while aiming to find the global optimum. However, these methods do not guarantee to find the global optimum.

A widely used strategy is based on gradients, such as the well known gradient descent algorithm [107], which evaluates the gradient at a certain point using the first-order derivative and then moves down the gradient until a local minimum is reached. Another commonly used algorithm is the Newton-Raphson algorithm [7]. It additionally requires the second-order derivative to optimize the applied step size and thus usually reaches the optimum faster. Importantly, these algorithms are only applicable if the corresponding function derivatives can be computed.

In cases where the function is either not differentiable or the derivatives are not easily computable, gradient-free algorithms can be used. One example is the Nelder-Mead algorithm [104], also called Simplex algorithm, a relatively simple heuristic approach that can deal with multi-dimensional optimization problems. For n dimensions it uses a n-simplex, i.e. the n-dimensional generalization of a triangle, with the $n+1$ vertices as evaluation points. Iteratively, geometric transformations, such as reflections, expansions and contractions are used to move the simplex towards the optimum. A drawback of the Nelder-Mead algorithm is that is can converge to non-stationary points [100], however, in practice it works well for most low-dimensional problems and is relatively fast compared to other methods [81].

Many modifications of the aforementioned methods exist as well as other strategies with different scopes of application and different disadvantages, but we will not go into detail here. More information about numerical optimization can be found, for example, in Nocedal & Wright 2006 [107].

Part II.

Method and Evaluation

5. Capturing target-specific protein-RNA interactions from truncation-based CLIP-seq data

We discussed in Section 3.6 the main challenges in detecting protein-RNA interactions from CLIP data and how existing methods address these. In this chapter we will present PureCLIP, a new method to detect protein-RNA interactions from truncation-based CLIP data, such as iCLIP or eCLIP. We will first discuss the required preprocessing steps of the data to allow an accurate and quantitative analysis. Then we will present a new computational method to detect target-specific interaction footprints at single-nucleotide resolution, while correcting for various sources of biases.

5.1. Preprocessing

We will briefly discuss the data processing steps required prior to the analysis for iCLIP and eCLIP experiments. As all other NGS data, CLIP data requires adapter removal, filtering based on read lengths and quality control. In the following we will describe the CLIP specific issues that we address in the preprocessing. Although most of the steps are necessary for other NGS data as well, it is important to note that for truncation-based CLIP data artefacts are particular harmful, since individual read start counts are used for the analysis. The used tools and the applied settings are described in Appendix A.3.

A crucial step is the mapping of the reads. We use the read aligner STAR [39], which is designed for RNA-seq data and allows mapping against the genome while taking into account information about possible splice junctions. For most proteins, in particular those that bind within introns, this is the appropriate choice. However, for proteins binding near or across exon-exon junctions on the mRNA, mapping against the genome causes a splitting of the CLIP signal. This in turn likely leads to a decreased sensitivity or to artefacts within the downstream analysis, at least if not explicitly addressed. For this reason, reads from such proteins should be mapped directly against mRNA transcripts (as for instance done in Haberman et al. [61]).

Furthermore, many read mappers clip off terminal regions of the reads to optimize the alignments. Since read start positions are used as diagnostic events, we disable this feature in STAR (`-alignEndsType EndToEnd`) to ensure that the start positions of the alignments correspond to the start positions of the original cDNAs. Moreover, in comparison to RNA-seq, CLIP data contains a higher number of deletions induced by crosslinks [133]. We consider this in the alignment setting by using a lower deletion open penalty (`-scoreDelOpen -1`), though the exact deletion rate is protein and protocol dependent and the best mapping strategy remains to be explored.

Another mapping related question is how to deal with reads mapping to multiple locations. Such multi-mapping reads often originate from repetitive regions, for example Alu elements, which are dispersed over the genome, important for regulatory functions and also bound by RBPs [155]. For CLIP data analysis such reads constitute a major challenge, because they cause huge peaks and pile-ups of read starts, both for target and non-specific background signals. The easiest and most stringent way is to discard all such multi-mapping reads and only run the analysis on uniquely mapping reads. Unfortunately, this also causes the loss of true signals in case the target protein binds to repetitive elements. To include such regions in the analysis, input control data should be included to normalize for this effect, since such mapping artifacts are likely to occur in both datasets. Alternatively, approaches such as CLAM [158] can be used that assign multi-mapping reads to individual locations based on the read coverage in the vicinity. However, in this case the question remains if read starts can still be used as diagnostic events, i.e. assuming crosslinks would occur in general at the same sites within repetitive elements, or if the analysis should rather be limited to peak-level resolution. For the context of this thesis we discard multi-mapping reads.

To allow for an accurate quantification of potential truncation events, we remove PCR duplicates based on the mapping locations and UMIs (see Section 3.3.1). This is important as PCR amplification rates are high, in particular for iCLIP datasets. Moreover, with increased amplification the number of PCR errors within the UMIs rises and leads to groups of different but similar UMIs originating from the same cDNA molecule. To address this, we use UMI-tools [130], a network based method to remove PCR duplicates which is able to handle errors within UMI sequences.

Finally, it should be noted that only reads originating from the 3'-end of the cDNA are used for the analysis (see Section 3.2.1, Step 13). For eCLIP data, which is paired-end, this is the second read of the pair.

5.2. Overview of the PureCLIP approach

PureCLIP aims to capture footprints caused by target-specific protein-RNA interactions from truncation-based CLIP data. For this purpose we model both broader binding footprints and, at high resolution, the signals caused by direct interactions between the protein and individual nucleotides. In order to accomplish this, we address two objectives: (1) detect regions enriched in mapped reads caused by pulled-down RNA fragments and (2) detect crosslink sites where a significant fraction of read starts accumulates at the same position, originating from truncated cDNAs (see Figure 5.1a).

The output of PureCLIP consists of individual crosslink sites associated with a score. Since multiple crosslink sites can occur within one binding region, the crosslink sites are optionally merged. In Section 5.3 we will first describe the basic model of PureCLIP, i.e. without correction for biases. In Section 5.4 we will then describe how this basic

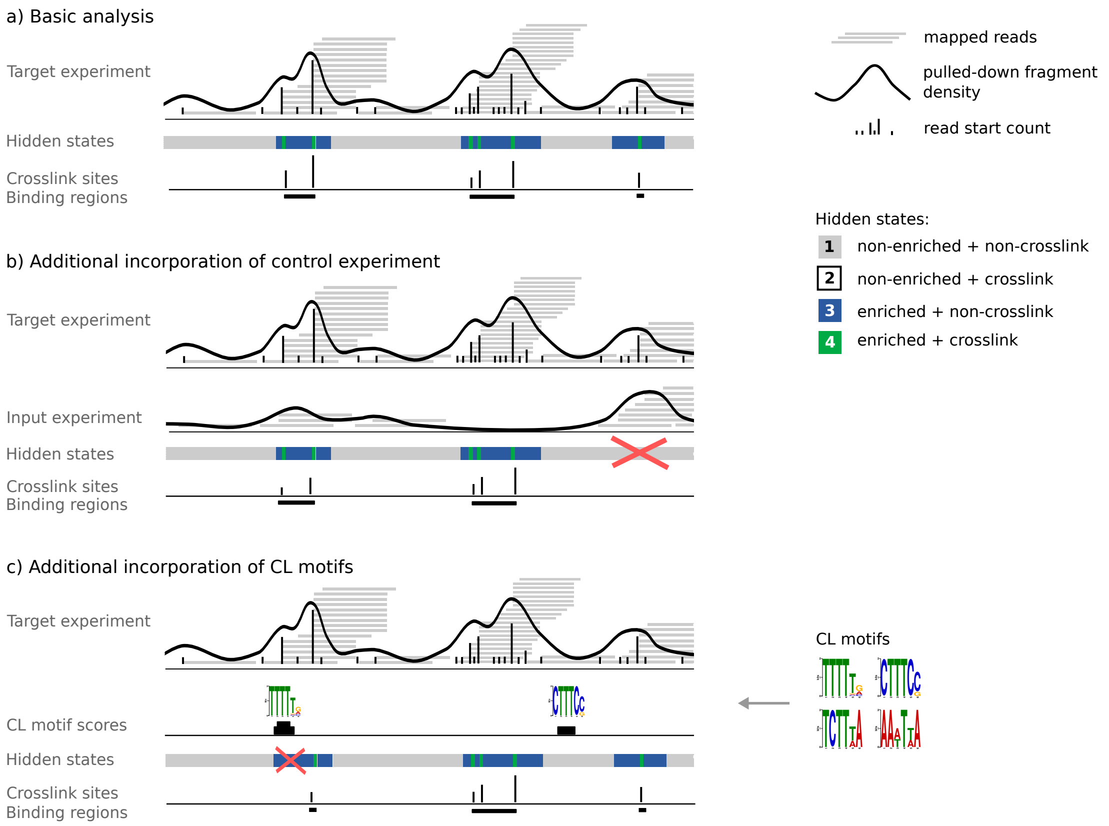


Figure 5.1.: Overview of the PureCLIP approach. a) PureCLIP starts with mapped reads from a target truncation-based CLIP experiment and derives two signals: the pulled-down fragment densities and individual read start counts. Based on these two observed signals it infers the most likely hidden state for each position. The goal is to identify all sites with an *enriched + crosslink* state. Individual crosslink sites can then be merged to binding regions. **b)** Additionally, information from input control experiments can be incorporated. Its fragment densities are used to correct for non-specific background signals, which reduces the number of false calls. **c)** PureCLIP can incorporate information about CL-motifs to reduce false calls caused by non-specific crosslinks.

model can be extended to incorporate additional factors into the model, such as a non-specific background signal from input experiments (see Figure 5.1b) and CL-motifs (see Figure 5.1c), to correct for biases. And finally, in Section 5.5 we present an extension to include multiple individual replicates.

5.3. PureCLIP hidden Markov model

CLIP data features a spatial dependency between neighbouring positions. In order to infer crosslink sites from the observed data we look at it as a segmentation problem and model this using a hidden Markov model (HMM) (see Section 4.2.5) at single-nucleotide resolution. In this section we will first briefly present the design of the HMM and the used signals, before we will describe PureCLIP's statistical concepts in detail in the following sections.

Hidden states

Each position t can be categorized either as *non-enriched* or *enriched*, indicating whether the position is enriched in protein bound fragments or not. In addition, each position can also be categorized as *non-crosslink* or *crosslink*, indicating whether it represents a crosslink site or not. This combination results in four hidden states (see Figure 5.1):

(1) *non-enriched + non-crosslink*,

(2) *non-enriched + crosslink*,

(3) *enriched + non-crosslink*,

(4) *enriched + crosslink*.

Non-enriched sites correspond to regions with no or low signal, which is assumed to be background noise. State (2) corresponds to non-specific crosslink sites and it is included in the model for mathematical completeness. We are interested in all sites with a hidden state (4), i.e. sites that are enriched in pulled down RNA fragments and show the truncation pattern (see Figure 5.1a). For the sake of clarity, we separate the hidden states into two state variables (see Figure 5.2): $S_t^{(1)} = z_i^{(1)}$ represents the enrichment state at position t with

$$z_i^{(1)} = \begin{cases} \textit{non-enriched}, & i = 0 \\ \textit{enriched}, & i = 1 \end{cases} \tag{5.1}$$

and $S_t^{(2)} = z_j^{(2)}$ represents the crosslink state with

$$z_j^{(2)} = \begin{cases} \textit{non-crosslink}, & j = 0 \\ \textit{crosslink}, & j = 1. \end{cases} \tag{5.2}$$

Observations

In order to detect *enriched + crosslinked* sites, PureCLIP uses two signals derived from the mapped reads: (1) the *pulled-down fragment density* C_t, which is a smoothed signal derived from the read start counts and holds information about the enrichment within the current region, and (2) the read start counts K_t themselves, which hold information about potential truncation events. Consequently, the HMM has two layers of observations: $\boldsymbol{c}$ and $\boldsymbol{k}$ (see Figure 5.2).

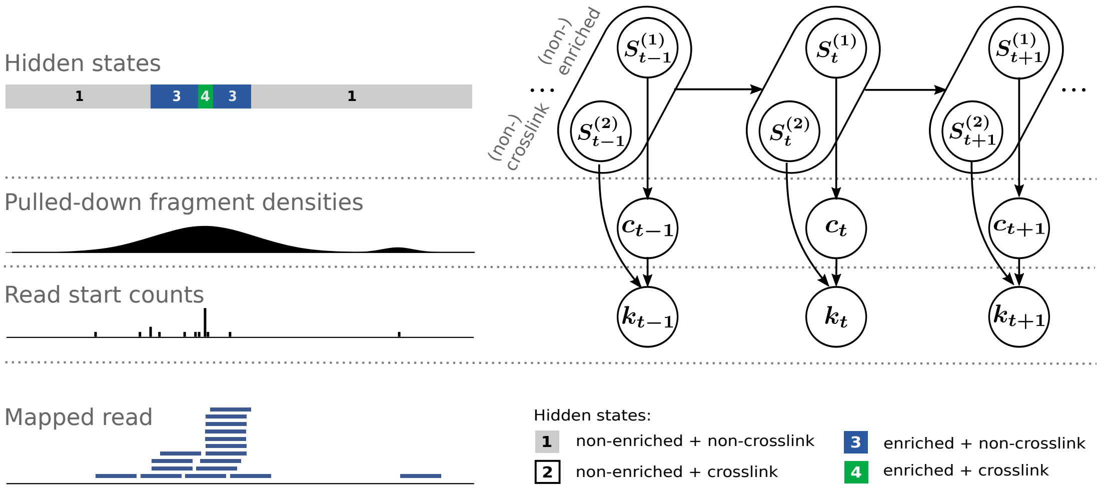


Figure 5.2.: Summary of the basic HMM framework. Left: Starting from the mapped reads (bottom), two signals that serve as observations in the HMM are derived for all nucleotide positions: individual read start counts and pulled-down fragment densities, obtained from smoothed read start counts. The model aims to reconstruct the most likely sequence of hidden states (top) from these signals. Right: A graphical representation of the corresponding HMM.

To estimate the pulled-down fragment density we do not use position-wise read counts, since for truncation based CLIP data these are strongly influenced by crosslinking events in the neighbourhood. On the other hand, using counts within larger bins would not be accurate in estimating the position-wise signal of the pulled-down fragments. To address this problem, we apply a smoothing on the read start counts $\boldsymbol{k}$ to estimate the density of pulled-down fragments at each position. This is done using a *kernel density estimation (KDE)* [43] with a Gaussian kernel function $\mathcal{K}$. For each position, the latter assigns a higher weight to nearby read starts, while still considering read starts which are further away, thereby providing a better estimate for the underlying pulled-down fragment density. We compute the smoothed signal at position t using

$$c_t = \frac{1}{h} \sum_{i=t-4h}^{t+4h} k_i \cdot \mathcal{K}\left(\frac{t-i}{h}\right), \tag{5.3}$$

where h is the kernel bandwidth (default: 50) and only positions within $4h$ before and after position t are considered to limit the computational complexity.

We constrain the HMM to *covered regions*. A covered region comprises one or more read start sites which are at most $4h$ nt apart as well as the $2h$ positions flanking the outer read start sites. Individual reads with no other read starting within $4h$ nt, referred to as *singleton* reads, are discarded.

Emission probabilities

We assume that the observed pulled-down fragment densities can be described with a mixture model, where one component is generated by *non-enriched* sites and one by *enriched* sites. Two gamma distributions are then used to model the fragment density values, with one set of parameters for the *non-enriched* state and one for the *enriched* state, assuming that the *enriched* state is more likely to cause high fragment density values than the *non-enriched* state.

Similarly, read start counts are modelled using two binomial distributions under the assumption that the *crosslink* state is more likely to generate a higher fraction of reads starting at one position than the *non-crosslink* state. In order to account for differently covered regions we exploit the hierarchical structure of the two observed signals to specify the model (see Figure 5.2), i.e. the size parameter of the binomial distributions depends on the pulled-down fragment density c_t.

The fragment density distributions and the read start count distributions are combined to obtain the emission probabilities for each of the four hidden states.

Goal and general remarks

For each position we can then address the question: which of the four hidden states did most likely cause the observed data? Our goal is to identify positions that are *enriched + crosslinked* (see state (4) in Figure 5.2), which are interpreted as target-specific interactions. Transitions between all four states are allowed and their probabilities are assumed to be homogeneous over the transcriptome [1]. The model parameters of the HMM are estimated using the Baum-Welch algorithm, as described in detail in Section 4.2.5, but with numerous model specific modifications. The latter mainly affect the learning of the emission probability parameters for the different states, while initial and transition probability parameters are computed as described in Equation 4.29 and 4.30, respectively. In the following sections we will describe in detail how we model the emission probability distributions for each state and how the corresponding parameters are learned.

5.3.1. Emission probabilities for *non-enriched* and *enriched* states

First, in order to distinguish between *non-enriched* ($z_0^{(1)}$) and *enriched* sites ($z_1^{(1)}$) we model the corresponding distributions of the fragment densities C_t.

Left-truncated gamma distributed emission probabilities

The fragment densities are non-negative continuous values with a right skewed distribution, which can be approximately described by a gamma distribution (see Ap-

[1] In practice the hidden states are not equally likely across the transcriptome, for example due to different sequence preferences of the target protein, however, we focus here on the correction of biases and not on the modelling of binding preferences.

pendix A.2). Furthermore, we do not want to fit the model to the large portion of sites which have a very low density or no read start to improve both the efficiency and the robustness of the model. Accordingly, for each state $z_i^{(1)}$ we model the emission probability distribution using a *left-truncated gamma (LTG)* distribution

$$P(C_t = c_t \mid S_t^{(1)} = z_i^{(1)}) = \begin{cases} f_{LTG}(c_t; \mu_i, \lambda_i, \tau) & \text{if } c_t > \tau \\ 1 & \text{if } c_t \leq \tau \land i = 0 \\ 0 & \text{if } c_t \leq \tau \land i = 1 \end{cases} \tag{5.4}$$

where μ_i and λ_i denote the mean and the shape parameter of the distribution and τ the truncation point. Fragment densities $c_t \leq \tau$ can only be emitted by *non-enriched* states. We set the truncation point τ to the fragment density value of a singleton read start (see Equation 5.3):

$$\tau = \frac{1}{h} \cdot \mathcal{K}\left(\frac{0}{h}\right), \tag{5.5}$$

since we only want to account for positions with at least one read start and additionally account for the fact that singleton read starts are discarded (see Section 5.3).

The probability density function of the LTG distribution is defined as

$$f_{LTG}(c_t; \mu_i, \lambda_i, \tau) = \begin{cases} 0 & \text{if } c_t \leq \tau \\ \frac{1}{1-F_G(\tau;\mu_i,\lambda_i)} \cdot \frac{c_t^{\lambda_i - 1} e^{-\frac{\lambda_i c_t}{\mu_i}}}{\left(\frac{\mu_i}{\lambda_i}\right)^{\lambda_i} \Gamma(\lambda_i)} & \text{if } c_t > \tau \end{cases} \tag{5.6}$$

with $\mu_i, \lambda_i > 0$. For $c_t > \tau$, the first term normalizes the distribution for the truncation (see Section 4.2.4), while the second term denotes the general gamma probability density function (see Section 4.2.2). $F_G(\tau; \mu_i, \lambda_i)$ denotes the cumulative gamma distribution function, which can be shown to be equivalent to the normalized lower incomplete gamma function for the integral from 0 to $\left(\frac{\lambda_i \tau}{\mu_i}\right)$:

$$\begin{aligned} F_G(\tau; \mu_i, \lambda_i) = \int_0^\tau f_G(x; \mu_i, \lambda_i) \,\mathrm{d}x &= \frac{1}{\Gamma(\lambda_i)} \int_0^\tau \left(\frac{\lambda_i}{\mu_i}\right)^{\lambda_i} x^{\lambda_i - 1} e^{-\frac{\lambda_i x}{\mu_i}} dx \\ &= \frac{1}{\Gamma(\lambda_i)} \int_0^{\left(\frac{\lambda_i \tau}{\mu_i}\right)} x^{\lambda_i - 1} e^{-x} dx \\ &= \frac{\gamma\left(\lambda_i, \frac{\lambda_i \tau}{\mu_i}\right)}{\Gamma(\lambda_i)}. \end{aligned} \tag{5.7}$$

While τ is fixed, the parameters μ_i and λ_i need to be learned.

Parameter estimation

We will now describe how the gamma parameters that maximize the expected log-likelihood of the data are estimated within each iteration of the Baum-Welch algorithm (see Section 4.2.5, maximization step) for the *non-enriched* and the *enriched* state. While the mean parameter for conventional gamma distributions can be estimated using a closed formula, this does not hold for the computation of the mean μ or shape λ parameter of the truncated gamma distribution. We therefore estimate the parameters that maximize the log-likelihood function weighted by the corresponding state posterior probabilities numerically.

In general, for given observations $\boldsymbol{y} = y_1, \ldots, y_T$ the log-likelihood function for the LTG distribution (see Equation 5.6) can be written as:

$$\begin{aligned}\ln \mathcal{L}_{LTG}(\mu, \lambda \mid \boldsymbol{Y} = \boldsymbol{y}, \tau) = &\sum_{t=1}^{T} \left[(\lambda - 1)\ln(y_t) - \frac{\lambda y_t}{\mu} - \lambda \ln\left(\frac{\mu}{\lambda}\right) - \ln(\Gamma(\lambda)) \right. \\ &\left. - \ln\left(1 - \frac{\gamma\left(\lambda, \frac{\lambda\tau}{\mu}\right)}{\Gamma(\lambda)}\right)\right]. \end{aligned} \tag{5.8}$$

We compute the position-wise state posterior probabilities

$$\gamma_{t,z_i^{(1)}} = P(S_t^{(1)} = z_i^{(1)} \mid \boldsymbol{C} = \boldsymbol{c}, \theta'), \qquad i \in \{0, 1\}, \tag{5.9}$$

using the previous model parameters θ' (see Equation 4.23). For each state $z_i^{(1)}$ we then estimate the updated parameters μ_i'' and λ_i'' as (see Equation 4.31):

$$\begin{aligned}(\mu_i'', \lambda_i'') = \arg\max_{\mu,\lambda} \sum_{t=1}^{T} \gamma_{t,z_i^{(1)}} \cdot &\left[(\lambda - 1)\ln(c_t) - \frac{\lambda c_t}{\mu} - \lambda \ln\left(\frac{\mu}{\lambda}\right) - \ln(\Gamma(\lambda)) \right. \\ &\left. - \ln\left(1 - \frac{\gamma\left(\lambda, \frac{\lambda\tau}{\mu}\right)}{\Gamma(\lambda)}\right)\right] \cdot I_t^{(1)}, \end{aligned} \tag{5.10}$$

where $I_t^{(1)}$ is an indicator variable defined as

$$I_t^{(1)} = \begin{cases} 1 & \text{if } k_t \geq 1 \\ 0 & \text{else.} \end{cases}$$

The latter ensures that the gamma distributions are only fitted to sites with at least one read start, which reduces the computational costs and additionally improves PureCLIP's robustness. The effect of this model choice on the performance will be shown in Chapter 9.

To numerically estimate the parameters that maximize a MLE, many different methods exist (see Section 4.2.8). However, gradient-based methods cannot be applied here, because the derivatives of the truncated gamma distribution with respect to λ_i become very complex. We therefore resorted to the Nelder-Mead optimization method, which requires only the values of the likelihood function itself (see Section 4.2.8), and is implemented in the GNU Scientific Library (GSL) [49]. The parameters μ_i' and λ_i'

learned in the previous Baum-Welch iteration are used as starting values, i.e. as one of the three starting vertices of a 2-simplex. They are likely to be already relatively close to the current optimum and thus lead to a relatively fast convergence. Moreover, for our optimization problem the parameter spaces are constrained. To ensure positive μ parameters, the corresponding optimizations are performed in log-space. Additionally, to enable arbitrary constraints for the shape parameter λ, the Nelder-Mead algorithm was extended by penalty-based soft constraints (see Appendix A.4).

Estimating the mean and shape parameters of a truncated gamma distribution tends to be relatively unstable. To regularize the estimates, by default, the shape parameter of the *non-enriched* state is set to $\lambda_0 = 1$, which constrains the emission probability distribution to an exponential shape. Additionally, the shape parameter of the *enriched* state is constrained with $\lambda_1 \geq 1$.

5.3.2. Emission probabilities for *non-crosslink* and *crosslink* states

To distinguish between *non-crosslink* ($z_0^{(2)}$) and *crosslink* sites ($z_1^{(2)}$) we model the corresponding read start counts K_t at position t, where we expect an increased count at *crosslink* sites due to the underlying truncation events.

Zero-truncated binomially distributed emission probabilities

The probability to observe k_t read starts can be modeled with a binomial distribution given a number of trials n_t and a probability p for each read to start at position t. Here, n_t denotes the number of fragments (or trials) from which a certain fraction results in reads starting at position t. n_t is unknown, however, we can use a surrogate value $\hat{n}_t$ directly deduced from the pulled-down fragment density c_t by a simple rescaling as described in the next paragraph. Furthermore, we again do not want to fit the distributions to the large number of sites with no read starting. Accordingly, for each state $z_j^{(2)}$ we model the emission probability distribution using a *zero-truncated binomial (ZTB)* distribution

$$P(K_t = k_t \mid C_t = c_t, S_t^{(2)} = z_j^{(2)}) = \begin{cases} f_{ZTB}(k_t; \hat{n}_t, p_j) & \text{if } k_t \geq 1 \\ 1 & \text{if } k_t = 0 \wedge j = 0 \\ 0 & \text{if } k_t = 0 \wedge j = 1. \end{cases} \tag{5.11}$$

Read start counts $k_t = 0$ can only be emitted from *non-crosslink* states. The probability density function is defined as

$$f_{ZTB}(k_t; \hat{n}_t, p_j) = \begin{cases} 0 & \text{if } k_t = 0 \\ \frac{1}{1-F_B(0;\hat{n}_t,p_j)} \binom{\hat{n}_t}{k_t} p_j^{k_t} (1-p_j)^{\hat{n}_t - k_t} & \text{if } k_t \geq 1, \end{cases} \tag{5.12}$$

where for $k_t \geq 1$ the first term normalizes the distribution for the zero-truncation with

$$F_B(0; \hat{n}_t, p_j) = (1-p_j)^{\hat{n}_t}. \tag{5.13}$$

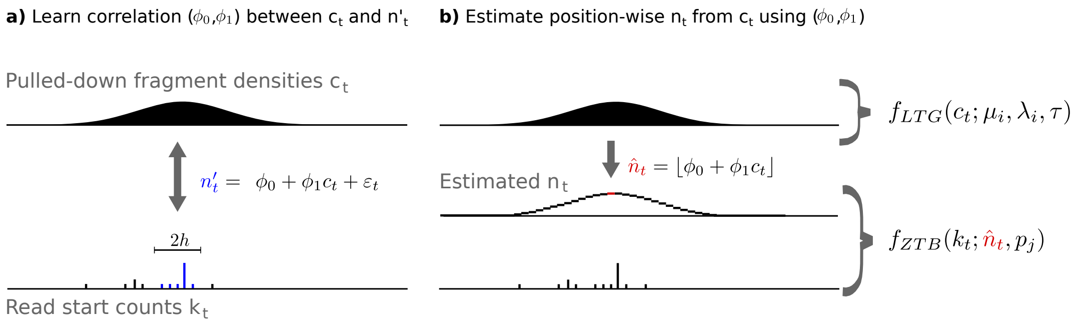

a) Learn correlation (ϕ_0,ϕ_1) between c_t and n'_t
b) Estimate position-wise n_t from c_t using (ϕ_0,ϕ_1)
Pulled-down fragment densities c_t
$n'_t = \phi_0 + \phi_1 c_t + \varepsilon_t$
$2h$
Read start counts k_t
Estimated n_t
$\hat{n}_t = \lfloor \phi_0 + \phi_1 c_t \rfloor$
$f_{LTG}(c_t; \mu_i, \lambda_i, \tau)$
$f_{ZTB}(k_t; \hat{n}_t, p_j)$

Figure 5.3.: Estimation of n_t, i.e. the number of trials or fragments from which a certain fraction results in read starts at position t, for the zero-truncated binomial (ZTB) distribution: Instead of using the read start count within a spanning window we estimate n_t based on the fragment density c_t given the learned regression coefficients (ϕ_0, ϕ_1).

The probability parameters p_0 and p_1 need to be learned, where p_1 reflects a protein specific truncation rate at *crosslink* sites. More precisely, for *crosslink* states p_1 reflects the read start rate arising from truncated and non-truncated cDNAs, while for *non-crosslink* states p_0 reflects the read start rate arising from non-truncated (or off-target truncated) cDNAs.

Estimation of the binomial n_t parameter

To model the read start counts using a binomial distribution we need the number of trials or pulled-down fragments n_t for each position t that can potentially generate a read start at this position, originating either from a truncated or non-truncated cDNA. A reasonable estimate $\hat{n}_t$ is crucial to ensure accurate estimates of p_j. We therefore take advantage of the already estimated pulled-down fragment density c_t. More precisely, we expect a linear relationship between the observed read start counts n'_t within a spanning window and c_t, which we can model using a linear regression:

$$n'_t = \phi_0 + \phi_1 c_t + \varepsilon_t. \tag{5.14}$$

We use windows of size $2h$ to compute n'_t, estimate the regression coefficients ϕ_0 and ϕ_1 and then use them to predict $\hat{n}_t$ for each position (see Figure 5.3). This estimate of n_t has the same advantages over bin-wise read start counts as the estimated pulled-down fragment density due to the Gaussian kernel function, which weights read starts within close proximity higher than read starts further away (see Section 5.3), and thus provides a more accurate estimate of the position-wise fragment coverage.

Parameter estimation

Since there exists no closed formula to compute the probability parameter p for the ZTB distribution (see Equation 5.12), we again use numerical optimization to estimate the parameters that maximize the log-likelihood function weighted by the corresponding state posterior probabilities [2]. Let $\boldsymbol{y} = y_1, \ldots, y_T$ be observed numbers of successes obtained from T zero-truncated binomial experiments with a common probability parameter p and with individual numbers of trials $\boldsymbol{n} = n_1, \ldots, n_T$, then the log-likelihood function can be written as:

$$\ln \mathcal{L}_{ZTB}(p \mid \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{n}) = \\ \sum_{t=1}^{T} \left[-\ln\left(1 - (1-p)^{n_t}\right) + \ln \binom{n_t}{y_t} + y_t \ln(p) + (n_t - y_t)\ln(1-p) \right]. \tag{5.15}$$

Note that for the likelihood maximization the term $\ln \binom{n_t}{y_t}$ can be omitted, since it does not depend on the probability parameter.

To weight the *non-crosslink* and *crosslink* log-likelihood functions, instead of using the posterior probabilities $\gamma_{t,z_j^{(2)}}$, we use the state posterior probabilities

$$\gamma_{t,z_1^{(1)}z_j^{(2)}} = P(S_t^{(1)} = z_1^{(1)}, S_t^{(2)} = z_j^{(2)} \mid \boldsymbol{K} = \boldsymbol{k}, \boldsymbol{n}, \theta'), \qquad \forall t \in \{1, \ldots, T\}, \tag{5.16}$$

i.e. the probability for a position t to be *enriched* and in state $z_j^{(2)}$. With this we aim to reduce the impact of regions containing background noise on the binomial parameter estimation. Then, for each state $z_j^{(2)}$ the new parameter p_j'' is estimated as:

$$p_j'' = \arg\max_p \sum_{t=1}^{T} \gamma_{t,z_1^{(1)}z_j^{(2)}} \cdot \left[-\ln\left(1 - (1-p)^{\hat{n}_t}\right) + k_t \ln(p) + (\hat{n}_t - k_t)\ln(1-p) \right] \\ \cdot I_t^{(2)}, \qquad j \in \{0, 1\}, \tag{5.17}$$

where $I_t^{(2)}$ is an indicator variable defined as

$$I_t^{(2)} = \begin{cases} 1 & \text{if } (k_t \geq 1) \wedge (\hat{n}_t \geq n_c) \\ 0 & \text{else.} \end{cases}$$

With $I_t^{(2)}$ we limit the parameter estimation to sites with at least one read start and with $\hat{n}_t \geq n_c$, a given threshold (default: n_c = 10). The latter is important because the largest fraction of read start counts occurs at positions with low $\hat{n}_t$, where the distributions for the *non-crosslink* and for the *crosslink* state are not well distinguishable.

In contrast to estimating the parameters for the *non-enriched* and *enriched* gamma distributions, we now need to learn only one parameter for each state. For this the implementation of Brent's method [16] in the Boost library [123] is used. Brent's method is a fast and robust, one-dimensional, derivative-free optimization algorithm that combines quadratic interpolation with a bisection method.

[2] Note that multiple explicit estimators for p exist [30], however, we opt for accuracy rather than computational efficiency.

5.3.3. Joint emission probabilities

In the previous sections we have seen how the fragment density (C_t) emission probabilities can be computed for the *non-enriched* and *enriched* states. Additionally, we have seen how the read start count (K_t) emission probabilities can be computed for the *non-crosslink* and *crosslink* states. We now combine them to compute the probabilities of joint observations, i.e. the emission probabilities of the four hidden states, as described in Section 5.3. Note that C_t and K_t are not conditionally independent, but since $\hat{n}_t$ is directly deduced from c_t the emission probability for the joint observation can be factorized accordingly (see Figure 5.2 for a graphical summary):

$$\begin{aligned} & P(C_t = c_t, K_t = k_t \mid S_t^{(1)} = z_i^{(1)}, S_t^{(2)} = z_j^{(2)}) \\ & = P(C_t = c_t \mid S_t^{(1)} = z_i^{(1)}) \cdot P(K_t = k_t \mid C_t = c_t, S_t^{(2)} = z_j^{(2)}) \\ & = f_{LTG}(c_t; \mu_i, \lambda_i, \tau) \cdot f_{ZTB}(k_t; \hat{n}_t, p_j). \end{aligned} \tag{5.18}$$

5.3.4. Initialization

Because the Baum-Welch algorithm might converge to local optima when using poor initializations, we choose initial parameters using a preprocessing step.

For the two LTG distributions, we start with a user defined discrete enrichment threshold e (default: 7), to compute a fragment density threshold c_e:

$$c_e = \frac{e}{h} \cdot K\left(\frac{0}{h}\right). \tag{5.19}$$

All sites with a fragment density $c_t \geq c_e$ will be defined as most likely *enriched*. More precisely, we set the *enriched* state posterior probability of those sites to 0.999 and the *non-enriched* state posterior probability to 0.001 (and vice versa for all other sites). Next, we estimate the gamma parameters using a weighted MLE as described in Section 5.3.1. Since the Nelder-Mead algorithm does not guarantee to find the global optimum (see Section 4.2.8), we restart the algorithm from different starting values and chose the parameters generating the highest log-likelihood value. However, in practice the found optima usually do not differ.

For the ZTB distributions, since only one parameter has to be estimated, their final estimation is less dependent on the initialization and we thus use predefined values (*non-crosslink* state: $p_0 = 0.01$; *crosslink* state: $p_1 = 0.15$).

5.3.5. Modified Baum-Welch algorithm

Transition, initial and emission probability parameters of the HMM are estimated using the Baum-Welch algorithm (see Section 4.2.5). We proceed in sequential Baum-Welch blocks to learn the emission probability distribution parameters of the two types of observations, namely the parameters of the LTG distributions modelling the observed fragment density C_t for each enrichment state $z_i^{(1)}$ (see Section 5.3.1), and the parameters of the ZTB distributions modelling the number of read starts K_t for each crosslinking state $z_j^{(2)}$ (see Section 5.3.2). First, the LTG parameters are learned with fixed initial ZTB parameters. Then, the ZTB parameters are learned using the preliminary LTG parameters before another Baum-Welch block is applied for a final update of the LTG parameters. Within each Baum-Welch iteration it is ensured that the corresponding emission probability distributions assigned to the states are not swapped, i.e. for the LTG parameters $\mu_0 \leq \mu_1$ and for the ZTB parameters $p_0 \leq p_1$. Transition and initial probabilities are updated in each Baum-Welch iteration.

As a convergence criterion the change of the overall likelihood of the model as provided by the forward algorithm $\sum_{i=1}^{\ell} \alpha_i(T)$ (see Equation 4.19) cannot be used here. The reason for this is that not all positions, which are considered for the forward algorithm with parameter-dependent emission probabilities, are used for the estimation of the emission probability parameters [3]. Therefore, within each block, the iterations are terminated when for all parameters the change is below a certain threshold or a maximum number of iterations is reached.

5.3.6. Inference and scoring of crosslink sites and binding regions

Finally, we use posterior decoding as described in Section 4.2.5 to determine the most likely hidden state for each position, and with that all *enriched + crosslink* sites. These are the sites of interest as they are interpreted as target-specific protein-RNA interaction sites. In a second step, the called crosslink sites are further combined to binding regions based on their distance. By default, called crosslink sites that are not more than 8 nt far away are merged.

To rank the called crosslink sites, we compute a score for each site. For this purpose, we implemented four alternative scoring schemes. First, without any prior emphasis we define the *unconditional posterior-ratio score* as

$$\text{score}_{UC}(t) = \ln\left(\frac{P(S_t = \textit{enriched + crosslink} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}{P(S_t = \textit{2nd most likely state} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}\right), \tag{5.20}$$

i.e. the log posterior probability ratio of the first and second most likely state, given the observed fragment densities $\boldsymbol{c}$ and read start counts $\boldsymbol{k}$. The idea behind this approach is to score less ambiguous state assignments higher than more ambiguous ones. The second likely hidden state varies for each called crosslink site and the question remains if there exist more optimal scoring strategies.

Since PureCLIP computes the state posterior probabilities given two observed signals, it could be that one is more important than the other. We therefore additionally define the *enrichment focused posterior-ratio score* as

$$\text{score}_{E}(t) = \ln\left(\frac{P(S_t = \textit{enriched + crosslink} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}{P(S_t = \textit{non-enriched + crosslink} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}\right) \tag{5.21}$$

[3] For example, positions that are located within a covered region and have a fragment density $c_t > \tau$, but contain no read start, are not used for the learning of the emission probability parameters. Nevertheless, such positions have assigned *non-enriched* and *enriched* emission probabilities depending on the learned parameters and remain in the model.

and the *crosslink focused posterior-ratio score* as

$$\text{score}_{CL}(t) = \ln\left(\frac{P(S_t = \textit{enriched + crosslink} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}{P(S_t = \textit{enriched + non-crosslink} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}\right), \tag{5.22}$$

in order to alternatively score the confidence that a site is enriched or crosslinked, respectively. Lastly, we define

$$\begin{aligned}\text{score}_B(t) =& \ln\left(\frac{P(S_t^{(1)} = \textit{enriched} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}{P(S_t^{(1)} = \textit{non-enriched} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}\right)\\ &+ \ln\left(\frac{P(S_t^{(2)} = \textit{crosslink} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}{P(S_t^{(2)} = \textit{non-crosslink} \mid \boldsymbol{C} = \boldsymbol{c}, \boldsymbol{K} = \boldsymbol{k})}\right),\end{aligned} \tag{5.23}$$

as the *balanced posterior-ratio score*. By default PureCLIP ranks crosslink sites based on the unconditional score as defined in Equation 5.20.

When merging crosslink sites to binding regions, the sum of the individual scores is used as a region-wise score.

5.4. PureCLIP non-homogeneous hidden Markov model

In the previous section we have described the basic model of PureCLIP to infer target-specific protein-RNA interaction sites from iCLIP or eCLIP data. However, the observed signals are often biased by numerous factors such as transcript abundances, binding of background proteins or sequence biases during the crosslink formation. Fortunately, often additional knowledge in the form of auxiliary data or known characteristics of a particular bias is available and can be used to improve the inference process. In this section, we will describe an extension of PureCLIP that allows for the incorporation of such information. For this purpose, the underlying HMM is extended to a non-homogeneous HMM, which allows for non-homogeneous emission probabilities across different positions. To model the influence of the covariates on the emission probabilities, we use generalized linear models (GLMs). According to the two types of observed signals used in our model, biases can be modelled in two different ways, either as an influence on the fragment density emissions or on the read start count emissions. While this extension of the model allows for the incorporation of various types of covariates, here we will focus on the following two: data from input control experiments to factor in non-specific background signal and sequence motifs that are know to preferentially cause the formation of crosslinks. Note that in contrast to the emission probabilities, transition probabilities remain to be position independent.

5.4.1. Incorporation of input control data

We expect positions within highly abundant RNAs to show a higher pulled-down fragment density than others, both in target binding regions and in regions containing

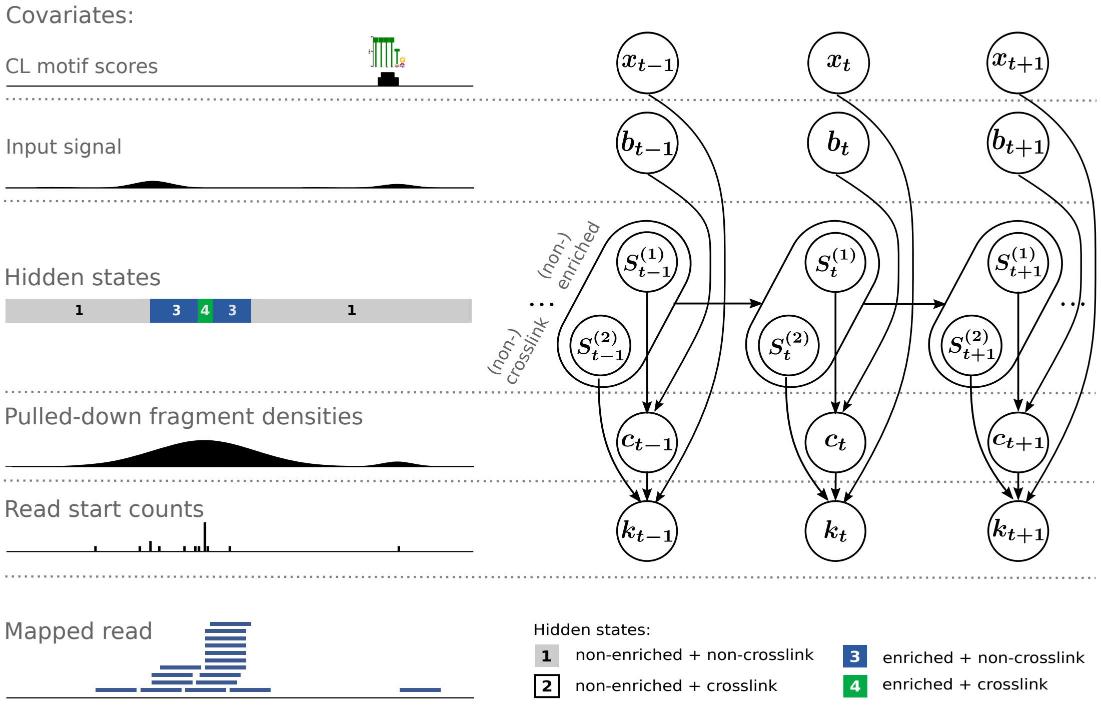


Figure 5.4.: Summary of the non-homogeneous HMM framework. Left: Starting from mapped reads (bottom), observations are deduced and modeled using additional covariates (top) to reconstruct the most likely sequence of hidden states (middle). Right: Graphical representation of the corresponding non-homogeneous HMM.

only non-specific background noise. Moreover, regions bound by highly abundant background proteins tend to contain also higher signals of non-specific background binding. To normalize for this and potentially other signal components that are not target specific, information from input control experiments can be included to model the emission probability distributions of the *non-enriched* and *enriched* states. With this, we aim to reduce the number of false positives caused by non-specific background signal (see Figure 5.1b) while increasing the sensitivity for regions with low input signal.

As described in Section 5.3.1, we assume that the fragment density C_t follows a left-truncated gamma distribution

$$C_t \mid S_t^{(1)} = z_i^{(1)} \sim LTG(\mu_i, \lambda_i, \tau), \qquad i \in \{0, 1\}, \tag{5.24}$$

for each enrichment state $z_i^{(1)}$, where τ denotes the truncation point. If a non-specific background signal is available, e.g. from an input control experiment, we can incorporate this as position-wise covariates into the model. Figure 5.4 illustrates the extended PureCLIP HMM framework.

A gamma generalized linear model for emission probabilities

This is done using a (left-truncated) gamma generalized linear model (GLM). The objective is to learn the correlation between the covariate b and the mean parameter μ_i of each enrichment state $z_i^{(1)}$. For this purpose, the underlying multiplicative effect of the background signal b_t at position t on the expected mean $\mu_{i,t}$ is modeled using a log link function:

$$\ln(\mu_{i,t}) = \alpha_{i,0} + \alpha_{i,1} b_t. \tag{5.25}$$

Note that for each enrichment state $z_i^{(1)}$ we assume to have a constant shape λ_i across the entire range of covariate values.

By default, PureCLIP uses the log fragment densities of the input experiment as covariates b_t, computed using a KDE with the same bandwidth as for the target fragment densities.

Parameter estimation

Similar to the basic model (see Section 5.3.1), the parameters that maximize the weighted MLE are estimated for each state $z_i^{(1)}$ using the Nelder-Mead algorithm, but instead of μ_i'' here $\alpha_{i,0}''$ and $\alpha_{i,1}''$ are estimated as follows:

$$\begin{aligned}(\alpha_{i,0}'', \alpha_{i,1}'', \lambda_i'') = & \arg \max_{\alpha_0, \alpha_1, \lambda} \sum_{t=1}^{T} \gamma_{ti} \\ & \cdot \left[(\lambda - 1)\ln(c_t) - \frac{\lambda c_t}{\exp(\alpha_0 + \alpha_1 b_t)} + \lambda \ln\left(\frac{\exp(\alpha_0 + \alpha_1 b_t)}{\lambda}\right) \right. \\ & \left. - \ln(\Gamma(\lambda)) - \ln\left(1 - \frac{\gamma\left(\lambda, \frac{\lambda\tau}{\exp(\alpha_0+\alpha_1 b_t)}\right)}{\Gamma(\lambda)}\right)\right] \cdot I_t^{(1)},\end{aligned}$$

where $I_t^{(1)}$ is defined as

$$I_t^{(1)} = \begin{cases} 1 & \text{if } (k_t \geq 1) \land (b_t > \tau) \\ 0 & \text{else.} \end{cases}$$

Note that the indicator variable $I_t^{(1)}$ limits the parameter learning, not only to sites with at least one read start, but also to sites with $b_t > \tau$. This prevents the learned regression coefficients α from being impaired by a large number of sites that have a very low fragment density as well as a lower correlation between the target and the input signal. In contrast to the basic model, the shape parameter of the *non-enriched* state is no longer constrained to $\lambda_0 = 1$. However, both shape parameters λ_0 and λ_1 are constrained with $\lambda_0, \lambda_1 \geq 1$.

Initialization

To ensure a good initialization and to prevent the Baum-Welch algorithm from converging to local optima, instead of using one enrichment threshold to assign initial state posterior probabilities as described in Section 5.3.4, we now use a threshold dependent on the background signal b_t. For that, we first learn the parameters α_0, α_1 and λ of the compound fragment density distribution, i.e. the mixture of the *non-enriched* and the *enriched* distributions. The Nelder-Mead algorithm is restarted from different starting values and the parameters maximizing the corresponding log-likelihood function are chosen. Then all sites with a fragment density $c_t \geq \exp(\alpha_0 + \alpha_1 b_t)$ are initially defined as most likely *enriched*. More precisely, we set the posterior probability of those sites to 0.9 for the *enriched* state and to 0.1 for the *non-enriched* state (and vice versa for all other sites). Next, the initial parameters for the *non-enriched* and *enriched* gamma GLMs are estimated using a weighted MLE as described in Section 5.4.1, once again using different starting values.

5.4.2. Incorporation of CL-motifs

Concerning the emission of read start counts, we expect a higher signal at positions within CL-motifs (see Section 3.3.4). Thus, to correct for this sequence bias, information about CL-motifs can be incorporated (see Figure 5.1c) to model the *non-crosslink* and *crosslink* emission distributions.

As described in Section 5.3.2, the read start counts K_t are modeled using a zero-truncated binomial distribution

$$K_t \mid S_t^{(2)} = z_j^{(2)} \sim ZTB(\hat{n}_t, p_j), \qquad j \in \{0, 1\}. \tag{5.26}$$

If position-wise information about the crosslinking bias is available, we can again incorporate this into the model. Assuming that we have given q CL-motifs, we can compute a corresponding motif match score $x_{m,t} \geq 0$ for each position t and motif $m \in \{1, \ldots, q\}$, representing the crosslinking affinity at each position. These scores are then used as covariates to model the influence on the emission probabilities (see Figure 5.4).

A binomial logistic regression model for emission probabilities

We use a (zero-truncated) binomial logistic regression for each crosslinking state $z_j^{(2)}$ to model the expected binomial probability parameter $p_{j,t}$ based on the position-wise CL-motif score $x_{m_t^*,t}$ as follows:

$$\ln \frac{p_{j,t}}{1 - p_{j,t}} = \beta_{j,0} + \beta_{j,m_t^*}\, x_{m_t^*,t}, \qquad m_t^* = \arg \max_{m \in 1,\ldots,q} x_{m,t}, \quad x_{m,t} \geq 0. \tag{5.27}$$

For simplicity, we assume that each position matches at most one CL-motif, i.e. for each position t we chose the motif m_t^* with the highest motif match score $x_{m_t^*,t}$ and use a motif specific regression coefficient β_{j,m_t^*} that has to be learned.

Computation of CL-motif scores

To obtain position-wise CL-motif scores $x_{m_t^*,t}$ we first need to learn the CL-motifs. To our advantage these can be learned from input control data, which represents RNA fragments crosslinked to a mixture of background proteins (see Section 3.4) and thus reflects common crosslinking preferences. The motifs and theirs scores are computed in a preprocessing step as follows:

1. We call crosslink sites on the input data using the basic version of PureCLIP, i.e. without incorporating covariates.
2. To learn CL-motifs we run DREME [10] on 10 bp windows spanning the called input crosslink sites, while using 10 bp windows located 20 bp upstream and downstream as a background control.
3. We use FIMO [55] to compute occurrences of those CL-motifs within the reference genome and their corresponding scores.

A more detailed description is provided in Appendix A.5.

Parameter estimation

Instead of estimating the probability parameter p_j that maximizes the weighted MLE for each crosslinking state $z_j^{(2)}$ as described in Section 5.3.2, we now need to estimate the regression coefficients $\boldsymbol{\beta}_j = \beta_{j,0}, \dots, \beta_{j,q}$. For this purpose we first estimate $\beta_{j,0}$ using the majority of positions which has no CL-motif match, i.e. a CL-motif score $x_{m_t^*,t} = 0$, as follows:

$$\begin{aligned}\beta''_{j,0} = \arg\max_{\beta_0} \sum_{t=1}^{T} \gamma_{t,1j} \\ \cdot \left[-\ln\left(1-(1-p)^{\hat{n}_t}\right) + k_t \ln(p) + (\hat{n}_t - k_t)\ln(1-p) \right] \cdot I_{0,t}^{(2)}, \qquad (5.28)\end{aligned}$$

where

$$p = \frac{1}{1 + exp(-\beta_0)}$$

and $I_{0,t}^{(2)}$ is an indicator variable defined as

$$I_{0,t}^{(2)} = \begin{cases} 1 & \text{if } (k_t \geq 1) \wedge (x_{m,t} = 0 \; \forall m \in \{1,\dots,q\}) \\ 0 & \text{else.} \end{cases} \qquad (5.29)$$

Since we assume that each position matches at most one CL-motif, the regression coefficients $\boldsymbol{\beta}_{j,1:q}$ for the q CL-motifs can be learned independently of each other. Given

$\beta_{j,0}$, we estimate the regression coefficient $\beta_{j,m}$ for each CL-motif $m \in \{1, \ldots, q\}$ as:

$$\beta''_{j,m} = \arg\max_{\beta_m} \sum_{t=1}^{T} \gamma_{t,1j} \cdot \left[-\ln\left(1-(1-p_t)^{\hat{n}_t}\right) + k_t \ln(p_t) + (\hat{n}_t - k_t)\ln(1-p_t) \right] \cdot I^{(2)}_{m,t}, \tag{5.30}$$

where

$$p_t = \frac{1}{1 + exp(-(\beta_{j,0} + \beta_m x_{m,t}))}$$

and

$$I^{(2)}_{m,t} = \begin{cases} 1 & \text{if } (k_t \geq 1) \wedge (x_{m,t} > 0) \wedge (m = m^*_t) \\ 0 & \text{else.} \end{cases} \tag{5.31}$$

The indicator variable $I^{(2)}_{m,t}$ limits the parameter learning for each CL-motif m to positions that have at least one read start, are located within an occurrence of CL-motif m and for which no other CL-motif has a higher match score.

Importantly, since all parameters can be estimated separately, we again make use of Brent's method for one-dimensional optimization problems.

Initialization

For initialization, the regression coefficients $\boldsymbol{\beta}_{j,1:q}$ corresponding to the q CL-motifs are set to 0. The intercept parameters $\beta_{0,0}$ and $\beta_{1,0}$ of the *non-crosslink* and *crosslink* state, respectively, are set to values according to the initial values of the corresponding probability parameters in the basic model (see Section 5.3.4): $\beta_{0,0} = \ln \frac{0.01}{1-0.01}$ and $\beta_{1,0} = \ln \frac{0.15}{1-0.15}$.

5.5. Extension to incorporate replicate information into PureCLIP

In the previous sections we described a model to detect protein-RNA interactions from one target CLIP dataset. This data can be either from one individual sample or from merged biological replicates.

When analysing CLIP experiments where biological replicates are produced there are several ways to handle those. One can apply an analysis method on the individual replicates and then use either the intersection or the union of the called sites, depending on whether one aims for high precision or high sensitivity. Alternatively, as already mentioned, one can merge the replicates prior to the analysis, which usually provides a compromise regarding precision and sensitivity. However, the exact effect depends on the applied method.

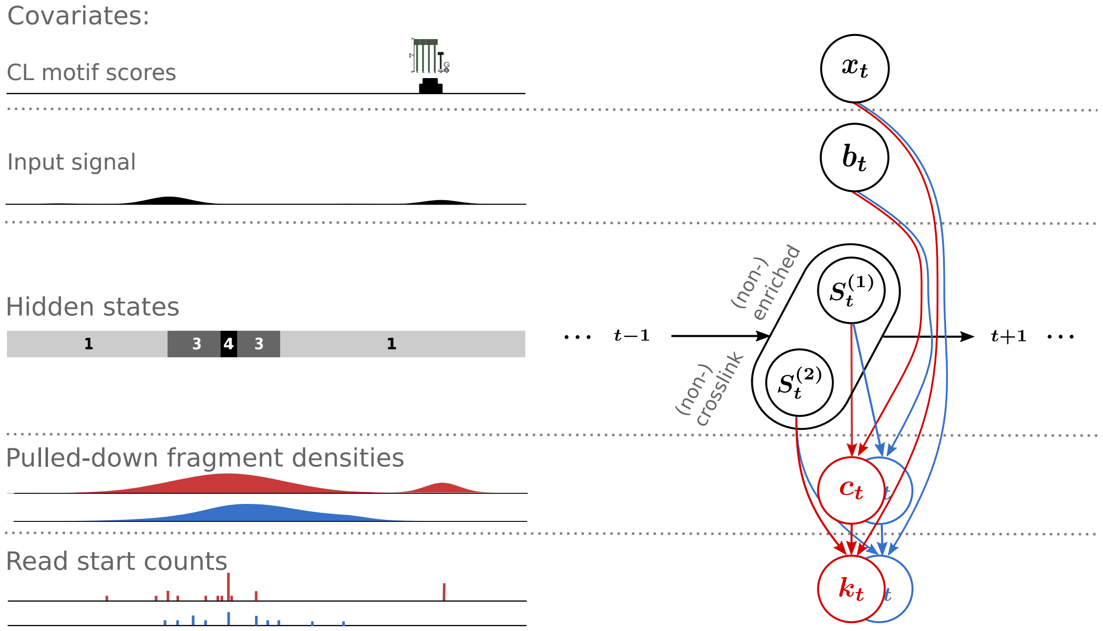


Figure 5.5.: Summary of the non-homogeneous HMM framework including multiple replicates. The emission probability parameters are learned separately for the individual replicates. The final transition and state posterior probabilities are then computed using the joint emission probabilities.

A more advanced way is to explicitly account for information from replicate experiments in the statistical model, as a few tools already do for CLIP peak-calling, such as PEAKachu [67] and OmniCLIP [41]. Different strategies are possible to account for replicate information, from computing joint probabilities using simple multiplications to explicitly modelling the variance between replicates [67, 89]. The aim is to reduce the number of false positive calls caused by signals only present in one replicate, while increasing the sensitivity for regions with signals present in multiple replicates.

In any case, care needs to be taken, since beside true protein-RNA binding signals also artefacts (see Section 3.3), for example within CL-motifs, are often highly reproducible. Given that our model is able to correct for such artefacts, we extended the PureCLIP framework to incorporate individual replicates as follows: since CLIP replicates often have no more than two replicates, we do not model the position-wise variance across replicates, but instead combine the emission probabilities learned for each replicate individually, assuming the observed signals are conditionally independent given the hidden states. Figure 5.5 depicts the extension of the method to include replicates schematically.

Assume we are given R individual CLIP replicates, each with a sequence of observed fragment densities $\boldsymbol{c}^r$ and read start counts $\boldsymbol{k}^r$, $r \in \{1, \dots, R\}$. To include the replicates, the model is modified in the following way:

1. An HMM is build and trained as previously described for the covered regions

(see Section 5.3) of each individual replicate r separately. The parameters θ^r are learned.

2. The intersection of covered regions across all replicates is computed.

3. Based on these intersected regions, a new HMM with the same structure is build, i.e. with four hidden states. The following initializations and modified emission probabilities are used:

   a) Initial probabilities

$$\pi_{(t)j} = \frac{1}{R}\sum_{r=1}^{R} \gamma_{tj}^{r}, \qquad \forall j \in \{1, \dots, 4\} \tag{5.32}$$

   b) Transition probabilities

$$a_{ij} = \frac{1}{R}\sum_{r=1}^{R} a_{ij}^{r}, \qquad \forall i, j \in \{1, \dots, 4\} \tag{5.33}$$

   c) Emission probabilities

$$\begin{aligned} &P(Y_t^1 = y_t^1, \dots, Y_t^R = y_t^R \mid S_t^{(1)} = z_i^{(1)}, S_t^{(2)} = z_j^{(2)}, \theta^{1:R}) \\ &= \prod_{r=1}^{R} P(Y_t^r = y_t^r \mid S_t^{(1)} = z_i^{(1)}, S_t^{(2)} = z_j^{(2)}, \theta^r), \qquad \forall t \in \{1, \dots, T\} \end{aligned} \tag{5.34}$$

   where $y_t^r = (c_t^r, k_t^r)$.

4. Initial, transition and state posterior probabilities are then updated using fixed emission probabilities.

Finally, as in the PureCLIP model for one target dataset, all sites that are most likely *enriched + crosslink* are reported.

Recall that due to the truncated distributions used to model the emission probabilities, sites that have a fragment density $c_t < \tau$ have an emission probability of 0 for the *enriched* state $z_1^{(1)}$ and an emission probability of 1 for the *non-enriched* state $z_0^{(1)}$ (see Section 5.3.1). Similarly, sites with no read start count ($k_t = 0$) have an emission probability of 0 for the *crosslink* state $z_1^{(2)}$ and an emission probability of 1 for the *non-crosslink* state $z_0^{(2)}$ (see Section 5.3.2). Consequently, only sites with $c_t^r \geq \tau$ and $k_t^r \geq 1$ in all R replicates can be classified as *enriched + crosslink*.

5.6. Implementation

PureCLIP is a command-line tool implemented in C++ using SeqAn [117], the GSL [49] and Boost [123]. OpenMP [37] is used for parallelization. In the following, we will briefly discuss some of the relevant implementation aspects. An overview of the main important user options is provided in Appendix A.12.

5.6.1. Training set

The HMM parameters are learned for each dataset individually to account for the specific target and background signal characteristics. More precisely, PureCLIP learns the parameters on a subset of the data based on a user defined list of chromosomes or contigs (default: all). Often, learning on a small subset of chromosomes does not significantly impair PureCLIP's performance, while reducing the memory and runtime consumption (for details see Section 9.2.2). Note that this also depends on the size of the dataset. For example, when analysing very sparse CLIP data or data from organisms with a relatively small genome or transcriptome size, it is probably best to use the whole dataset for parameter learning. After the parameters are learned, the HMM is applied to the whole dataset, i.e. state posterior probabilities are computed and posterior decoding is applied to infer the hidden states.

5.6.2. Numerical stability and arithmetic precision

Recall that the forward probability defined in Equation 4.18 denotes the probability to observe a certain subsequence $\boldsymbol{y}_{1:t}$ and being in state z_i at position t under the model. Similarly, the backward probability defined in Equation 4.22 denotes the probability to observe the subsequence $\boldsymbol{y}_{t+1:T}$. Both probabilities decrease drastically with increasing sequence lengths and take values close to zero.

To avoid arithmetic underflows, the forward-backward algorithm [114] is computed in log-space. Thus, for the forward algorithm, instead of $\alpha_i(t)$ we compute

$$\begin{aligned}\log \alpha_i(t) &= \log \sum_{j=1}^{4} \alpha_j(t-1) a_{ji} e_i(y_t) \\ &= \log \sum_{j=1}^{4} \exp\left(\log\left(\alpha_j(t-1)\right) + \log\left(a_{ji}\right) + \log\left(e_i(y_t)\right)\right), \end{aligned} \tag{5.35}$$

where $e_i(y_t) = P(C_t = c_t, K_t = k_t \mid S_t^{(1)} = z_i^{(1)}, S_t^{(2)} = z_j^{(2)})$. However, since $exp(x)$ approaches zero for large negative x, the problem remains that the typically large values of $\log\left(\alpha_j(t-1)\right) + \log\left(a_{ji}\right) + \log\left(e_i(y_t)\right)$ quickly cause underflows. To address this, we make use of the so called *log-sum-exp trick* [103], which allows us to optimize the range on which the exponential operations are applied:

$$\begin{aligned}\log \sum_{j=1}^{4} \exp(x_j) &= \log \sum_{j=1}^{4} \exp(x_j - b)\exp(b) \\ &= \log\left(\exp(b) \sum_{j=1}^{4} \exp(x_j - b)\right) \\ &= b + \log \sum_{j=1}^{4} \exp(x_j - b), \end{aligned} \tag{5.36}$$

where $b = \max_j \ x_j$. This forces the largest value of $(x_j - b)$ to be zero. As a consequence, when values of similar size are summed up, this method effectively prevents underflows. When values of a larger range are summed up and $exp(x_j - b)$ is below the available arithmetic precision, this term will be lost, but the introduced error is negligible in practice. Equivalently to the log-forward probabilities the log-backward probabilities are computed. Both are then used to compute the posterior, initial and transition probabilities.

A general drawback of the log-sum-exp trick is an increased computational cost due to additional, more expensive $\log$ and $\exp$ operations replacing simple additions. To avoid increased running times, log-sum-exp values are precomputed for a certain range of $(x_j - b)$ values, while ensuring a reasonably high accuracy.

In addition to the forward and backward probabilities, the emission probabilities themselves can become fairly small, potentially causing arithmetic underflows and thus taking values of zero. They depend on the learned parameters and extremely low values are usually caused by outliers, i.e. high fragment densities or read start counts, often originating from mapping artefacts, not properly removed PCR duplicates or pile-ups of multi mapping reads. To address this issue, the individual *non-enriched* and *enriched* emission probabilities as well as the *non-crosslink* and *crosslink* emission probabilities are first computed using the precision of the C extended-precision floating-point type (*long double*). On the de facto standard x86 CPU architecture this type usually corresponds to an 80-bit floating-point format, allowing values in the range from approximately $3.36 \cdot 10^{-4932}$ to $1.19 \cdot 10^{4932}$.

The joint emission probabilities are again computed in log-space to avoid underflows caused by the multiplication of small values. This ensures a high numerical stability, as long as the emission probabilities obtained from the individual gamma or binomial distributions are greater than zero. By default, the log emission probabilities are stored using the C double-precision floating-point type (*double*). Usually this type occupies 64 bit and can represent values in the range from approximately $2.22 \cdot 10^{-308}$ to $1.8 \cdot 10^{308}$. If for one position all emission probabilities would become zero during the parameter learning procedure, PureCLIP would return an error. In contrast, if this would happen during the application step, i.e. for a sequence not used for parameter learning, the corresponding covered interval would be discarded and PureCLIP would return a warning. This implementation enables accurate computations for datasets with a wide range of different signal characteristics.

5.6.3. Choice of numerical optimization techniques

The choice of numerical optimization techniques for parameter estimation was beside efficiency reasons also influenced by practical implications. First, only a limited number of C++ libraries exists providing a limited number of well documented and tested numerical optimization algorithms. Second, we aimed to keep the number of used external libraries as low as possible in order to simplify the installation process of PureCLIP for users. For this reason, for example, we made use of the GSL [49] implementation of the Nelder-Mead algorithm and extended it to allow for parameter constraints using penalty-based soft constraints (see Appendix A.4).

5.6.4. Computational complexity

The runtime and memory consumption grow linearly with respect to the total length of covered regions T (see Section 5.3). For each position t the used observations, i.e. read start count k_t, fragment density c_t and estimated number of fragments n_t, need to be stored as well as the covariates, i.e. background fragment density b_t and CL-motif score x_t, if applicable. Additionally, for each position and state the emission probabilities and state posterior probabilities need to be stored.

The running time is dominated by the Baum-Welch algorithm to estimate the model parameters. Given that we have a constant number of states, within each Baum-Welch iteration we need to compute the emission probabilities and the forward-backward variables, each with a time complexity of $\mathcal{O}(T)$. Additionally, we need to estimate the emission probability parameters for each state using numerical optimization strategies. This requires $\mathcal{O}(T_l)$ function evaluations within each iteration, where T_l is the number of positions used for learning, i.e. with fragment densities and read start counts above a certain threshold. The number of required iterations differs between states and typically decreases with Baum-Welch iterations. It also strongly depends on the dimensionality of the optimization problem, i.e. for the simplex algorithm this increases from two to three dimensions when including covariates (see Section 5.4.1). Recall that we apply the Baum-Welch algorithm in sequential blocks (see Section5.3.5), where either the *non-enriched* and *enriched* or the *non-crosslink* and *crosslink* emission probability parameters are learned. For these blocks, the number of Baum-Welch iterations typically varies between between 6 and 25. PureCLIP's memory and running time consumption in practice will be shown for an example dataset in comparison to other methods in Section 9.1.

5.6.5. Availability

PureCLIP is licensed under the GPLv3 and freely available at `https://github.com/skrakau/PureCLIP`. The documentation is provided at `https://pureclip.readthedocs.io`. Additionally, it can be easily installed using Bioconda [59] (`https://bioconda.github.io`). Moreover, it was integrated by the Freiburg Galaxy Team into the European Galaxy server `https://usegalaxy.eu/`, an online platform for reproducible computational biological research [1].

6. Data

In this chapter we will describe the publicly available experimental CLIP datasets used for the evaluations presented in the following chapters. Moreover, we will present a new method to simulate truncation-based CLIP data and how it was applied to simulate several datasets with different characteristics.

6.1. Experimental eCLIP and iCLIP datasets

We analysed three eCLIP datasets targeting the proteins PUM2, RBFOX2 and U2AF2 [143] and two iCLIP datasets targeting U2AF2 [155] and PTBP1 [29], all generated on human cell lines. The data was preprocessed as described in Section 5.1, more precisely the reads were mapped to the human genome (hg19) while accounting for splice junctions (using Ensembl Release 75 annotations) and keeping only uniquely mapping reads. Further details are described in Appendix A.3. Table 6.1 provides a summary of the used datasets and shows the known binding characteristics of the target proteins, which will be, among other criteria, used for the method evaluation.

Table 6.1.: Datasets used in the evaluation. The PTBP1 iCLIP dataset comprises four replicates, all others two.

Protein and protocol	Binding characteristics	Cell line	Accession no.	Total reads (pooled)	Filtered reads (pooled)	Input control
PUM2 eCLIP	sequence motif	K562	GSE91965	28,648,140	2,211,125	yes
RBFOX2 eCLIP	sequence motif	K562	GSE92030	38,506,096	11,178,409	yes
U2AF2 eCLIP	upstream of 3’ splice site	K562	GSE92143	25,644,172	8,679,323	yes
U2AF2 iCLIP	upstream of 3’ splice site	HeLa	E-MTAB-1371	22,905,120	12,060,930	no
PTBP1 iCLIP	upstream of 3’ splice site of silenced exons	HeLa	E-MTAB-3108	12,127,611	6,425,133	no

6.2. Simulated truncation-based CLIP data

In order to evaluate our method's performance, we aimed at realistic truncation-based CLIP data with known RBP binding sites. Since the only available CLIP simulator is limited to PAR-CLIP and HITS-CLIP data [75], we implemented our own simulation workflow to mimic the experimental steps of the iCLIP protocol as described in Section 3.2.1. Starting from real RNA-seq data and known binding regions of a certain protein, our simulation aims to reproduce the characteristics of truncation-based CLIP data as accurately as possible. Real RNA-seq data has the advantage that it has already realistic transcript abundances, sequencing errors and mapping characteristics. Using known binding regions further ensures a realistic distribution of the simulated signals across the transcriptome.

6.2.1. Simulation framework

To simulate target signal, our workflow uses aligned RNA-seq data. It pulls down a certain fraction of the fragments that cover a known binding region and then applies truncations according to a given rate. Furthermore, non-specific binding of background proteins is simulated using previously published common background regions [119] and by adding random noise from RNA-seq data.

Figure 6.1 illustrates the simulation workflow, which comprises the following steps:

1. **Fragmentation:** To obtain RNA fragment lengths comparable to those of iCLIP experiments (30-300 bp, as described in [126]), we first simulate new fragment lengths using a normal distribution (mean: 165 bp, standard deviation: 50 bp).

2. **Binding regions:** Given the known sequence motif of an RBP, we compute genome-wide motif occurrences using FIMO [55] (`-thresh 0.01`) and keep those located within transcripts and with a minimal distance of 100 bp to the nearest annotated splice site.

3. **Crosslink sites:** For each binding region i, the number of crosslink sites is drawn from a uniform distribution $c_i \in \{1, \ldots, c^T_{max}\}$ (default: $c^T_{max} = 4$). The c_i crosslink sites are then randomly assigned to positions within the binding region.

4. **Pull-down:** RNA fragments overlapping binding regions are pulled-down with a certain rate (default: 1.0). With this, different RNA binding affinities of the target protein can be simulated.

5. **Reverse transcription:** We use the 5'-end read for each fragment and simulate one of the following events:

   a. **On-target truncation**

      The read start is shifted to one of the simulated crosslink sites within the current binding region according to a given truncation probability (default: 0.7).

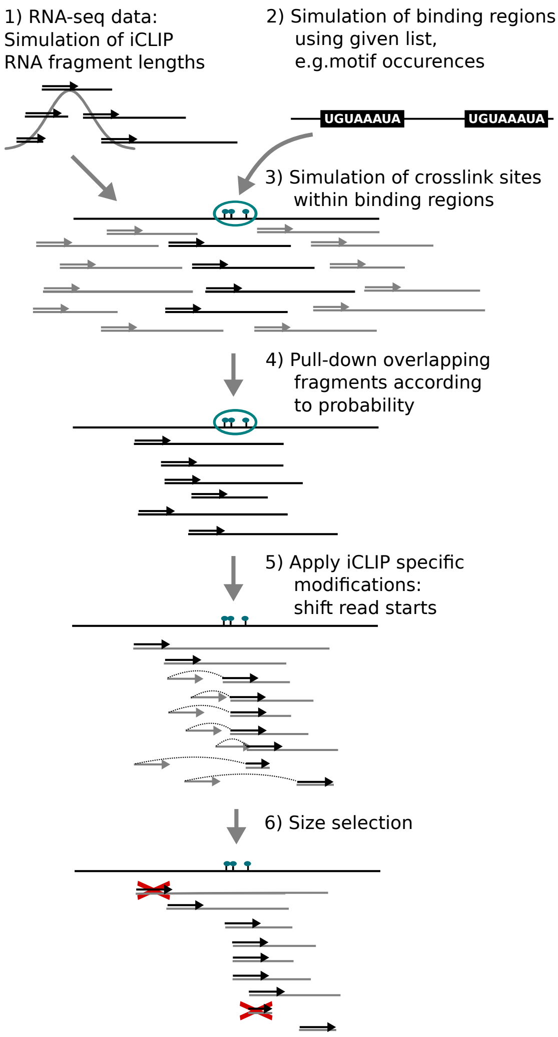


Figure 6.1.: Core of the iCLIP data simulation workflow.

b. **Off-target truncation**

The read start is shifted to any other position within the fragment according to a given off-target truncation probability (default: 0.2).

c. **Read-through**

The read start is not changed (default: 0.1).

6. **Size selection:** To obtain a broad range of cDNA lengths, we keep reads with underlying fragment lengths between 30-140 nt (as recently recommended in [61]).

Note that we start the simulation with the 5'-ends of the given RNA-seq fragments and adjust the 3'-ends according to the simulated length (see Figure 6.1, Step 1). For the final read generation, the length is set to 50 bp and shortened accordingly for shorter cDNA fragments remaining after step (6).

In addition to the RBP binding signal, we also simulated background noise that can be for example caused by sticky RNAs or by the binding of non-specific background proteins (see Section 3.3.2). We did this by applying the aforementioned steps on the list of known common background binding regions published in [119], with the following modifications to the core workflow:

- We pull down RNA fragments overlapping the background regions with a rate proportionally to the original region-wise background binding scores as reported [119]. The pull-down rate is obtained using a scaling factor (default: 0.005), while ensuring values between 0 and 1.
- The on-target truncation probability is slightly decreased (default: 0.5), assuming less specific crosslink sites within background binding regions.
- As background binding regions are typically longer than the used motif occurrences for the target signal, we increased the number of simulated crosslink sites within such a region: $c_i \in \{c^T_{max}, \dots, c^B_{max}\}$ (default: $c^B_{max} = 15$).

We supplement this non-specific background binding signal with reads randomly sampled from the RNA-seq data with a certain rate (default: 0.01) in order to simulate random noise.

6.2.2. Implementation and availability

The main part of the simulation workflow is implemented in C++ using SeqAn [117] and OpenMP [37] and is provided as a command-line tool. It is freely available under the GPLv3 license and can be downloaded from `https://github.com/skrakau/sim_iCLIP`. It requires mapped RNA-seq data, a list of target binding regions and, if applicable, background binding regions.

The aligned reads are then written to a BAM file, the simulated binding regions and crosslink sites to BED files. Note that while the read lengths are adjusted according to simulated fragment ends and possible truncations, only dummy sequences are written to the BAM file, i.e. a realigning is not possible.

6.2.3. Generation of simulated data

The simulated data used in this thesis is based on the total RNA-seq dataset ENCSR885DVH from ENCODE (Homo sapiens, cell line K562, whole cell fraction). We used the provided reads already aligned with STAR against the genome (hg19) and merged the datasets of the available replicates. Soft-clipped reads were discarded. In order to obtain a realistic distribution of target binding regions we used the PUM2 sequence motif retrieved from the ATtRACT database [54] (see Figure 8.4a) and computed its occurrences within the genome using FIMO. iCLIP data was simulated using the default parameters as described in Section 6.2.1, if not stated otherwise.

To evaluate PureCLIP's performance under different conditions, we then generated datasets with different target binding and background noise characteristics. For this purpose we varied the rate of random noise (0.0, 0.01, 0.05), the pull-down factor of the background binding signal (0.001, 0.005, 0.01), the truncation rates (0.8, 0.7, 0.5) as well as the pull-down rates for the target signal (1.0, 0.5, 0.25).

7. Results from PureCLIP's model training

Before we will compare PureCLIP's performance to that of competing methods, we will briefly give some insights into the behavior of the underlying model. Based on an example, the previously described PUM2 eCLIP dataset, we will show the empirical distributions of the two observed signals, the learned emission probability distributions and the resulting classifications. For the non-homogeneous HMM we additionally show the predicted gamma mean parameters $\mu_{i,t}$, dependent on the input control signal at position t, and the predicted binomial probability parameters $p_{j,t}$, dependent on the CL-motif score.

7.1. *Non-enriched* and *enriched* emission probabilities

Since we do not know the empirical fragment density distributions corresponding to the assumed *non-enriched* and *enriched* components of our model, we compare the (compound) empirical distribution to the compound emission probability distribution learned by PureCLIP in basic mode, i.e. without incorporating control data. The results are shown in Figure 7.1. Note that for PureCLIP in basic mode the shape of the *non-enriched* gamma distribution is always exponential, because its value is constrained with $\lambda_0 = 1$ for stability reasons (see Section 5.3.1). In contrast, the shape parameter of the *enriched* gamma distribution is often greater than 1, i.e. it has no exponential form and instead a mode at some positive fragment density value. The results in Figure 7.1 show that the learned *non-enriched* and *enriched* fragment density emission probability distributions together approximate the empirical distribution. Moreover, it is noteworthy that the majority of sites has very low fragment densities and is classified as *non-enriched.*

Next, we addressed the incorporation of input control eCLIP data. The results presented in Figure 7.2a show the notable correlation between the target eCLIP and input fragment densities with a Pearson correlation coefficient of 0.59 and a p value $< 2.2e - 16$. PureCLIP learns for both gamma distributions the expected mean parameter $\mu_{i,t}$ dependent on the input signal b_t, which enables a bias-corrected classification into *non-enriched* or *enriched* sites (see Figure 7.2b).

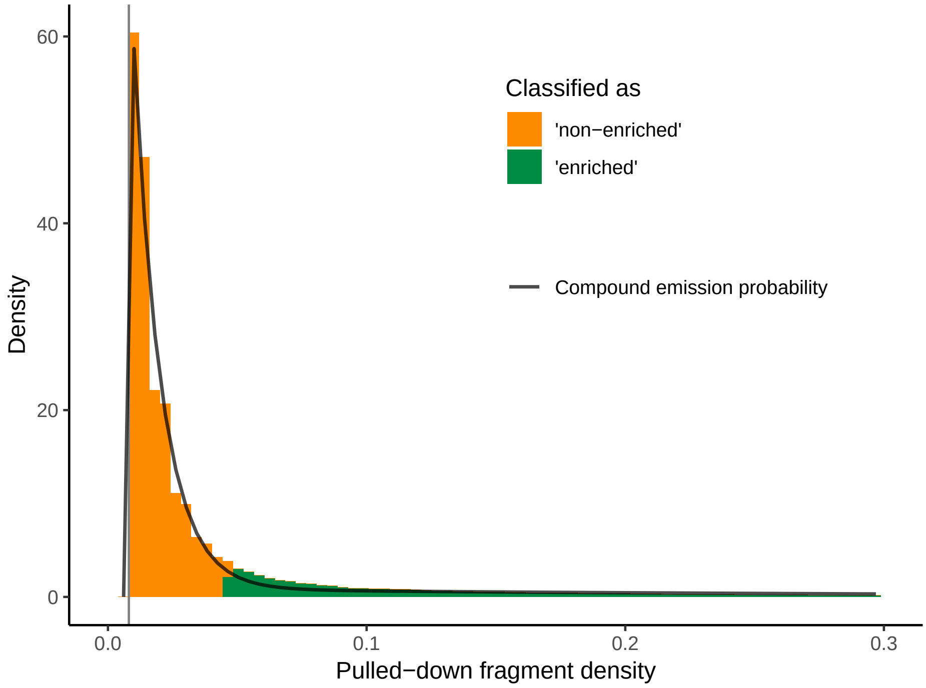


Figure 7.1.: Distribution of empirical fragment densities at sites with at least one read start (histogram) and the resulting classification by PureCLIP for the PUM2 eCLIP data. The black line depicts the compound emission probability distribution of the fragment densities: $w_0 f_{LTG}(c_t; \mu_0, \lambda_0, \tau) + w_1 f_{LTG}(c_t; \mu_1, \lambda_1, \tau)$, where the weights w_0 and w_1 are derived from the corresponding posterior probabilities. Note that the compound probability distribution is shown here solely for comparison and is not used within the model itself.

7.2. *Non-crosslink* and *crosslink* emission probabilities

Similarly to the *non-enriched* and *enriched* case, we compared the learned *non-crosslink* and *crosslink* binomial emission probability distributions to the empirical read start count distributions. Since the binomial distributions depend on the position specific size parameter n_t, we plotted the empirical and the learned emission probability distributions for different n values (see Figure 7.3). Furthermore, instead of the compound distribution we now plotted the *non-crosslink* and *crosslink* binomial distributions separately to illustrate their increasing separation with increasing n values (see Figure 7.3, column 2). For sites with $n_t \leq 10$ the *non-crosslink* and *crosslink* emission probability distributions largely overlap, which causes a higher ambiguity in classification. However, since also transition probabilities play an important role for state inference and only *enriched + crosslink* sites are considered, this effect on the results is limited. Moreover, such ambiguities would be reflected by low scores. Figure 7.3 (column 3 and 4) shows the resulting distributions of sites classified as *non-crosslink* or *crosslink*

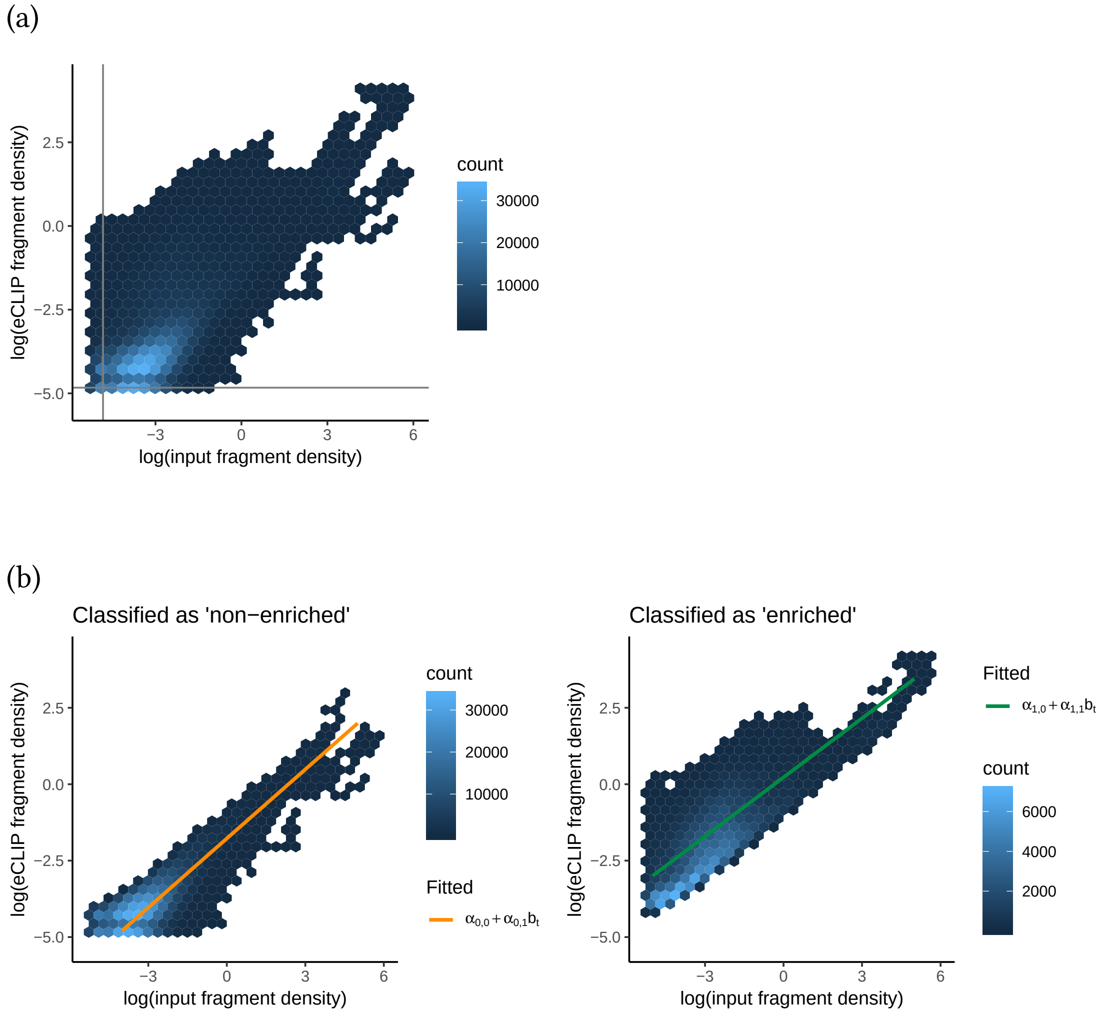


Figure 7.2.: Correlation between PUM2 eCLIP and input fragment densities at positions with at least one read starting (not including singleton reads): **a)** for all positions and **b)** limited to the set of positions classified as *non-enriched* (left) and *enriched* (right) by PureCLIP based on the input signal. The expected pulled-down fragment densities modelled based on the input fragment densities b_t are shown as orange and green lines for the *non-enriched* and *enriched* states, respectively.

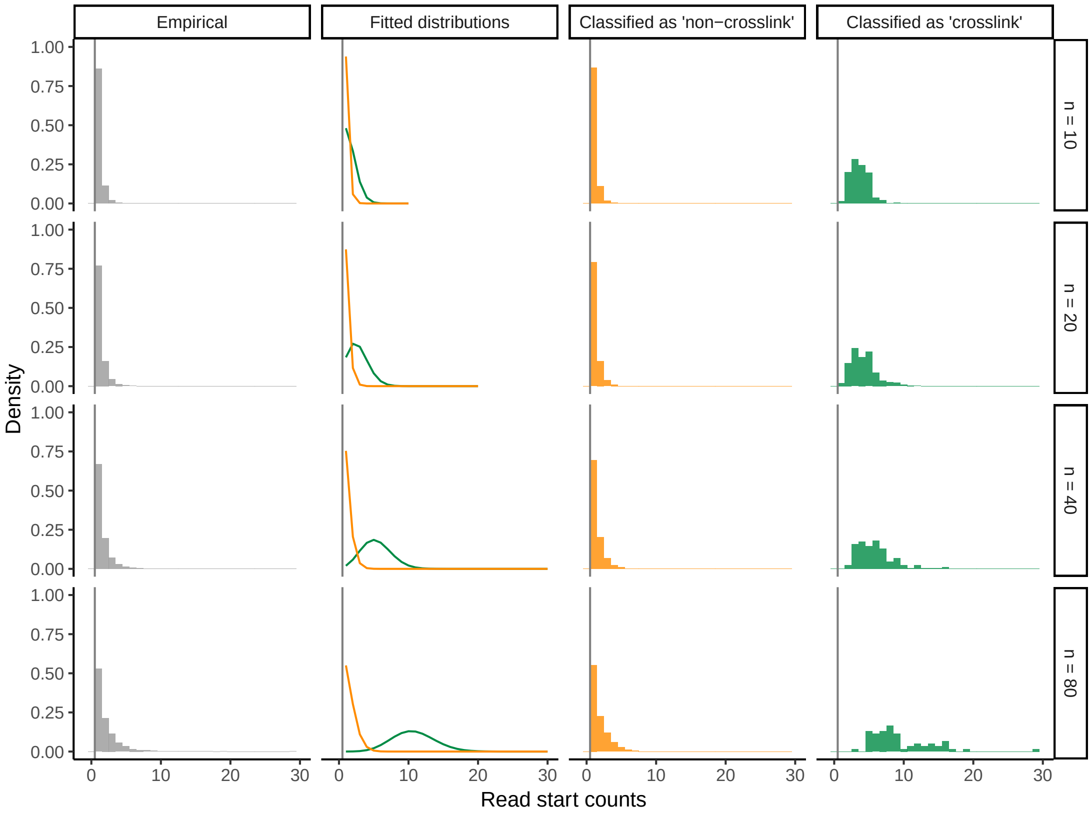


Figure 7.3.: Read start count distributions for different values of $\hat{n}_t$ and fitted emission probability distributions for PUM2 eCLIP data. The shown empirical distributions comprise only those sites that are used for the parameter fitting, i.e. with at least one read start. The fitted distributions for the *non-crosslink* and *crosslink* states are shown in orange and green, respectively. The learned binomial probability parameters are: $p_0 = 0.01$ and $p_1 = 0.13$. Additionally, the empirical distributions restricted to sites classified as *non-crosslink* or *crosslink* are shown. The vertical gray lines depict the zero-truncation of the binomial distributions.

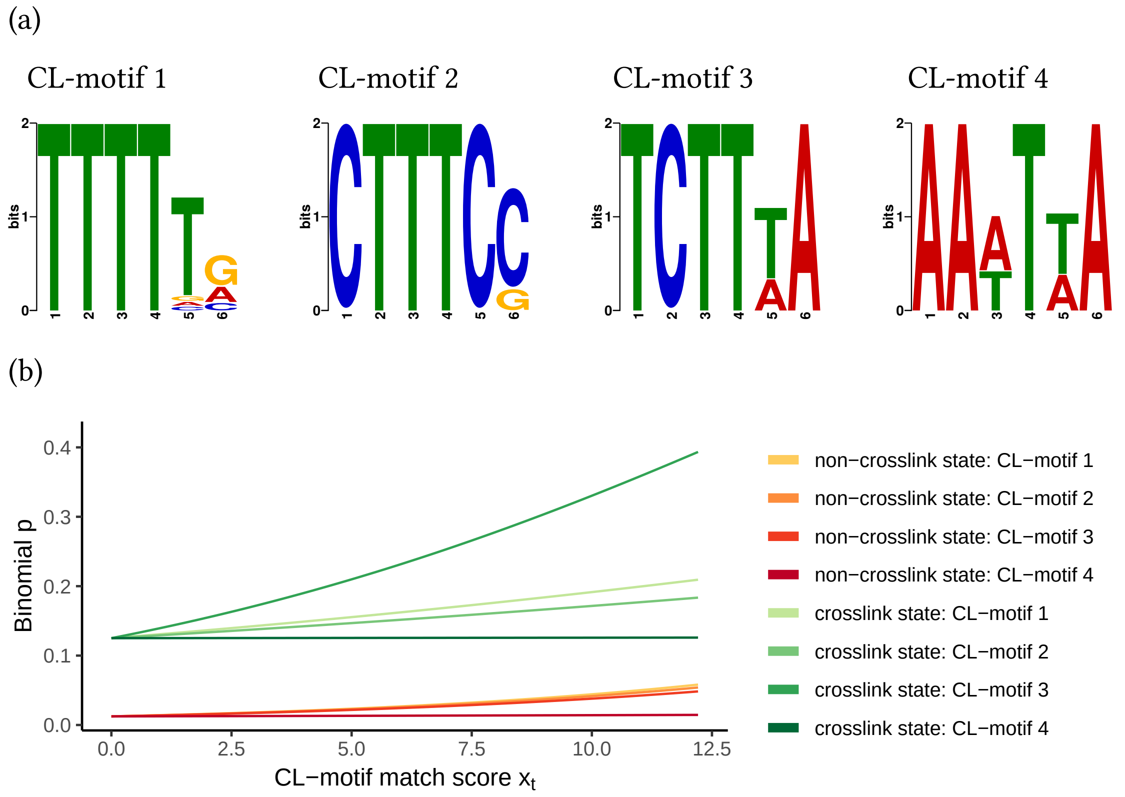


Figure 7.4.: CL-bias correction for PUM2 eCLIP data. **a)** CL-motif analysis. Logo representation of the four top scoring motifs among the top 5000 crosslink sites called on the input dataset. **b)** Predicted binomial probability parameter dependent on the CL-motif score x_t for the *non-crosslink* and *crosslink* state and for each of the shown CL-motifs.

for different n values.

To correct for the crosslinking sequence bias, the CL-motifs shown in Figure 7.4a were obtained from the input control data, as described in Section 5.4.2. As previously reported [61], U/T-rich motifs are overrepresented. The CL-motifs were then used to compute CL-motif match scores for the given reference sequence. For both the *non-crosslink* and *crosslink* state, the predicted binomial probability parameters for each CL-motif based on the position-wise CL-motif match score x_t are shown in Figure 7.4b. The analysis of the size-matched input control data from the RBFOX2 and U2AF2 eCLIP experiments revealed similar CL-motifs, which are shown in Figure A.2.

8. Evaluation

8.1. Overview

The available methods to analyse CLIP data can be divided into two main categories: those, that detect individual crosslink sites and those that detect binding regions, where some of the former provide a strategy to merge crosslink sites to binding regions. This also applies to PureCLIP (see Figure 5.1) and we therefore perform the evaluation both on the crosslink site and on the binding region level.

8.1.1. Binding regions versus individual crosslink sites

Before we describe the evaluation in more detail, we first briefly discuss and define the term *binding region.* While it is clear that crosslink sites correspond to single nucleotides that were in direct contact with the protein (see Section 3.2), binding regions are less well defined. In general, we use this term to refer to a region of direct interactions between the target RBP and the RNA. For proteins binding via classical RNA-binding domains (RBDs), as described in Section 2.3, it is known that the individual domains often bind to regions that are only a few nucleotides long [8]. However, many proteins harbour multiple RBDs, often causing multiple, spaced regions of direct interactions. In such cases it is less obvious how to define a binding region.

In the following chapter we focus on the three proteins PUM2, RBFOX2 and U2AF2. PUM2 contains one RBD which binds to an 8 nt sequence motif, while allowing direct interactions at all 8 bases [90]. RBFOX2 also contains only one RBD, which is known to recognize a 6 nt sequence motif [40, 149], causing crosslinks predominantly at two specific nucleotides. In contrast, U2AF2 contains two RBDs, however, it is known that these RBDs recognize a 9 nt polypyrimidine track in a side-by-side conformation [3]. Of course, we can not exclude interactions outside of these regions and also aim detect those.

8.1.2. Evaluation strategies

Evaluating a method's performance in analysing CLIP-seq data is not trivial, since no gold-standard of binding regions or crosslink sites exists. We addressed this by comparing the methods based on three different strategies:

1. *Simulated data*

   We compared their performance on simulated data (see Section 8.2.1 and 8.2.2).

2. *Proteins with known binding characteristics*

   We used used real iCLIP and eCLIP datasets of proteins with known binding characteristics, such as known sequence motifs or known predominant binding regions. This evaluation was performed on the proteins PUM2, RBFOX2 and U2AF2 (see Section 8.3.2 and 8.7).

3. *Replicate agreement*

   Knowledge about protein-binding characteristics is often limited and it is unclear how far the protein of interest can also bind to alternative regions. We therefore additionally assessed the agreement of called crosslink sites between PUM2, RBFOX2 and U2AF2 eCLIP replicates, assuming that true protein-specific binding signals are reproducible (see Section 8.3.4). Note that this evaluation was not performed on the level of binding regions.

It is noteworthy that within previous studies [26, 41, 124] binding regions were often simply interpreted as true positives if they overlap with known binding regions, e.g. motifs. This evaluation strategy indirectly favours methods calling broader binding regions, independently of whether the called regions reflect the true binding regions. Since we aim to capture interactions with high resolution, we designed our evaluation strategies such that they measure the precision of the individual called binding regions.

8.1.3. Performance measurements

For all main evaluation strategies, we investigated the performance of the different tools by assessing their precision for different score thresholds, where the computation of the precision depends on the evaluation strategy and will be described in the corresponding contexts. By using different thresholds we aim to additionally evaluate the rankings of the called crosslink sites or binding regions provided by each method. As scores we use the method's reported p values, FDRs or custom scores. Importantly, in the following we refer to low p values or FDRs as high scores and vice versa. We cannot compute precision-recall curves, since for experimental CLIP data the number of false negatives is unknown. For these reasons we plot the precision (or a corresponding estimate) versus the number of called sites or regions for each score threshold. In the following we refer to this as the *predicted positives (PP)*. This allows comparable and consistent plots throughout the evaluation. An explanatory example is shown in Figure 8.1.

To generate such precision-PP curves, the applied score thresholds were chosen so that the whole range of reported scores is represented. Consequently, for each method the lowest *number of predicted positives (#PP)* corresponds to all sites reported with the highest score, while the highest #PP corresponds to all reported sites. In this context it is noteworthy that for some methods the range of reported scores is limited by an upper value, for example, by a p value of 0 (referred to as highly scoring). Applying the different thresholds thus gives us for each method a range of obtained #PPs, which is supported by different scores. Since the compared methods were designed with different goals concerning the trade-off between precision and sensitivity, the supported

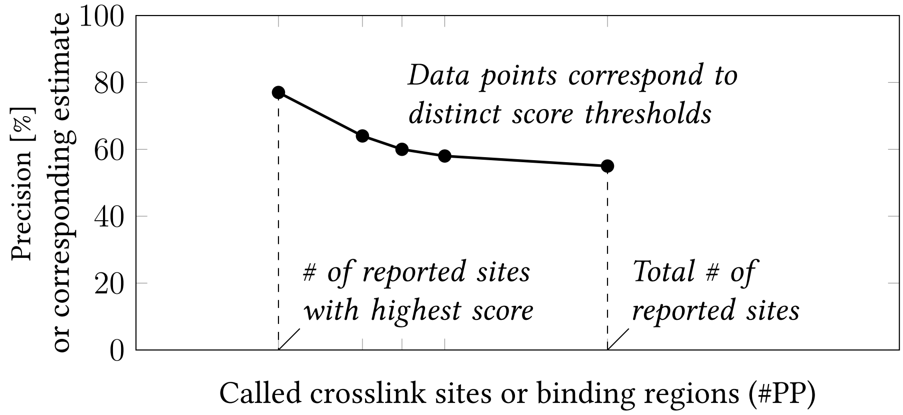


Figure 8.1.: Schematic precision-PP curve used for evaluation throughout this thesis.

#PP ranges of the different methods do not always overlap. For this reason the area under the curve (AUC) – as commonly computed for precision-recall curves – would not provide comparable performance measurements in this context.

8.1.4. Evaluated methods

In the following we briefly describe the applied settings for the different tools used for comparison. For a detailed description of the methods see Section 3.6.2. Note that although some tools provide functionalities to preprocess CLIP data, e.g. to remove PCR duplicates, we used only the core analysis methods to detect protein-RNA interactions and used the same preprocessed alignments for all tools to allow for an unbiased comparison.

Individual crosslink site detection

On the crosslink site level we compared PureCLIP's performance against CITS [149], iCount [36] and against applying a simple threshold.

CITS The scripts were run with default parameters as described in the documentation [1], except that we set `-gap 0` in order to disable the merging of called crosslink sites. We used CITS v1.1.2 and the reported p values to rank the sites.

iCount The command-line interface was used to call crosslink sites. This comprised running the modules `iCount xlsites` to detect candidate crosslink sites and quantify corresponding cDNA starts, `iCount segment` to parse a given genome annotation file (Ensembl Release 75 annotations) and obtain corresponding genome segmentations and, finally, running `iCount peaks` to detect sites significantly increased in cDNA starts. As described in Section 3.6.2, iCount can normalize read start counts in a

[1] https://zhanglab.c2b2.columbia.edu/index.php/CTK_Documentation

gene-wise, transcript-wise or transcript feature-wise manner. We choose the transcript feature-wise normalization (`-group_by transcript_id -features CDS intron UTR3 UTR5 ncRNA`), to account for local differences as far as possible. To ensure a fair assessment, we also compared the performance of iCount when using the default gene-wise normalization strategy (see Figure A.3). Apart from that, all modules were run using iCount's default parameters. This comprises the *half-window* parameter used for the moving sum set to 3. Moreover, it is noteworthy that for the evaluation of called crosslink sites we used the provided FDR values. In case one site is reported as significantly crosslinked for multiple transcripts, we choose the highest FDR value to reduce the impact of artefacts caused by incorrectly assigned genomic features (see Section 3.6.2). We used iCount v2.0.0.

Simple threshold Additionally, we applied the simplest possible approach, namely calling all sites with a read start count above a certain threshold. This gives us an understanding of how different methods perform in different scenarios compared to this naive approach. We called sites with a minimum read start count of 5 and used the read start counts as scores.

PureCLIP The software was run using the parameter `-iv` to train the HMM only a subset of the chromosomes, reducing the computational costs. For pooled data we used chromosomes 1 to 3. When using individual replicates we used chromosomes 1 to 6. Beside this, the default settings as described in Section A.12 were used if not stated otherwise. As already done for the results presented in the previous chapter we used PureCLIP v1.2.0.

Detection of binding regions

We compared our performance against the peak-calling methods Piranha [141] and CLIPper [88] as well as against CITS and iCount using settings to merge crosslink sites.

Piranha The tool was run with a p value threshold (`-p`) of 0.001 and a bin size (`-b`) of 20bp (as done in [153]). A drawback of Piranha is that when using BAM files as input, it calls peaks based on the left-most read positions, i.e. for reverse mapped reads peaks are called based on read end sites instead of read start sites (see also [139]). Since for iCLIP and eCLIP data the read starts contain the information about potential truncation events, we preprocessed the data and used BED files with position-wise read start counts as input for Piranha.

Moreover, we incorporated input control data as covariates, to allow a fair comparison against PureCLIP when incorporating the input signal. For that we additionally used the parameter `-l` to convert covariates to log scale. We used Piranha v1.2.1 and ranked the reported regions based on the associated p values.

CLIPper We used the most recent master version of the tool at the time of preparing this document and archived it [2]. We specified the reference genome with `-s hg19`. Since CLIPper calls peaks while using transcript-wise normalization, in case of overlapping transcripts it might report overlapping peaks, from which we choose the one with the lowest p value. It is worth noting that the eCLIP data published by the ENCODE consortium [129] was analysed using CLIPper followed by a post processing step, in which input control experiments were included to compute input normalized p values [143]. Since the latter is not part of the CLIPper software, we do not include this post processing step in the comparison. The regions' reported p values are used for ranking.

CITS We run CITS scripts with the same parameters as for calling individual crosslink sites (see above), except that we enabled the merging of called crosslink sites within a distance of 8 nt by setting `-gap 8`.

iCount Crosslink sites (obtained as described above) were merged using the `iCount clusters` module. To obtain crosslink clusters comparable in length to CITS and PureCLIP, we used the parameter `-dist 8`. Each cluster has an associated score, independent of the FDR values of the individual crosslink sites, which contains information about the cDNA start counts of the crosslink sites.

PureCLIP We applied the same settings as for the crosslink site detection. This comprises the default parameter `-d 8`, causing called crosslink sites within a distance of 8 nt being merged to binding regions. This distance was chosen corresponding to the motif length of PUM2, assuming rather short binding regions for the evaluated proteins.

In the following sections we will present the evaluations based on using simulated data, experimental data for proteins with known binding regions and the agreement of called sites between replicates.

8.2. Evaluation based on simulated truncation-based CLIP data

As described in detail in Section 6.2, we simulated iCLIP data, for which we have given the individual simulated crosslink sites, as well as the simulated binding regions harbouring those. In the following, we will present the precision of the different methods in capturing these simulated sites or regions for different background and target signal characteristics. Note that Piranha and PureCLIP are only run in their basic modes, i.e. without the incorporation of external data as covariates.

[2] tagged as 'evaluation' at `https://github.com/skrakau/clipper/`

8.2.1. Detection of simulated crosslink sites

To evaluate the performance of the compared tools as described in Section 8.1.3, we calculated the precision for each method and each score threshold as the fraction of true positives among the called crosslink sites. The results in Figure 8.2 demonstrate that PureCLIP achieves a higher precision in detecting individual crosslink sites than other tools for all used simulation settings, except one. In particular for the top ranking sites, it has a far better precision compared to the other methods. The only simulation setting where this is not the case, is the one using the highest random noise subsampling rate ($r_s = 0.05$) and no background binding, in which case iCount reaches a higher precision.

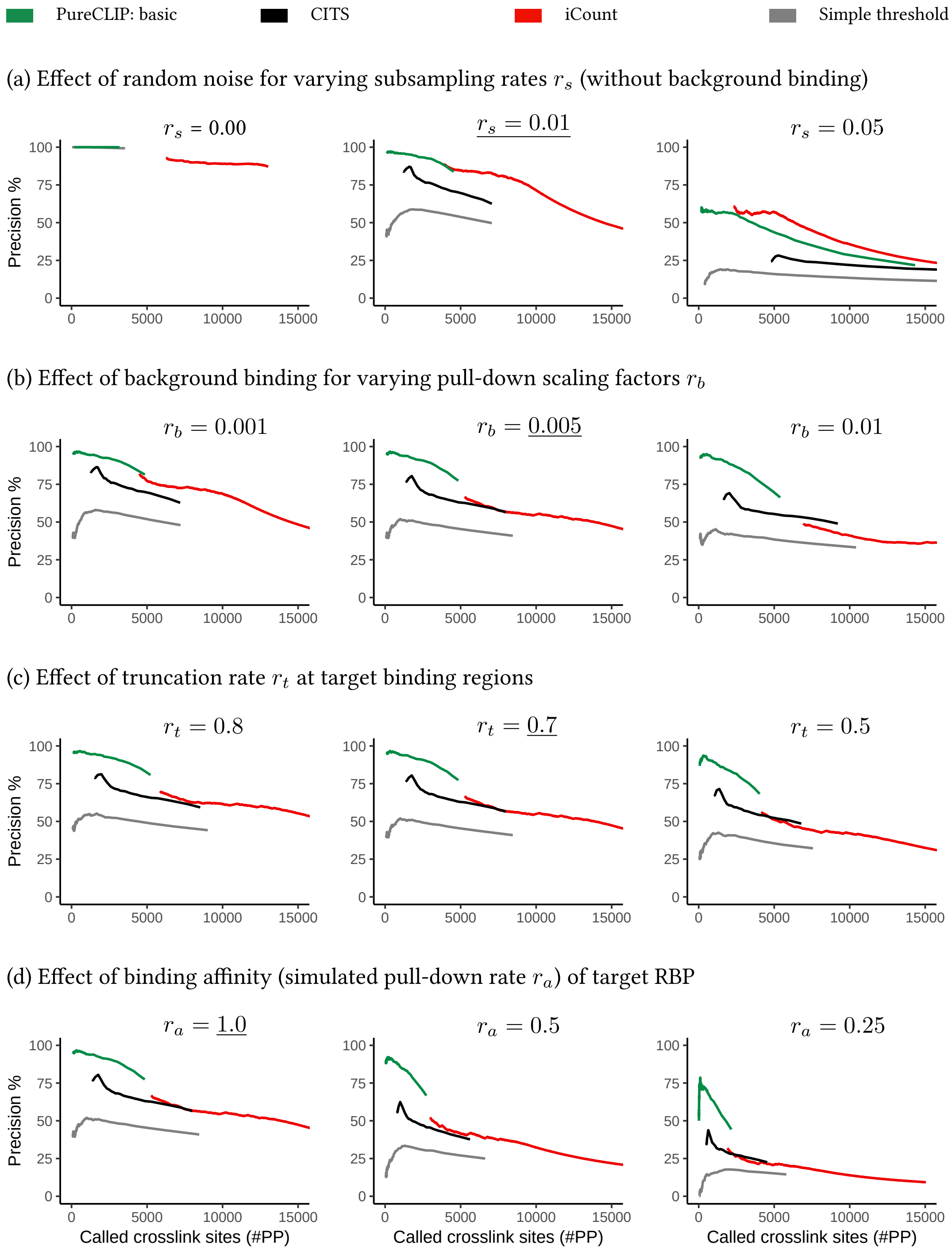
Without random noise or background binding, all methods except iCount reach a precision close to 100% (see Figure 8.2a, left). As expected, with increasing random noise (see Figure 8.2a) and background binding (see Figure 8.2b) we observed a decreased precision for all tools. Interestingly, random noise can have a relatively strong effect on the precision for all methods. This is likely because the randomly subsampled reads also reflect different transcript abundances and artefacts from the RNA-seq data. For example, not properly removed PCR duplicates, mapping artefacts as well as intron-exon junctions within highly abundant transcripts can cause pile-ups of read starts which are then erroneously interpreted as crosslink sites. We expect a similar effect when analysing real iCLIP or eCLIP data in the presence of sticky RNAs or other non-specific background signal. Notably, iCount outperforms all other methods on the simulation setting with the highest amount of random noise, while CITS and PureCLIP reach consistently higher precisions for the dataset with more background binding ($r_b = 0.01$). This demonstrates that iCount performs well in distinguishing target signal from random noise, but less good in distinguishing it from background binding signals containing crosslinking patterns.

When decreasing the truncation rate at target binding regions from 0.8 to 0.5, we observed a moderate decrease in precision comparable for all methods (see Figure 8.2c). Moreover, as expected, we observed a decreasing precision with decreasing pull-down rates for target binding regions – representing lower binding affinities (see Figure 8.2d). When decreasing the target pull-down rate to 25%, the precision of CITS and iCount drops below 50% across all #PPs. In contrast, PureCLIP reaches a precision of up to $\sim$ 75% and even for the highest #PP still maintains a precision of $\sim$ 50%.

We conclude that PureCLIP outperforms the other compared tools in detecting individual simulated crosslink sites over a range of different background noise characteristics and still performs well for datasets with lower binding affinities of the target protein.

8.2.2. Detection of simulated binding regions

Next, we compared PureCLIP's performance in detecting simulated binding regions to that of other peak-calling or crosslink cluster detection methods. For that we compute the precision for each called binding region as the fraction of positions overlapping with a simulated binding region. We then investigated the mean region-wise precision



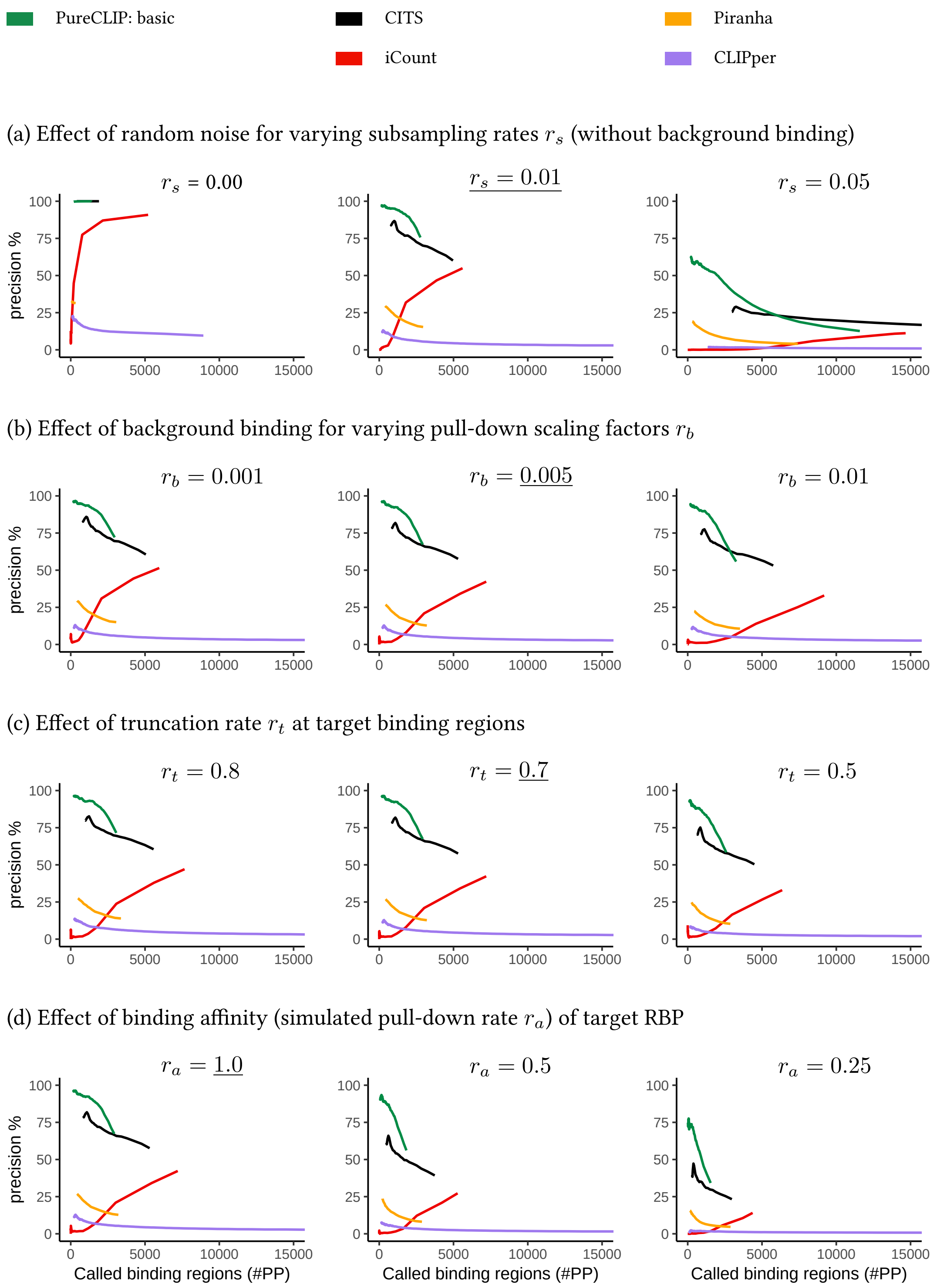
(a) Effect of random noise for varying subsampling rates r_s (without background binding)

(b) Effect of background binding for varying pull-down scaling factors r_b

(c) Effect of truncation rate r_t at target binding regions

(d) Effect of binding affinity (simulated pull-down rate r_a) of target RBP

Figure 8.2.: Evaluation of crosslink sites called by different methods on simulated data with different background and target signal characteristics. The default parameters are underlined and used to generate the data for the remaining results, if not stated otherwise.



(a) Effect of random noise for varying subsampling rates r_s (without background binding)

(b) Effect of background binding for varying pull-down scaling factors r_b

(c) Effect of truncation rate r_t at target binding regions

(d) Effect of binding affinity (simulated pull-down rate r_a) of target RBP

Figure 8.3.: Evaluation of binding regions called by different methods on simulated data with different background and target signal characteristics. The default parameters are underlined and used to generate the data for the remaining results, if not stated otherwise.

for each method and each score threshold.

In general the results shown in Figure 8.3 demonstrate that PureCLIP recovers simulated binding regions with a higher precision compared to other tools over a large range of simulation settings. Only for the highest rate of random noise (see Figure 8.3a, right) and the highest pull-down rate for background binding regions (see Figure 8.3b, right), CITS reaches a higher precision compared to PureCLIP and only for higher numbers of PPs. That this is observed only when evaluating called binding regions and not for individual crosslink sites indicates a difference between CITS and PureCLIP regarding the merging of crosslink sites to binding regions. Since both methods merge sites within a distance of 8 nt, we can infer that the distribution across the transcriptome somehow differs. Furthermore, we observed that for all simulated datasets the crosslink site or cluster detection methods achieve a far higher region-wise precision compared to the peak-calling methods Piranha and CLIPper. Since Piranha uses bin-wise read start counts and CLIPper uses whole reads to define binding regions, these are expected to be longer and less precise compared to regions obtained from crosslink detection methods. We will investigate the lengths of the called binding regions in more detail based on experimental eCLIP data in Section 8.3.3.

For iCount it can be seen that, in contrast to the crosslink site evaluating (see Figure 8.2), the precision increases with the number of called binding regions. The reason for this is that individual iCount crosslink sites are ranked based on their FDR, while binding regions obtained through clustering of such sites are ranked simply based on their read start counts. The results demonstrate that the latter strategy scores non-specific background signals higher than target-specific signals.

Recall that this evaluation based on the described simulation framework assumes relatively short target-specific binding regions corresponding to the PUM2 sequence motif, and read starts outside of these regions to be caused either by non-truncated cDNAs or cDNAs truncated at non-specific sites. Thus the presented results cannot necessarily be transferred to proteins which do not follow these characteristics, but are known to rather slide along the RNA or bind in clusters within a longer region. Nevertheless, for the simulated data, PureCLIP outperforms the existing methods in precisely detecting the simulated binding regions for a large range of simulation settings.

8.3. Evaluation on experimental iCLIP and eCLIP data

We used the PUM2, RBFOX2 and U2AF2 eCLIP and iCLIP datasets (see Section 6.1) to assess the performance of the different tools in calling crosslink sites or binding regions that correspond to the proteins' known binding regions, whose characteristics we described in Section 8.1.1.

In the following chapter we will first define the bona fide binding regions and then comprehensively evaluate the performance of the different tools in detecting those for the different datasets, first on the crosslink site and then on the binding region

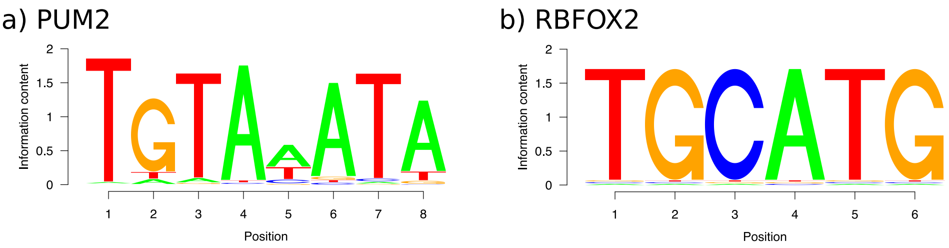


Figure 8.4.: Sequence motifs used for PUM2 and RBFOX2 to define bona fide binding regions, retrieved from the ATtRACT database [54].

level. Additionally, we will present the evaluation based on the agreement of reported crosslink sites between replicates.

8.3.1. Definition of bona fide binding regions

For the following evaluations we define the *bona fide* binding regions for PUM2 and RBFOX2 using their known sequence motifs (see Figure 8.4) [3]. For U2AF2, a sequence motif based definition of the binding region is not applicable, since U2AF2 binds to poly(U) tracts which coincide with non-specific CL-motifs. In this case, we make use of its known predominant binding region $\sim$ 11 nt upstream of 3' splice sites [155], based on Ensembl Release 75 annotations.

8.3.2. Detection of individual crosslink sites within known binding regions

To gain insight into the precision of the different tools, we plot the distribution of the called crosslink sites around the bona fide binding regions, while using the same number of x top-ranking calls for comparison, where equally scoring sites were shuffled randomly. For each crosslink site the distance to the closest motif start site or 3' splice site was used. We refer to these plots as *crosslink site maps* (see Figure 8.5, left).

Furthermore, for a broad range of different score thresholds we assessed the methods' precisions in relation to the number of predicted positives as described Section 8.1.3. Called crosslink sites within bona fide regions are defined as true positives. More precisely, for PUM2 and RBFOX2 the precision was defined as the fraction of called crosslink sites that are located within 2 bp of a motif occurrence. For U2AF2 the precision was defined as the fraction of called sites that are located 11 ± 4 nt upstream of a 3' splice site. Note that the numbers of called crosslink sites obtained with the compared methods differ by an order of magnitude (see Table A.1). iCount is by far the most sensitive among all tools and for PUM2 eCLIP data, for example, the achieved #PP range, i.e. defined by the highest and lowest score, does not overlap with the #PP ranges obtained with PureCLIP or CITS. Within this thesis we opt for high precision

[3] Genome-wide motif occurrences were computed using FIMO [55] (--thresh 0.001 --norc).

over sensitivity and thus focus in the following on the lower #PP range obtained with CITS, PureCLIP and the simple threshold method. For comparison the results for the full range are shown in Figure A.3.

General remarks

For the PUM2 data, the crosslink site maps depicted in Figure 8.5a (left) reveal for all methods except iCount an accumulation of called sites at the 5' end of PUM2 motif occurrences and another slightly weaker accumulation towards the 3' end. For RBFOX2 eCLIP data we observe an accumulation of called crosslinks at the two guanines within the motif (see Figure 8.5b, left). These crosslinking patterns are in agreement with previous findings [143, 149] and, since crosslinks preferentially occur at uridines and not at guanines, are most likely caused by target-specific protein-RNA interactions.

For U2AF2 all tools show an enrichment of called crosslink sites at the known binding site $\sim$ 11 nt upstream of 3' splice sites (see Figure 8.5c and d, left). It should be noted that U2AF2 preferentially binds to poly(U) motifs, which largely coincide with the top CL-motifs. Therefore, we expect an increased crosslinking efficiency for U2AF2 at target-specific binding regions compared to other proteins. Indeed we can observe a higher precision for U2AF2 eCLIP and iCLIP data compared to PUM2 and RBFOX2 eCLIP data for all tools (see Figure 8.5, right).

PureCLIP's performance without incorporating external data as covariates in comparison to other methods

We first investigated PureCLIP's performance in basic mode, i.e. without the incorporation of any covariates, in comparison to the other tools. When looking at the crosslink site maps, we observe that PureCLIP calls a higher fraction of sites within the bona fide binding regions in PUM2 eCLIP and U2AF2 eCLIP and iCLIP data compared to all other methods (see Figure 8.5, left). Furthermore, for these datasets PureCLIP reaches a higher precision than any other method consistently across its entire #PP range, as shown in Figure 8.5 (right). For RBFOX2 eCLIP data, CITS reaches a higher precision.

Among the existing methods, CITS performs best in detecting individual sites within the bona fide regions and generates a similar crosslink site distribution as PureCLIP. In contrast, for the evaluated #PP range, the precision of iCount is for the most part only slightly above the precision of the simple threshold method. Moreover, the crosslink site maps show that for iCount the distribution of called sites differs notably from all other methods. Especially for RBFOX2 it does not recover the predominantly crosslinked sites within the sequence motif (see Figure 8.5b, left). The main reason for this is likely that in its default setting iCount not only considers the read starts for each position, but also those that occur 3 nt up- and downstream (see Section 3.6.2). However, the predominant crosslink sites reported for RBFOX2 are 4 nt apart and, consequently, cannot enhance each other. On the other hand, non-specific crosslink sites, for example within CL-motifs, might enhance each other and may thus be more likely to be called by iCount. Furthermore, for the RBFOX2 eCLIP data, iCount does not provide different ranks for the top $\sim$ 15,000 sites, which all have an associated FDR of 0.0. Therefore,

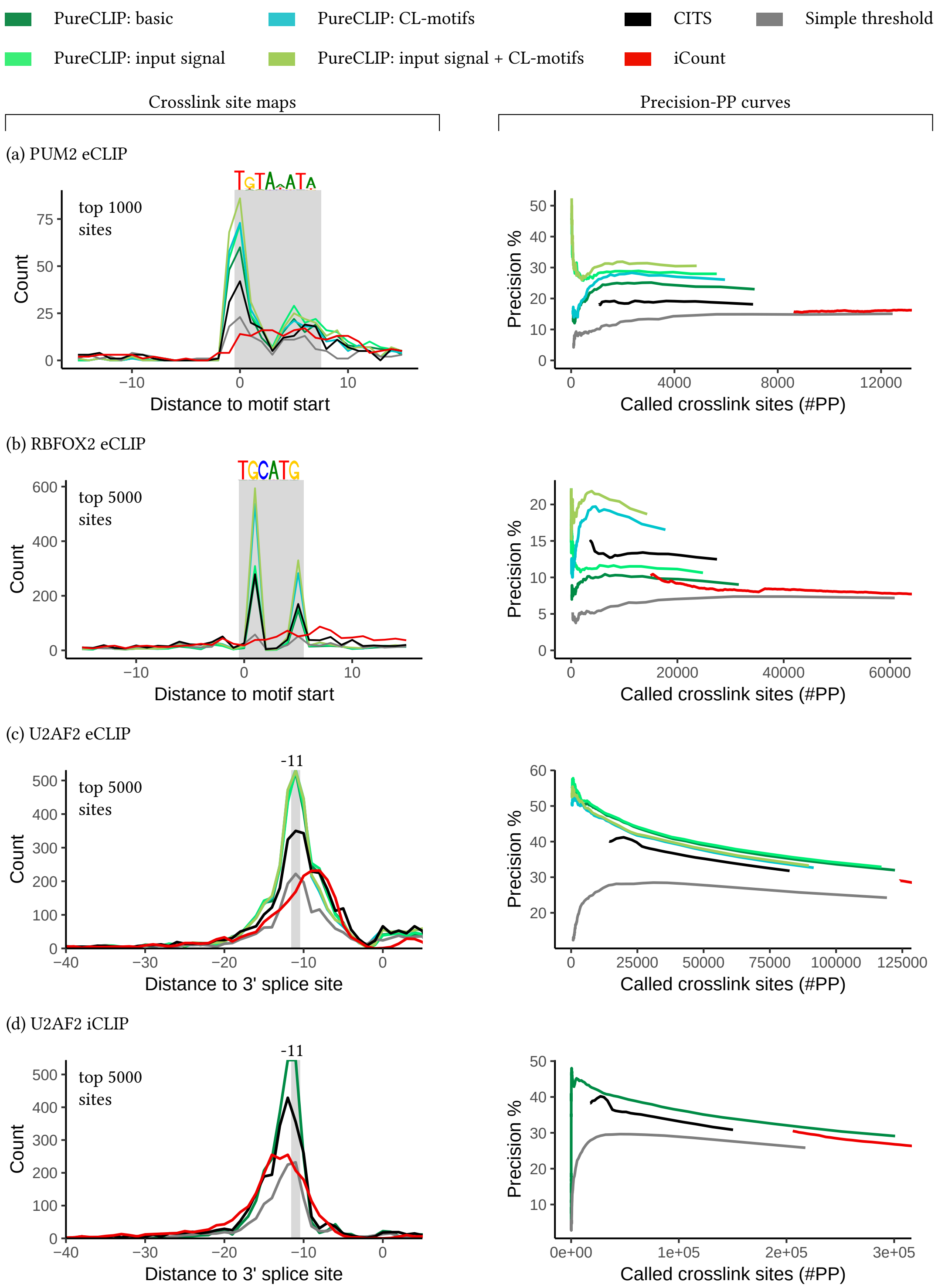


Figure 8.5.: Performance in detecting crosslink sites within bona fide binding regions (highlighted in gray). **Left:** Distribution of the of top ranking sites around the closest motif start position or 3' splice site. **Right:** Precision of the called sites in relation to the number of predicted positives.

the top 5000 sites where randomly sampled from this set. This might be an additional reason why a relatively low fraction of the top sites is located within the bona fide binding regions. Nevertheless, not providing different ranks for such a high number of high scoring crosslink sites is a clear drawback of the method.

Incorporation of input control data and CL-motifs greatly improves PureCLIP's crosslink site detection

We then investigated the performance of PureCLIP when additionally incorporating covariates to correct for non-specific background binding or the CL-bias. Compared to PureCLIP in basic mode, the results show that the incorporation of input signal improves the precision for PUM2 and RBFOX2 eCLIP datasets over the entire range of #PPs (see Figure 8.5a-c, right), in particular for the top ranking sites. For U2AF2 eCLIP data the results to not change much. One possible reason for this is the relatively high crosslinking efficiency of U2AF2, so that background binding is less of an issue. Nevertheless, these results demonstrate that artefacts caused by non-specific background signal can cause top scoring sites and highlight the need to correct for such.

Alternatively, incorporating CL-motif scores also greatly improves the precision for PUM2 (see Figure 8.5a) and in particular for RBFOX2 eCLIP data (see Figure 8.5b). Moreover, we can see that for U2AF2, whose sequence motif coincides with CL-motifs, the performance of PureCLIP is robust and not impaired by the incorporation of CL-motif scores. Altogether, we could see that when incorporating CL-motifs PureCLIP consistently performs better than all other crosslink sites detection methods for all datasets, even without considering input control data (see Figure 8.5a-c). Note that for the U2AF2 iCLIP data, no covariates are incorporated (see Figure 8.5c), as no matching input control dataset is available.

Interestingly, the simultaneous incorporation of input signal and CL-motif scores can improve the precision of PureCLIP even further (see Figure 8.5a, b).

Characteristics of the called crosslink sites

To gain a better understanding of the behaviour of the different tools and the effect of the different biasing factors on their performance, we explored some characteristics of the called sites regarding their pulled-down fragment densities as well as their location within different bias prone regions. The fragment densities reflect both the local binding affinities and the local transcript abundances, and we explore them to understand how highly regions have to be covered to be called by the different methods. Figure 8.6 shows these characteristics for the PUM2 eCLIP data as an example, equivalent plots for RBFOX2 and U2AF2 eCLIP data are shown in Figures A.4 and A.5.

Although these characteristics depend on the target protein, the results demonstrate that there is a large fraction of sites with high fragment densities called by CITS and the simple threshold method that is not called by PureCLIP (see Figures 8.6a and A.4a). The reason for this is most likely that PureCLIP is designed to capture the strongest protein-RNA interaction footprints, while accurately modeling the crosslink truncation patterns, rather than detecting the highest peaks. The crosslink sites called

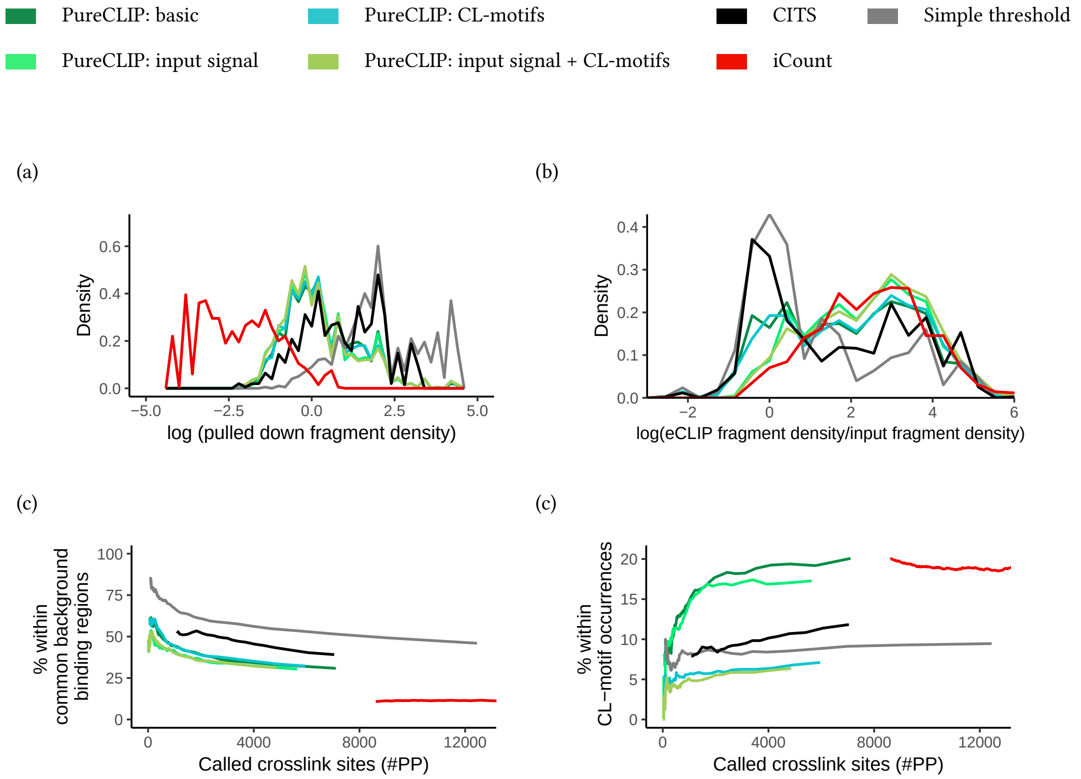


Figure 8.6.: Characteristics of the crosslink sites reported by the different methods for PUM2 eCLIP data. **a)** Distribution of the log fragment densities and **b)** of the log-fold fragment density enrichments over the input for the top 1000 called sites. **c)** Fraction of called sites located within common background binding regions and **d)** within CL-motif occurrences for different numbers of predicted positives.

by iCount, on the other hand, have the lowest fragment densities. Furthermore, CITS and the simple threshold method call a certain fraction of crosslink sites in regions with fragment densities not enriched over the input signal, likely caused by non-specific background binding (see Figure 8.6b). iCount and PureCLIP call notably less sites in those regions.

We also investigated how many of the sites called by each method are located within known common background binding regions. These background binding regions were taken from [119], as already done for the simulation. In this analysis, we used only regions with observed background binding in at least 6 different CLIP-seq datasets, and extended them upstream and downstream by 200 bp. Such regions are prone to false positive calls caused by highly abundant RNAs as well as by highly abundant background proteins binding these RNAs. The *simple threshold* method clearly calls more crosslinks at sites that are located within known background binding regions compared to other methods (see Figure 8.6, c). In contrast, PureCLIP calls the lowest fraction of sites within such background binding regions in all settings.

Moreover, we were interested in how much each of the methods suffers from the CL-bias. CL-motif occurrences were obtained with FIMO [55] (as done for the computation of CL-motif scores, see Figure A.5). Interestingly, for all datasets the highest fraction of called sites within CL-motif occurrences is called by PureCLIP in modes not correcting for the sequence bias (see Figure 8.6, d). A relatively high bias is expected, since PureCLIP is designed to capture cDNA truncation footprints. Consequently, if the data contains a large fraction of non-specific crosslinks, either solely caused by background binding or by co-purified proteins bound to the same RNA fragments at sites with higher crosslinking efficiencies (see Section 3.3.4), PureCLIP also detects those. For RBFOX2, which has a sequence binding motif distinct from reported CL-motifs, we observed that up to 40% of the sites called by PureCLIP in basic mode overlap with CL-motif occurrences (see Figure A.4). This explains the relatively low precision of this setting for RBFOX2 data and highlights the necessity to correct for this bias in the model. However, in general high sensitivity for crosslink patterns is indented, because we aim to also capture protein-RNA interactions within lowly abundant transcripts. Importantly, the incorporation of CL-motifs into the PureCLIP model drastically reduces this bias (see Figure 8.6, d), enabling a high precision across all datasets.

8.3.3. Detection of known binding regions

Besides single-nucleotide crosslink sites, we investigated the performance of PureCLIP in calling binding regions (see Section 8.1.1), and compared it against other existing tools designed for peak-calling or able to detect crosslink clusters. For this purpose we show the distribution of called regions around the protein's bona fide binding regions, using for each tool the same number of top-ranking regions (see Figure 8.7, left). Note that here for each position the density of binding regions covering it is plotted for each tool. We use the density, because we aim to recover the true binding region at a high resolution and will refer to this measurement as the *binding region coverage density*.

The resulting plots will be referred to as *binding region maps*.

Similar to the evaluation of called crosslink sites, we aimed to assess the methods' precisions in relation to the number of predicted positives by applying different score thresholds (see Section 8.1.3). Since we do not want to classify called regions simply as either true or false positives, we need a measurement additionally representing the region-wise precision. For PUM2 and RBFOX2 we assume that for a called region, a higher target motif density corresponds to a higher precision. However, we do not know if the protein only binds sites within the motif or also within close proximity. We therefore make use of a smoothing approach to score sites located closely to the motif higher than sites further away. We computed region-wise motif scores as follows:

1. Genome-wide motif occurrences were detected and scored using FIMO [55] (`-thresh 0.005`).

2. The scores were smoothed using a kernel density estimation (KDE) [43] with a Gaussian kernel function and a default bandwidth of 5 nt. KDE values within one motif occurrence of the same score were flattened by using the minimum value.

3. A score $s_{\text{motif}} = \text{mean}(KDE)$ is assigned to each reported binding region.

4. For a set of called binding regions we then use the mean score as an estimate for the precision.

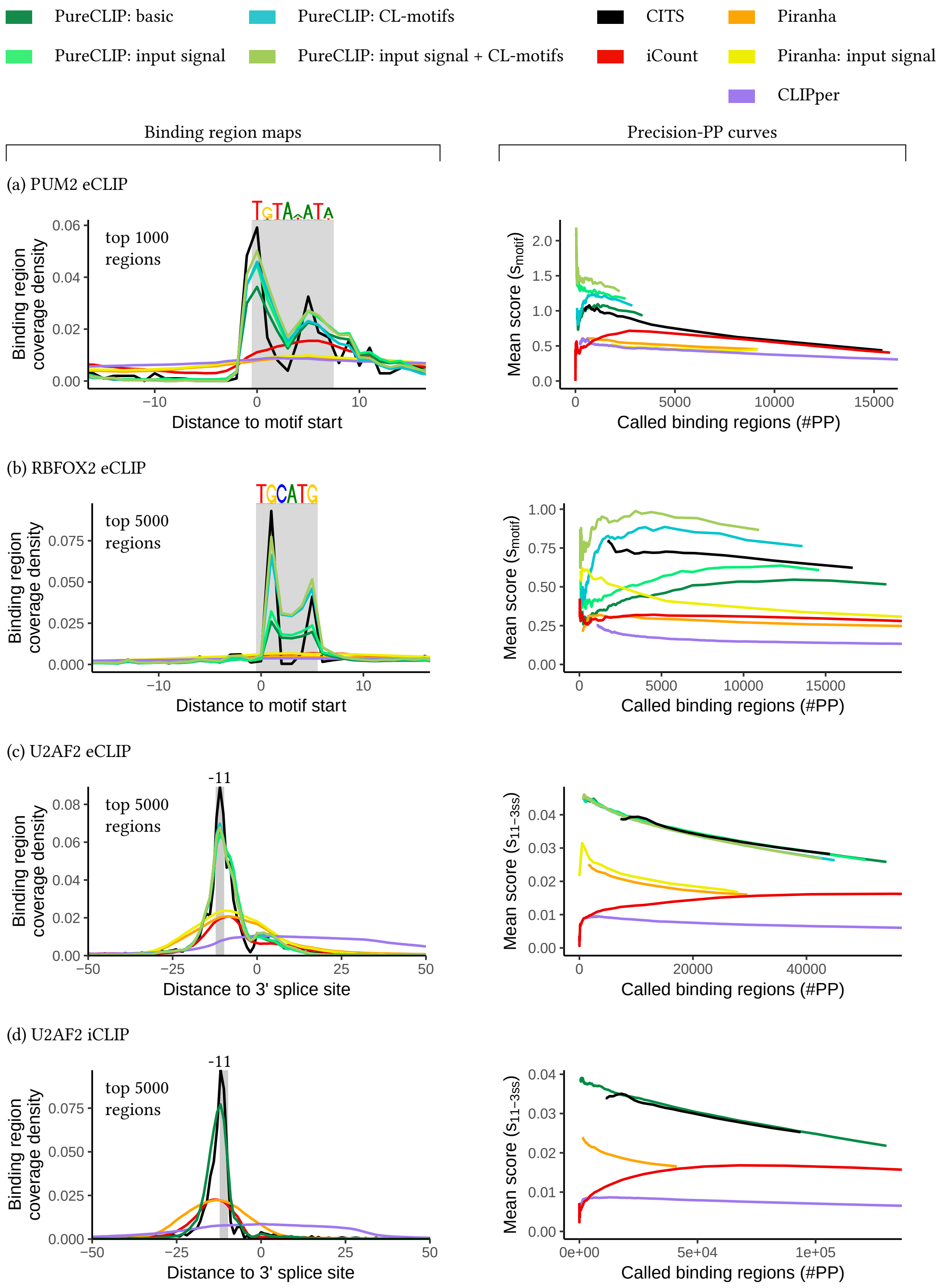
Figure 8.7a and b (right) show the corresponding (estimated) precision-PP curves for the PUM2 and RBFOX2 eCLIP data.

For U2AF2 a similar region-wise score is computed. Here we assign a value of 1 to positions 11 nt upstream of 3' splice sites and again apply a smoothing. We refer to this score as s_{11-3ss}. The results are shown in Figure 8.7c and d (right) for a KDE bandwidth of 5 nt and in Figure A.6 for a bandwidth of 10 nt.

PureCLIP's performance in comparison to other methods

The binding region maps depicted in Figure 8.7 (left) reveal similar distributions for CITS and PureCLIP around bona fide binding regions as previously observed for individual crosslink sites, with strong accumulations within the bona fide regions. While for CITS these accumulations still occur at individual sites, PureCLIP's binding regions cover larger parts of the bona fide binding regions, which becomes particularly apparent for RBFOX2 (see Figure 8.7b, left). In contrast, Piranha, CLIPper and surprisingly also iCount show a much broader positional distribution of binding regions around the bona fide regions, with CLIPper showing the broadest distribution for all datasets. The latter is expected, since CLIPper is the only tool using whole reads and not read starts for the analysis.

The corresponding precision-PP curves estimated using the mean region-wise scores s_{motif} and s_{11-3ss} demonstrate that CLIPper consistently achieves the lowest score, while CITS and PureCLIP reach the highest scores (see Figure 8.7, right). For Piranha



(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

(d) U2AF2 iCLIP

Figure 8.7.: Performance in detecting binding regions based on the agreement with bona fide binding regions (highlighted in gray). **Left:** Binding region coverage density around target motif start positions or 3' splice sites. **Right:** Mean binding region scores in relation to the number of predicted positives.

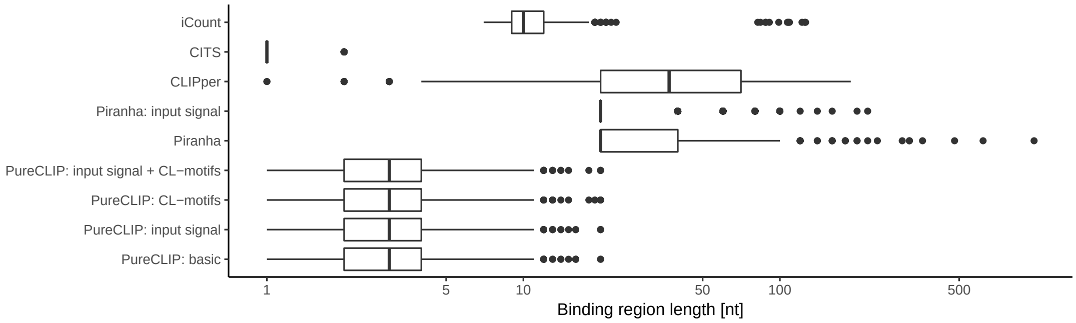


Figure 8.8.: Length distribution of the top 1000 called binding regions by each method for PUM2 eCLIP data.

we observed that the incorporation of input signal improves its precision, mostly for the RBFOX2 eCLIP data. Nevertheless, in general it still reaches lower mean scores than CITS and PureCLIP, which can be expected as it calls regions of at least 20 bp (see Section 8.1.4). For the PUM2 and RBFOX2 eCLIP data PureCLIP outperforms all other methods when incorporating CL-motifs, while CITS and PureCLIP perform comparably for the U2AF2 eCLIP and iCLIP data. In this context it is noteworthy that the applied scores, although penalizing positions outside of bona fide binding regions, do not distinguish between called regions covering the whole bona fide region and regions covering only parts of it. This directly arises from the objective to estimate the region-wise precisions, which provides valuable information about how much we can trust the called regions.

Characteristics of the called binding regions

To better understand the behaviour of the different tools used for peak-calling and crosslink cluster detection, we explored the length distributions of the called binding regions for each method, shown in Figure 8.8 for PUM2 (for equivalent plots for RBFOX2 and U2AF2, see Figure A.8). Recall that for the three crosslink site detection tools CITS, iCount and PureCLIP, called sites within a distance of 8 nt were merged to obtain binding regions. Accordingly to the previously shown results, we observed that on average CLIPper binding regions are the longest, while CITS binding regions are the shortest. Interestingly, the vast majority of CITS binding regions spans only one position. This indicates that CITS strongly favours individual crosslinks sites, or in other words, that strong crosslinking patterns at nearby positions prevent each other from being called by CITS. In comparison, the binding regions of PureCLIP in all four settings have a median length of 3 nt, while iCount calls regions with a median length of 10 nt. The longer regions of iCount can at least partly be explained by that fact that is uses a moving sum on the read start counts computed for 7 nt windows to detect crosslink sites. Obviously, for CITS, iCount and PureCLIP, beside the distribution of the individual called crosslink sites, the lengths of the called regions also depend on the distance parameter used to merge crosslink sites. In any case, this remains to be explored, although it is not trivial to infer the optimal setting, as this highly depends on the objective, for example, if the binding of multiple proteins within close proximity or of multiple RBDs is of interest.

Additionally, similarly to the evaluation of called crosslink sites, we explored selected characteristics of the called binding regions regarding their fragment densities and their location within different bias prone regions (see Figure A.7). However, since these characteristics strongly depend on the length distribution of the called binding regions and are difficult to compare, we omit these details here.

8.3.4. Agreement of called crosslink sites between eCLIP replicates

Since the previously described evaluation strategies have the limitation that the exact binding regions and crosslink sites remain unknown, we additionally aimed to assess the performance of the different methods independently of bona fide binding regions. For this purpose, we explored each method's precision based on the agreement of called crosslink sites between replicates, assuming that target-specific binding events are likely to be observed in both replicates. Note that we use this measurement only for the evaluation of called crosslink sites, but not for binding regions, since for the latter it would be additionally influenced by the methods' length distributions of binding regions and thus not suitable for comparison. Using the replicate agreement as an estimate for the precision, we can again create precision-PP plots. We applied all crosslink detection methods to the individual replicates and measured for each score threshold how many of the x called crosslink sites in one replicate overlap with the x top ranking crosslink sites in the other replicate.

However, non-specific signals caused by highly abundant background proteins binding to highly abundant RNAs or by the CL-bias are likely to be reproducible as well, and thus contribute to the measured replicate agreement. We therefore count only those sites to the agreement that are also enriched over the input and located outside of regions that are known to be prone to background binding (as published in [119]). With this we avoid overestimating the precision of methods that consistently call false crosslink sites in both replicates due to systematic, reproducible biases. This potentially also excludes a certain number of true positives that cannot be distinguished from non-specific background noise, but we expect this to affect all methods more or less equally and thus still allow for a fair comparison. We refer to this measurement as the *bias-corrected replicate agreement.*

To further prevent a contribution of common non-specific crosslinks, for PUM2 and RBFOX2 we counted only sites to the bias-corrected replicate agreement that are not located within CL-motif occurrences. Since the target motifs of these two proteins are clearly distinct from CL-motifs, we expect that we do not miss relevant target-specific sites by doing so. For U2AF2, whose sequence motif coincides with a common CL-motif, we omit this step, since we would loose a large fraction of true positives. Furthermore, in comparison to other proteins, U2AF2 has most likely a higher crosslinking efficiency at target-specific interactions sites and, as a consequence, less biased crosslink calls (see Section 8.3.2). The U2AF2 iCLIP data is excluded from this evaluation, since no input control experiment is available and thus the bias-corrected replicate agreement can not be computed.

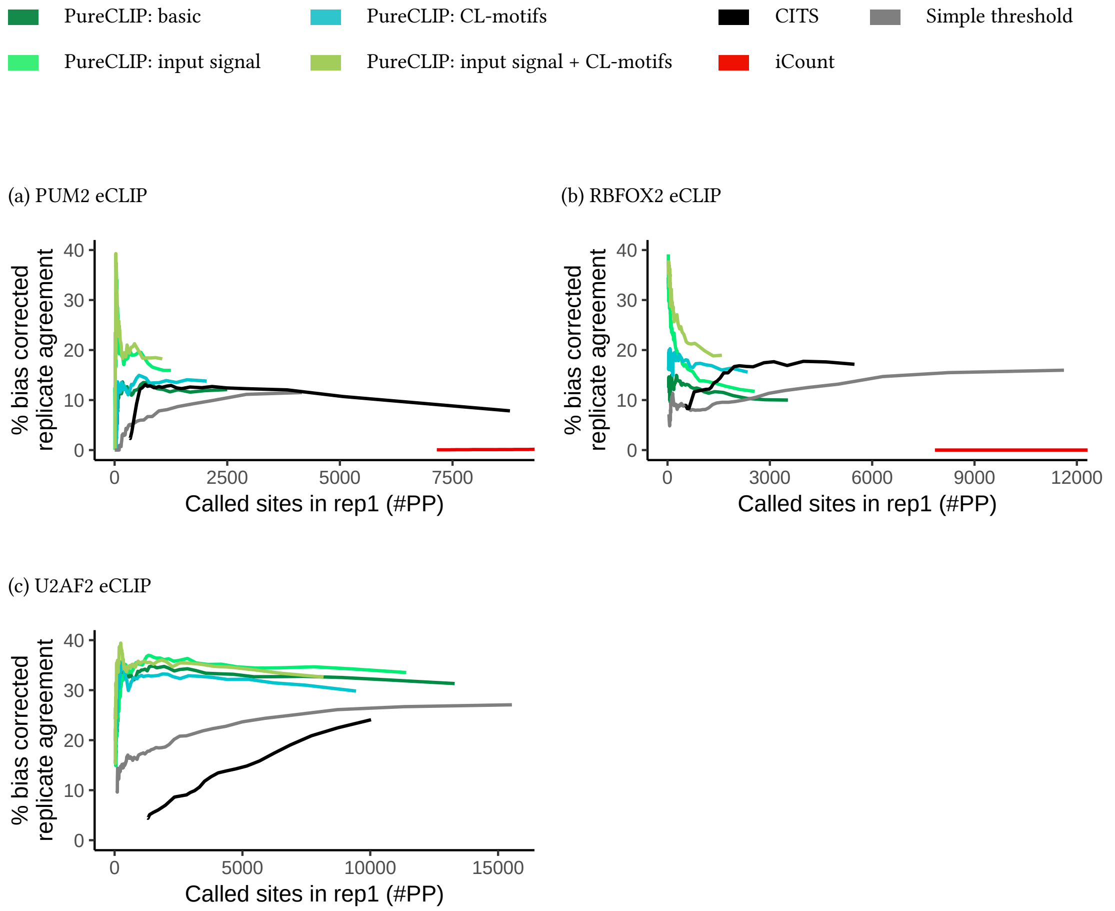
In summary, to compute the bias-corrected replicate agreement, we define sites that (1) have sufficient enrichment over the input signal, (2) are not located in common background regions or (3) in CL-motifs (for PUM2 and RBFOX2). For (1) we chose an individual log fold-change threshold for each protein dataset, based the distribution of log fold-change values at bona fide crosslink sites compared to the distribution at all sites with at least one read starting (see Figure A.10). In this context, a bona fide site is a site that is called by PureCLIP (in basic mode) and located within the target sequence motif. We then chose a threshold with the aim to separate the two distributions. Common background binding regions as well as CL-motif occurrences were obtained as previously described (see Section 8.3.2 and 5.4.2).

PureCLIP has a higher bias-corrected agreement of called sites between eCLIP replicates compared to other methods

Interestingly, when not accounting for any biases the *simple threshold* method achieves by far the highest agreement for all three eCLIP datasets compared to other methods (see Figure A.9). However, we found that the *simple threshold* method also calls the most crosslinks at sites where the fragment density is not enriched over the input and at sites that are located within known background binding regions (see Figures 8.6, A.4 and A.5). These results underline that beside target-specific binding regions or crosslink sites, transcriptomic regions that are particularly prone to false positive calls contribute to the raw replicate agreement.

When correcting for biases, our evaluations show that compared to the other crosslink detection methods PureCLIP has a higher replicate agreement for the top ranking sites in all four settings (in basic mode and when correcting for biases) and over all three eCLIP datasets, as shown in Figure 8.9. This is a particularly valuable result, since we assume that the strongest target-specific protein-RNA interactions cause signals in both CLIP replicates, and are thus ideally part of the top ranking calls of both replicates. Furthermore, the performance of PureCLIP in basic mode is in general comparable to that of other methods, while when incorporating input signal and CL-motifs PureCLIP strictly outperforms all other methods. Similar to the results on bona binding regions (see Figure 8.5b, right), we observed that PureCLIP reaches a relatively low precision for RBFOX2 when not correcting for biases, which notably increases when incorporating CL-motifs. While the individual use of the covariates already improves the agreement, the best results are again obtained when both are incorporated simultaneously.

Beside the described biases, there might be a remaining bias affecting the replicate agreement caused by different fragment densities. We expect that crosslink sites with a high fragment density are more likely to be called in both replicates. However, in comparison to CITS and the simple threshold method PureCLIP calls sites with comparable or lower fragment densities (see Figures 8.6a, A.4a and A.5a) and thus a potentially remaining bias would likely lead to an under-estimation of PureCLIP's



(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

Figure 8.9.: Bias-corrected replicate agreement on PUM2, RBFOX2 and U2AF2 eCLIP datasets. For each given number x of called sites (corresponding to a certain score threshold) in one replicate, we report the percentage that was also found within the x top ranking called sites of the other replicate.

performance.

9. Additional evaluations

In the previous chapter we demonstrated the high precision of PureCLIP both in calling crosslink sites and binding regions in comparison to other methods by using different evaluation strategies. We will now first show the run time and memory requirements in comparison to the other methods. Then we will review PureCLIP's performance over different settings and show that it also correctly captures the binding region for PTBP1, which is known to create longer clusters of crosslink sites. We will furthermore discuss a potential incorporation of RNA-seq data for normalization, which could be useful, for example if no input control data is given. We show that PureCLIP can incorporate this type of data and discuss some of the pitfalls and challenges that should be considered by the user. Finally, we will show that the individual incorporation of CLIP replicates can further boost PureCLIP's precision. For the sake of conciseness, we will mostly limit the following results to the crosslink site evaluation strategy based on bona fide binding regions as described in Section 8.3.2 and consider only CITS for comparison, which performs best among the other existing crosslink detection methods. Moreover, since the performance of all compared methods is rather similar for the U2AF2 eCLIP and iCLIP datasets, we will omit them here.

9.1. Run time and memory requirements

In Section 5.6.4 we discussed the computational complexity characteristics of the PureCLIP method. In Table 9.1 we now present its actual run time and memory requirements of for the RBFOX2 data, which is the largest among the three analysed eCLIP datasets (see Table 6.1), in comparison to the other methods. All tests were carried out on systems with 80 CPU cores (Intel Xeon E7-8891 at 2.80GHz) and 1 TB RAM running Linux.

We observed that Piranha has by far the lowest run time and memory consumption, both when comparing methods that incorporate covariates and methods that do not. The reason for this is its relatively simple model, which models bin-wise read start counts using one (background) distribution. CITS achieves the second lowest run time and memory consumption. It is noteworthy that CITS, iCount and CLIPper all perform permutation tests, which likely constitute the largest part of their run times (see Section 3.6.2). However, while CITS performs these permutation tests on clusters of read starts, iCount and CLIPper use whole transcripts or genes, which likely explains their larger run times to some degree. When running with only one thread, CLIPper has the longest run time and PureCLIP the second longest. PureCLIP has the highest memory consumption, which is, as already discussed, mainly caused by multiple

Table 9.1.: Run times and memory consumption of compared methods on RBFOX2 eCLIP data. Methods supporting multi-threading were run using 1, 10, 20 and 40 threads. For comparison, we highlighted for each method the lowest achievable run time and memory consumption.

	run time [min]				*peak memory [GB]*			
# threads method	1	10	20	40	1	10	20	40
without covariates								
CITS	40.17	-	-	-	2.31	-	-	-
iCount	405.52	-	-	-	15.51	-	-	-
Piranha	1.18	-	-	-	2.10	-	-	-
CLIPper	3,114.71	344.25	196.27	165.74	6.28	10.58	17.85	32.20
PureCLIP	835.18	130.24	91.59	65.28	25.37	33.97	33.12	38.33
with covariates								
Piranha: input signal	26.25	-	-	-	2.28	-	-	-
PureCLIP: input signal + CL-motifs	1,779.68	236.6	154.28	100.01	30.97	38.20	40.30	37.31

position-wise probability values that have to be stored. Importantly, PureCLIP's run time can be notably reduced when using multi-threading, resulting in run times lower than those of iCount. Moreover, while in general requiring a relatively large amount of memory, when increasing the number of threads the memory consumption increases only moderately.

9.2. PureCLIP's performance for different settings

9.2.1. Validity of model choices

As described in Section 5.3, only covered regions with more than one read start are considered within the model. Moreover, the gamma and the binomial probability parameters, modelling the fragment densities and read start count emission probabilities, respectively, are only fitted to sites with at least one read start (see Sections 5.3.1 and 5.3.2). With this we aimed to reduce the computational costs and, additionally, to improve the method's robustness by reducing the impact of noise.

To test the validity of these model choices we investigated their influence on PureCLIP's precision, separately for the gamma and the binomial distributions. For the sake of conciseness, we summarize in the following the main findings and show the detailed results in Appendix A.9. For the gamma distributions, we observed that when not incorporating input control data, PureCLIP's precision is robust for the different model choices. However, when incorporating input signal, both including singleton read starts as well as including all positions for the parameter estimation lowers PureCLIP's precision notably. This indicates that the learning of the GLM regression coefficients α

is impaired by a large number of sites that have very low fragment densities as well as a lower correlation between the target and the input signal. Similarly, we observed that for the binomial distributions PureCLIP reaches a higher precision when using only positions with at least one read start for parameter estimation [1]. In summary, beside the reduced computational cost, the precision results for all three eCLIP datasets demonstrate the superiority of these model choices over the naive approach of using all positions.

9.2.2. Influence of training set

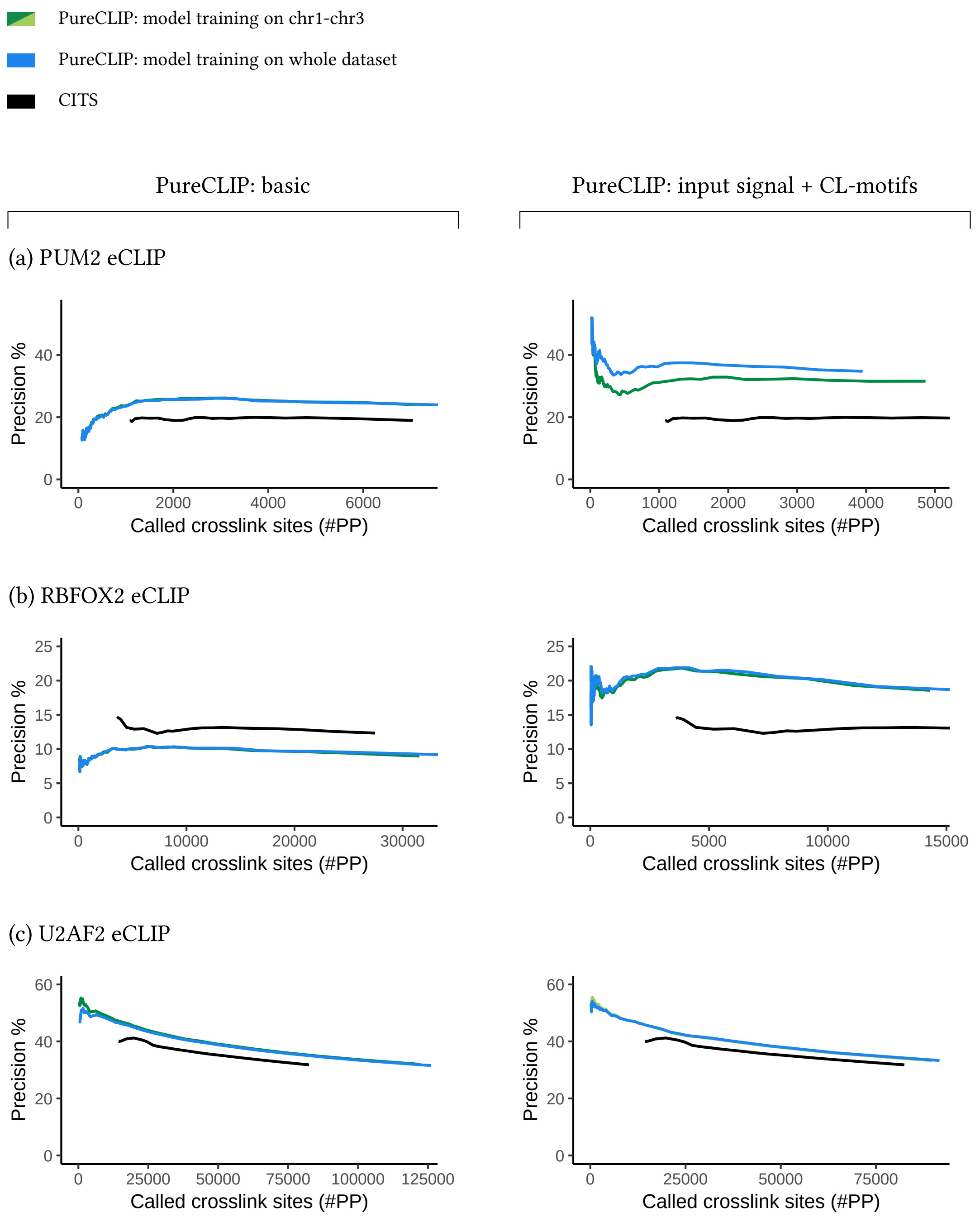
Throughout the previously shown evaluations, PureCLIP's model was trained only on a subset of the provided data, i.e. on chromosomes 1 to 3 for pooled datasets. This choice was made mainly due to run time considerations. We found that the performance of PureCLIP in this setting is surprisingly robust in comparison to training on the full dataset, as shown in Figure 9.1. Only for PUM2 eCLIP data – which is the smallest among the analysed datasets – and when additionally incorporating input signal and CL-motifs, PureCLIP reaches a higher precision when training on the whole dataset. However, even when training the model on the subset, PureCLIP still strictly outperforms all other compared methods in calling crosslink sites (see Figure 8.5a) and binding regions (see Figure 8.7a) for this dataset. It is noteworthy that when using the whole dataset for model training, for RBFOX2 eCLIP data PureCLIP's run time in basic mode was increased by a factor of 2.5 and its memory consumption by a factor of 4. We conclude that for the analysed datasets PureCLIP outperforms other methods when training its model on chromosomes 1 to 3. Moreover, if time and memory resources are not limited, it is advisable to train PureCLIP on a larger subset or on the entire dataset (default setting) to achieve optimal performance.

9.2.3. Robustness over a range of different bandwidth parameters

One parameter PureCLIP depends on is the bandwidth used to smooth the read start counts when estimating the fragment densities (see Section 5.3). The default bandwidth is 50 nt, which was also used in the previously shown evaluations. Importantly, the resulting fragment densities are not only used by the model to distinguish between *non-enriched* and *enriched* sites, but also to estimate the position-wise binomial n parameters which are then used to distinguish between *non-crosslink* and *crosslink* sites (see Section 5.3.2). Although it is possible to run PureCLIP with separate bandwidths for the two tasks, here we consider the default case of using the same bandwidth. Additionally it is worth noting that when input control data is incorporated, by default the same bandwidth is also used to compute the input fragment densities.

The optimal bandwidth is not obvious and depends on the characteristics of the CLIP data. In the following we will briefly discuss the expected effects of the most important

[1] Singleton read starts do not play a role in this context, since only positions with a fragment coverage $n_t \geq 10$ are used for the binomial probability parameter estimation (see Section 5.3.2).



(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

Figure 9.1.: Comparison of PureCLIP's performance when using only a subset (chromosomes 1 to 3) and when using the whole dataset for model training. The performance of CITS is shown for comparison.

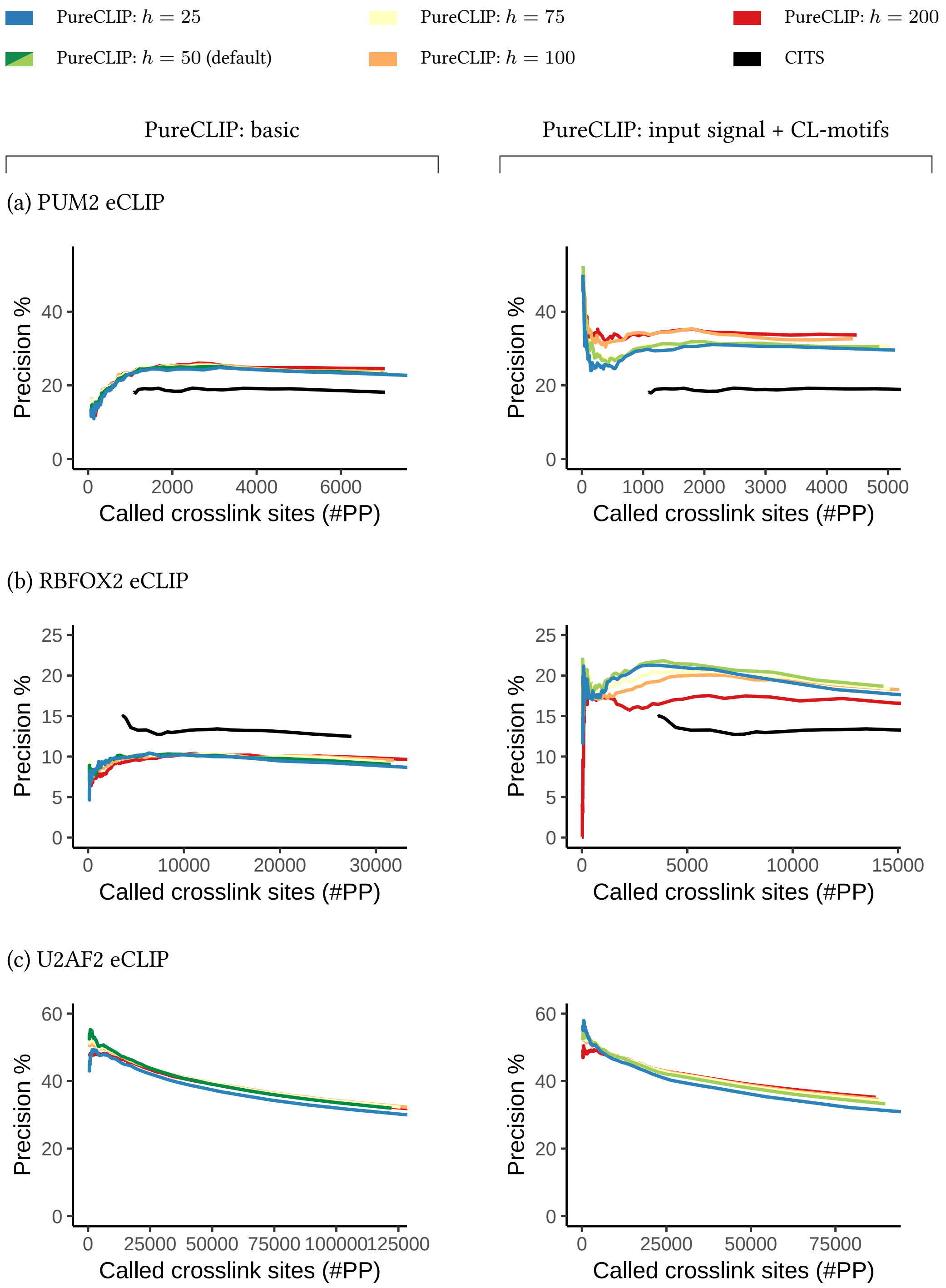


Figure 9.2.: Comparison of PureCLIP's performance when using different bandwidth parameters h to estimate the pulled-down fragment densities. The performance of CITS is shown for comparison.

ones on the different layers of the PureCLIP model, i.e. on the classification of sites as *non-enriched* or *enriched* and on the classification as *non-crosslink* or *crosslink*. First, to accurately distinguish between *non-enriched* and *enriched* sites, the bandwidth should ideally roughly reflect half of the width of the protein-RNA interaction footprints. The width of these footprints depends on the protein's binding characteristics, i.e. whether it interacts with rather short, defined regions, as PUM2 and RBFOX2 do, or whether it binds to longer regions, for example via multiple RNA binding domains. Moreover, such footprints comprise all read starts originating from cDNAs pulled-down via the target protein, including non-truncated and off-target truncated cDNAs. Consequently, the optimal bandwidth also depends on the cDNA length distribution, i.e. the longer the cDNAs the larger the optimal bandwidth.

As already mentioned, to distinguish between *non-crosslink* and *crosslink* sites the same bandwidth, or more precisely, the same fragment densities are used to estimate the binomial n parameters. Recall that the number of read start counts k_t at position t is then modelled in relation to n_t. As a consequence, the bandwidth affects how crosslinking patterns in the neighbourhood affect the position's emission probabilities. However, PureCLIP learns the protein-specific *non-crosslink* and *crosslink* binomial probability parameters p_0 and p_1, respectively, which counterbalances varying bandwidths for the most part.

Since these effects are difficult to assess for the user, we investigated PureCLIP's performance for a range of different bandwidths. The results shown in Figure 9.2 (left) demonstrate that PureCLIP's performance in its basic mode is highly robust for different bandwidths. When incorporating input signal (using the same bandwidth) and CL-motifs, the optimal bandwidths vary for different eCLIP datasets (see Figure 9.2, right). One possible reason for this is that the input signals – caused by a mixture of proteins – generate broader footprints compared to the target proteins and require larger bandwidths for some datasets. For other datasets a higher resolution might be beneficial, allowing a better distinction between target signals and nearby background signals. Nevertheless, PureCLIP reaches a higher precision robustly for all tested bandwidth parameters compared to CITS. It is also worth noting that the runtime and memory consumption of PureCLIP increases with increasing bandwidths (see Section 9.1). For this reason and since it produces a consistently good precision for all datasets, we chose a default bandwidth of 50 nt.

9.2.4. Performance of alternative scoring schemes

We further explored the scoring scheme used by PureCLIP to rank the called crosslink sites. Recall that alternative schemes were described in Section 5.3.6, where the default $score_{UC}$ is the log ratio of the state posterior probability of the most likely hidden state, i.e. *enriched + crosslink*, and of the second most likely hidden state. However, we do not know if the position-wise fragment density enrichment and the crosslinking information are equally important to capture target-specific interactions. We expect that this varies for the different protein-specific binding characteristics. For example, for a protein such as RBFOX2, which causes rather sharp truncation patterns at the two predominantly crosslinked sites within the bound sequence motif, the *crosslink* posterior probability might be more important than for proteins causing less sharp truncation patterns. With the alternatively implemented scoring schemes we additionally aim to score the confidence that a site is enriched ($score_E$), crosslinked ($score_{CL}$) or both equally weighted ($score_B$). The evaluation of these scores allows us to achieve a better understanding of the contribution of the individual signals. For the sake of completeness, we additionally compared the described scoring schemes against the raw posterior probability.

The results shown in Figure 9.3 (left) confirm that when running PureCLIP in basic mode, i.e. not correcting for any biases, the default $score_{UC}$ is among the best performing scores for all three eCLIP datasets. It reaches the same precision as $score_{CL}$ and the raw posterior probability. The score with the lowest precision is the enrichment focused $score_E$. These results clearly demonstrate that when not correcting for biases, the position's crosslinking signal is more important than its fragment density enrichment for all three proteins.

In contrast, when incorporating input signal and CL-motifs, for PUM2 eCLIP data $score_E$ reaches the highest precision, while $score_{CL}$ reaches by far the lowest precision (see Figure 9.3a, right). For the RBFOX2 and U2AF2 eCLIP data the different scores perform more similarly, but while for RBFOX2 again $score_E$ performs best, for U2AF2 $score_E$ performs worst (see Figure 9.3b,c, right). One possible reason for the high difference between the performance of $score_E$ and $score_{CL}$ on the PUM2 data is that the protein causes crosslinks at multiple sites within its binding region (see Figure 8.5,a, left) and consequently less sharp truncation patterns at individual sites. Taken together, the results indicate that $score_B$, which equally weights the confidence that a site is enriched and crosslinked, might be a good choice when correcting for biases and no prior knowledge is given about the protein's binding characteristics. However, since these results are highly protein dependent, this should be evaluated on a larger scale in the future.

9.3. Incorporation of RNA-seq data

Previously we described how we incorporate data from input control experiments to correct for non-specific background noise (see Section 5.4.1) and demonstrated how this improves PureCLIP's precision (see Section 8.3.2). Beside input data, it is also possible to include data from other types of control experiments or, to normalize for transcript abundances, from RNA-seq experiments. We preliminarily investigated if and how RNA-seq data can be included for the three analysed eCLIP datasets. In the following we will discuss the main challenges in this context, the obtained results as well as the potential for future improvements.

When normalizing for transcript abundances based on RNA-seq data, it is crucial to use data that reflects the abundances of the bound transcripts as accurately as possible. In this context, as proteins act differently in different subcellular compartments, the compartment specific transcript abundances play an important role (see

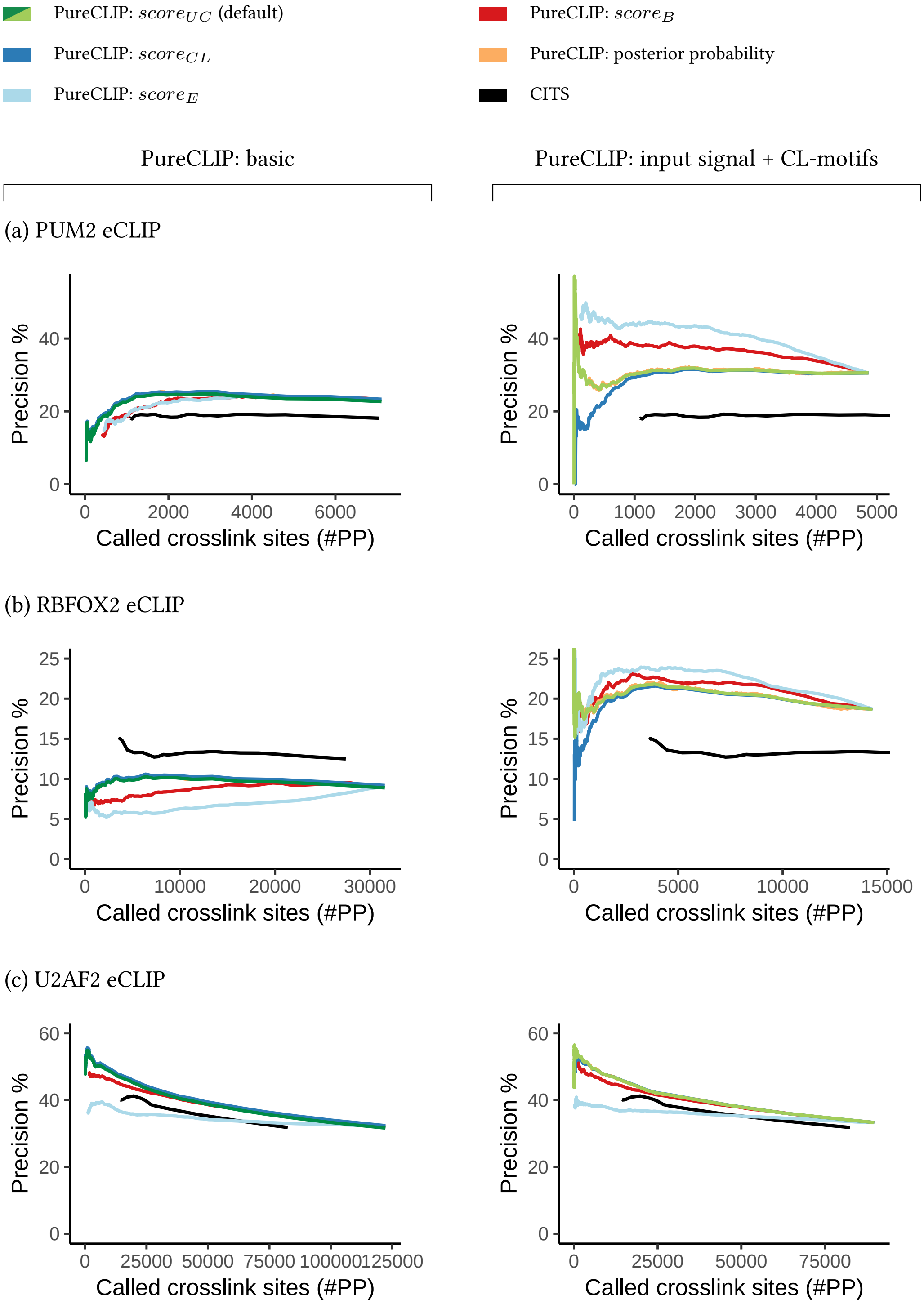


Figure 9.3.: Performance of PureCLIP for the four alternative scoring schemes. The orange line (raw posterior probability) is overlapped by the green line ($score_{UC}$). The performance of CITS is shown for comparison.

Section 3.2.2 and 3.4). In order to account for this, we used the K562 RNA-seq datasets of different subcellular compartments from ENCODE for comparison: whole cell (ENCSR885DVH, total), cytoplasmic fraction (ENCSR000COK, polyA), nucleoplasmic fraction (ENCSR000CQA, total) and chromatin fraction (ENCSR000CPY, total).

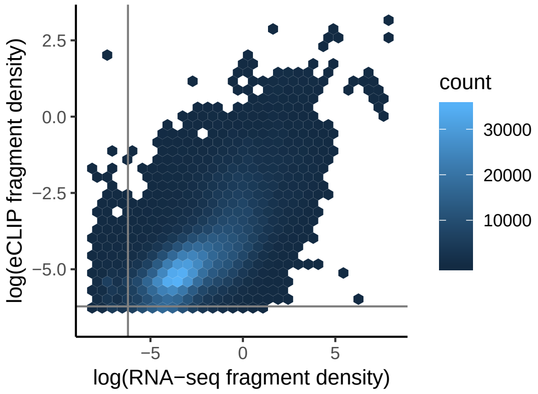
Another question to address is the resolution at which the RNA-seq signal should be included. Since PureCLIP uses position-wise fragment densities to detect *enriched* sites, it seems natural to use a corresponding signal from the given RNA-seq data. In comparison to gene- or transcript-wise signals, this also accounts for local changes in transcript abundances (see Section 3.3.3). Moreover, this avoids the problem of dealing with overlapping annotations. There remains the question which bandwidth to use to compute the RNA-seq fragment densities. A larger bandwidth might be more robust and able to reduce noise. On the other hand, signals might be more likely to be impaired by nearby intron-exon junctions, causing sharp jumps in read coverages. For the sake of simplicity, in the following we use the same bandwidth to compute target CLIP and RNA-seq fragment densities.

To explore which of the four compartment specific RNA-seq datasets is most suitable for normalization for each of the eCLIP datasets, we compared PureCLIP's performance when incorporating the signals using a bandwidth of 200 nt to compute the fragment densities. We found that for PUM2 and RBFOX2, whole cell RNA-seq data improves the precision of the called crosslink sites the most, while for U2AF2 the nucleoplasmic RNA-seq data does (see Figure A.13). These results are in agreement with the proteins known predominant subcellular locations [14]. As an example, Figure 9.4a shows the correlation between PUM2 eCLIP and whole cell RNA-seq fragment densities. Figure 9.4b shows PureCLIP's classification into *non-enriched* and *enriched* sites, normalized for the RNA-seq fragment densities.

For each eCLIP dataset we included the most suitable RNA-seq dataset and compared the obtained precision for different bandwidths (see Figure A.14). Of the compared bandwidths, 200 nt achieves the largest improvements. The results presented in Figure 9.5 show that the incorporation of the RNA-seq signals generally leads to only moderate improvements of the overall precision over PureCLIP in basic mode. The overall precision is comparable to the precision that is achieved when including input signal, however, the ranking of the sites is notably worse. Note that in combination with the incorporation of CL-motifs for PUM2 and RBFOX2 eCLIP data, the benefit of using RNA-seq data for normalization almost vanishes (see Figure A.15). The most likely reason for this is that the incorporation of RNA-seq data mainly prevents false positives with relatively weak interaction footprints located within CL-motifs in highly abundant RNAs, which can already be avoided when correcting for the crosslinking sequence bias. Furthermore, for RBFOX2 larger bandwidths result in lower precisions, as already observed when incorporating input signal and CL-motifs (see Figure 9.2b, right).

We conclude that PureCLIP is able to incorporate RNA-seq signals to normalize for transcript abundances, although the increase in precision is moderate and the biggest impact occurs when not simultaneously correcting for the sequence bias. For most cases, input control experiments – which indirectly control for background

(a)



(b)

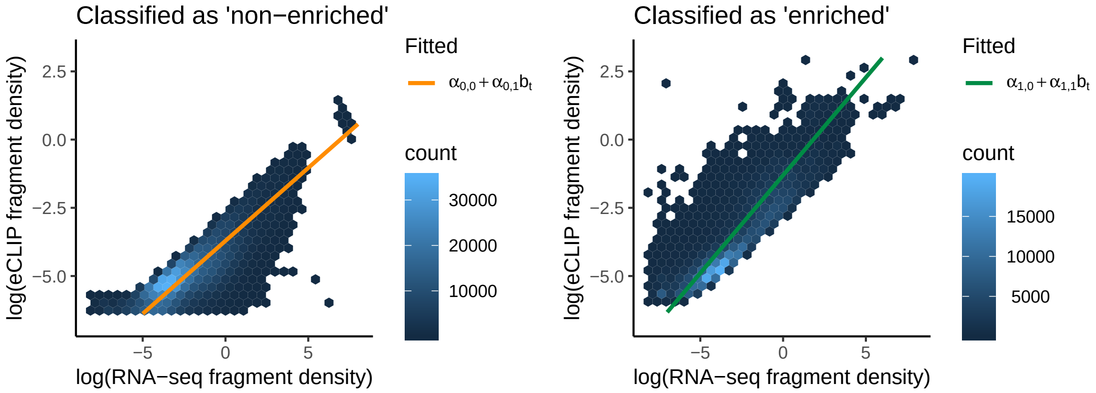


Figure 9.4.: a) Correlation between PUM2 eCLIP and whole cell RNA-seq fragment densities, computed using a KDE bandwidth of 200 nt and plotted at all sites with at least one read start. The grey lines represent the fragment density thresholds, i.e. the fragment density corresponding to a singleton read start, used when learning the HMM parameters (see Section 5.4.1). **b)** The orange and green lines represent the mean parameters $\mu_{0,t}, \mu_{1,t}$ predicted by PureCLIP for the *non-enriched* and *enriched* emission probability distributions, respectively, based on the position-wise RNA-seq fragment density b_t. Additionally, the resulting classification into *non-enriched* and *enriched* sites is shown.

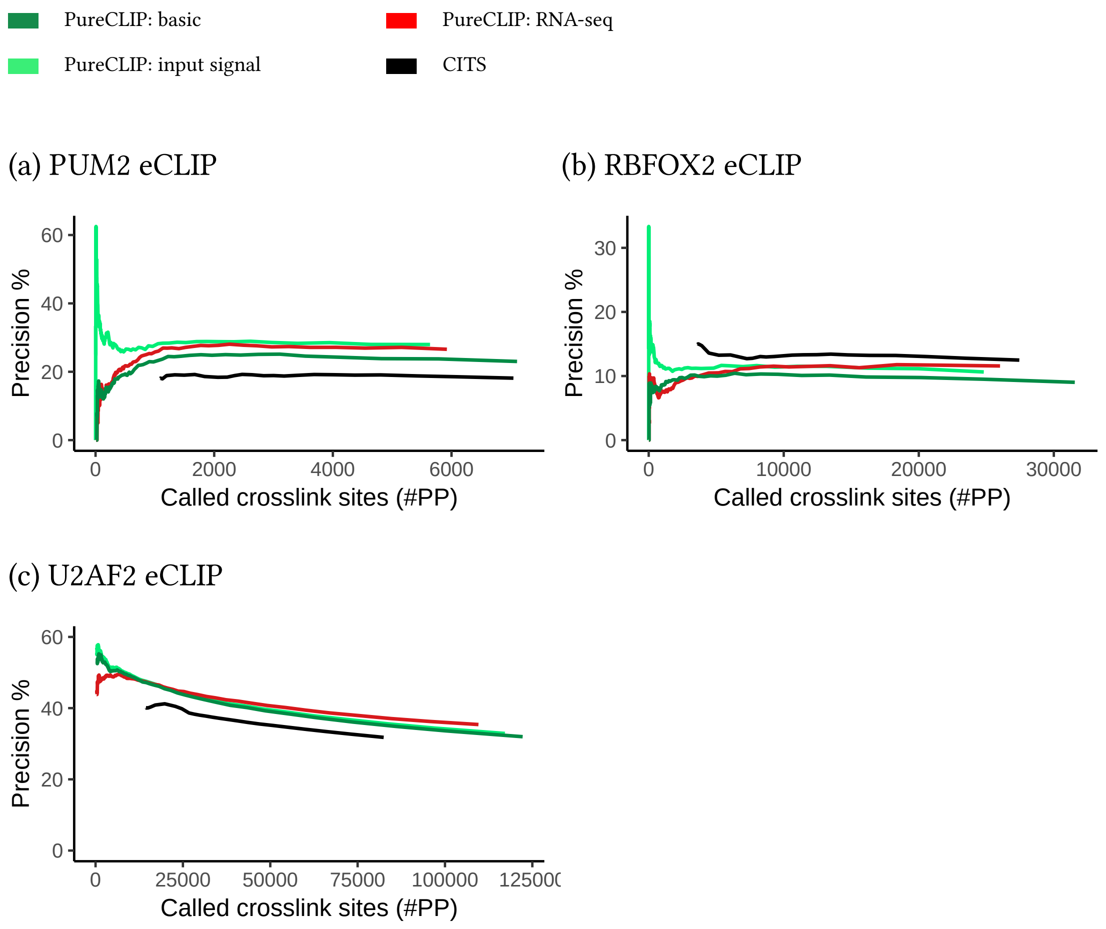


Figure 9.5.: PureCLIP's precision when including RNA-seq data from the best suitable available cellular compartment (bandwidth $h = 200$) in comparison to PureCLIP in basic mode and when including input control data (bandwidth $h = 50$). **a)** and **b)** Incorporation of whole cell RNA-seq data. **c)** Incorporation of nucleoplasm RNA-seq data.

binding, crosslinking preferences and transcript abundances – are more suitable for bias correction. However, in particular for CLIP experiments performed for subcellular compartments, such as Fr-iCLIP [17] experiments, and for which matching RNA-seq but no input control experiments exist, the ability to incorporate this data can be useful.

9.4. Performance on PTBP1 iCLIP data

We furthermore investigated PureCLIP's performance on PTBP1 iCLIP data, based on its known binding upstream of 3' splice sites of silenced exons, similar to the study published by Chakrabarti et al. [24]. For this we used a list of regulated exons obtained from an RNA-seq analysis of CRISPR PTBP1 knockout cells provided by the Ule lab [24]. PTBP1 has four RNA-binding domains, binds to CU-rich regions and is known to build clusters of crosslink sites, which are on average 29 nt long [61, 63]. Moreover, it is known that PTBP1 can bind in clusters of multiple proteins, causing the formation of higher-order complexes [28]. As a consequence, we expect read starts originating from truncated cDNAs being more distributed within such clusters and thus to form less sharp truncation patterns compared to, for example, PUM2 or RBFOX2 which bind to short defined sequence motifs. However, it is noteworthy that this evaluation is less significant compared to the evaluations described for PUM2, RBFOX2 and U2AF2 (see Section 8.3.2) due to the relatively low number of silenced exons (776) available for the analysis. Furthermore, the results should be interpreted with care, since for some silenced exons PTBP1 might also bind within or downstream of the exons [63]. Nevertheless, it provides an important insight into the performance of PureCLIP for data with less strong truncation patterns compared to other existing methods.

We used the top 17,629 ranking sites of all methods, defined based on the lowest number of called crosslink sites obtained with CITS to allow a fair comparison, and investigated the distribution of calls around the bona fide binding region. Although the exact binding site of PTBP1 is unknown, as expected we observed an accumulation of crosslink sites and binding regions upstream of 3' splice sites of silenced exons (see Figure 9.6) for most methods. Interestingly, the highest density was reported by PureCLIP, both for called crosslink sites and binding regions, 37 and 34 nt upstream of 3' splice sites of silenced exons, respectively.

Because iCount calls much more crosslink sites than other methods and was used in previous studies on PTBP1 iCLIP data [24] with its default setting, we additionally investigated weather its performance would be notably different in a higher #PP range. Therefore we performed an equivalent analysis as described above for the top 100,000 sites (see Appendix A.11). For this we used a PureCLIP setting optimized for longer crosslink clusters (see Section A.12), which increases its sensitivity for individual sites within longer clusters and enables the comparison in a higher #PP range. We observed that here iCount calls the highest density of crosslink sites $\sim$ 35 nt upstream of 3' splice sites of silenced exons, while PureCLIP still reaches a higher density of binding regions within this region (see Figure A.16).

In summary, PureCLIP performs well also in detecting bona fide protein-RNA inter-

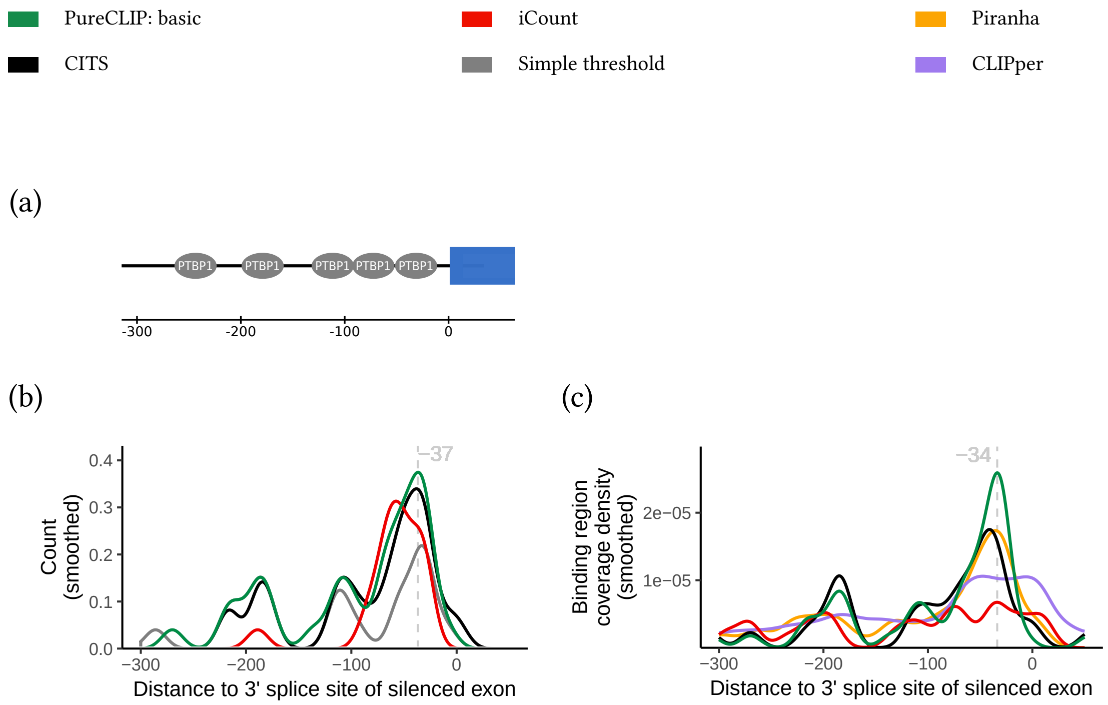


Figure 9.6.: (a) PTBP1 binding upstream of silenced exons as schematically depicted by Haberman [60]. (b) Distribution of the top 17,629 crosslink sites and (c) binding regions called by each method for PTBP1 iCLIP data around 3’ splice sites of silenced exons. The distributions were smoothed with a Gaussian kernel density estimate using a bandwidth of 10 nt to reduce the noise caused by the relatively low counts and densities. The dashed gray lines denote the position with the highest signal.

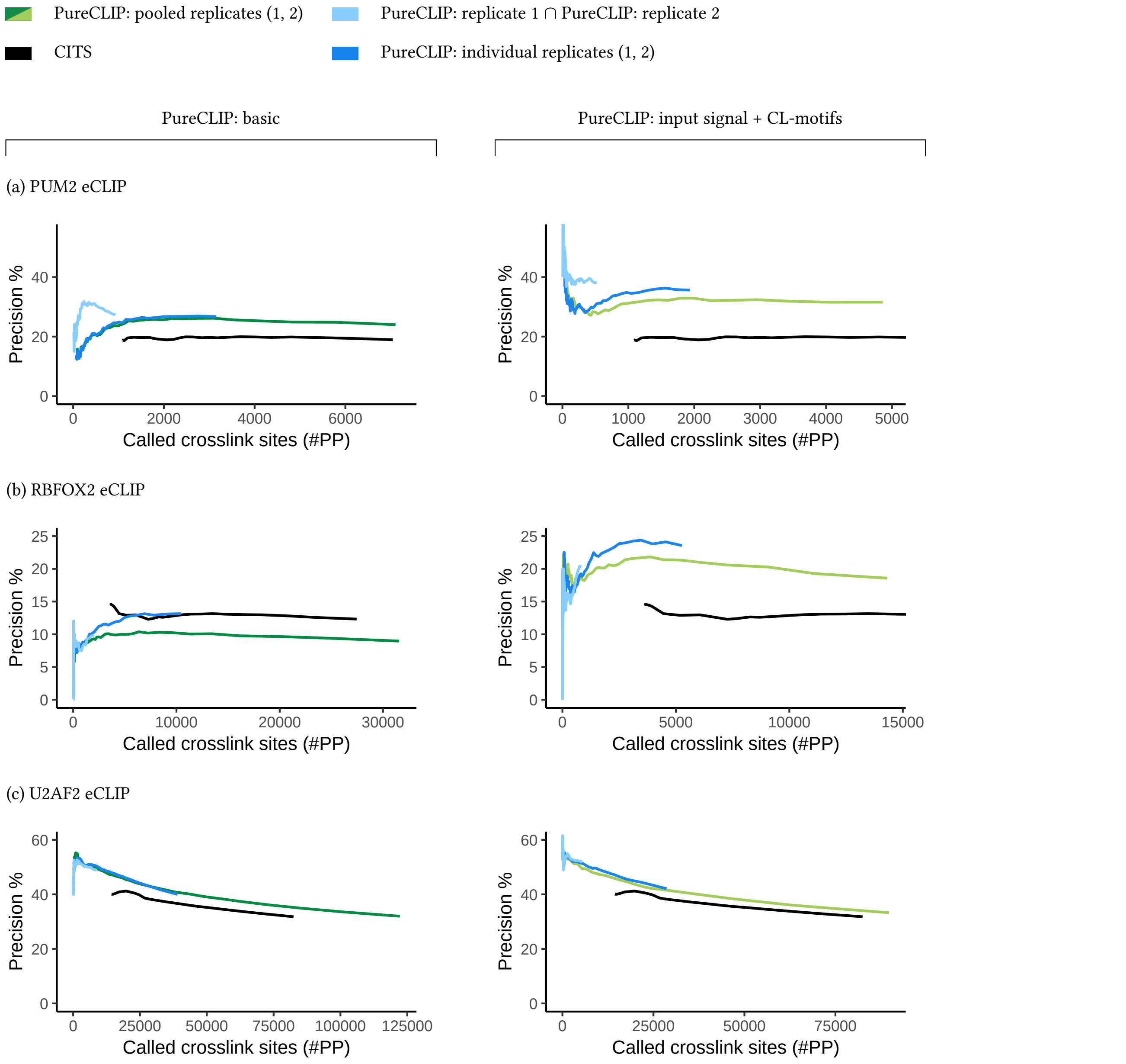
actions for proteins generating larger crosslink clusters, such as PTBP1, both when using its default setting as well as when using a setting optimized for larger crosslink clusters.

9.5. Incorporation of individual CLIP replicates

As described in Section 5.5, we extended PureCLIP to incorporate individual replicates. To recapitulate, for this purpose PureCLIP uses joint emission probabilities, while learning the emission probability parameters separately for each replicate.

To investigate whether this strategy achieves a gain in performance, we compared it 1) to PureCLIP run on pooled replicates and 2) to the intersection of crosslink sites obtained by running PureCLIP on each replicate separately. The results shown in Figure 9.7 demonstrate that PureCLIP reaches a higher precision when including replicates over a broad range of #PPs for PUM2 and RBFOX2 eCLIP datasets in comparison to using pooled data. For U2AF2 eCLIP data, for which the precision is already relatively high using pooled data, the precision does not change notably. Interestingly, for PUM2 the precision is only increased when simultaneously incorporating input signal and CL-motifs. One possible reason for this is that biases are often also reproducible between replicates, counteracting the gain obtained from reproducible target signals. Thus, to ensure target specificity, it is also important to correct for biases when including replicate information. As expected, the sensitivity, reflected by the number of obtained PPs, is reduced, i.e. for all datasets the number of called crosslink sites is less than half of the number called on pooled data. In comparison, for PUM2 eCLIP data the intersection strategy reaches an even higher precision over all #PPs, while reaching a similar precision for RBFOX2 and U2AF2 data. However, when including individual replicates PureCLIP is clearly more sensitive and calls up to five times as many crosslink sites. Moreover, due to a lower precision of the top ranking sites, for RBFOX2 eCLIP data the overall precision when including individual replicates is also higher than for the intersected sites.

We conclude that including individual replicates further improves PureCLIP's precision for some datasets in comparison to using pooled replicate data, while enabling a much higher sensitivity in comparison to the naive intersection approach.



(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

Figure 9.7.: Performance of PureCLIP when including individual replicates into the model compared to running PureCLIP on pooled replicates and to using the intersection of called crosslink sites obtained from running PureCLIP on each replicate separately. All datasets consist of two replicates.

Part III.

Discussion and Appendices

10. Discussion and Conclusion

In this last chapter we will first discuss the presented work and gained insights, then give an outlook on possible applications and future extensions, and finally conclude this thesis with some general remarks.

10.1. Discussion

The detection of target-specific protein-RNA interaction sites from single-nucleotide resolution CLIP-seq data remains a challenge. Previous methods for the analysis of such data typically report a large fraction of false positives, as they are sensitive to different sources of biases. Peak callers such as Piranha, which call regions enriched in read coverage without explicitly modelling read start counts as an indicator for truncation sites, are prone to capture high background signals. On the other hand, permutation based crosslink site detection methods such as CITS and iCount call sites with a significant fraction of read starts, but cannot distinguish whether such sites are caused by target-specific crosslinks or by non-specific crosslinks or artefacts within highly abundant regions.

To overcome these limitations, we presented a new statistical approach called PureCLIP. It calls crosslink sites considering both regions enriched in protein-bound fragments and the specifics of iCLIP/eCLIP truncation patterns. It also explicitly models possible sources of errors, such as non-specific background binding and the crosslinking sequence bias in order to reduce the number of false positives. This number can then be further reduced with our method by incorporating several individual replicates at once. We have demonstrated in a comprehensive evaluation, based on simulated and real data, that PureCLIP reaches a higher precision in calling both individual crosslink sites and binding regions compared to previous methods. For most datasets and settings, this already holds for PureCLIP in basic mode, i.e. without incorporating any covariates.

In the following, we will first discuss our model design choices and their practical implications retrospectively in comparison to other methods, and then a few general issues concerning truncation-based CLIP signals.

10.1.1. Model design

Footprint modeling

The strength of PureCLIP is that it does not simply detect the highest peaks or pile-ups of read starts but accurately models the two signals, the position-wise pulled-down fragment density, and the read start count in relation to this fragment density. In this way, read start counts are indirectly normalized for local changes in transcript abundances. As a consequence, for the analysed eCLIP datasets PureCLIP calls far fewer crosslink sites within regions of high read coverage and within known common background binding regions [119] than CITS and the simple threshold method. iCount calls sites in even less covered regions, likely because it performs a transcript-wise normalization, however, it also reaches consistently lower precisions than PureCLIP across all evaluations.

In comparison to general peak-calling methods, crosslink site detection methods are more likely to be affected by the CL-bias. This particularly applies to PureCLIP, as the modeling of the read start counts aims to detect footprints also at low-affinity binding sites or within lowly abundant RNAs. Nevertheless, this unique feature allows for the distinction between target-specific interactions and non-specific crosslinking patterns within highly abundant RNAs. For datasets containing a high fraction of non-specific truncation patterns, caused for example by CL-motifs within proximity of less crosslinking affine target binding regions, this bias can be corrected.

The accurate modeling of the signals allows for the correction for different biases at the two layers, i.e. broader background binding footprints and increased individual read start counts caused by the CL-bias. We demonstrated that the incorporation of input signals as well as the incorporation of CL-motif scores greatly improves the precision of our method, which for the latter strictly outperforms all other methods over all analysed datasets. Both covariates can also be incorporated simultaneously, which increases PureCLIP's precision even further.

Two-component mixture models: tradeoff between precision and sensitivity

Different CLIP analysis methods achieve different precision-sensitivity ranges: while PureCLIP provides high precision, iCount provides a relatively high sensitivity even for the lowest applicable FDR threshold. The reason for this are different underlying model assumptions. iCount models a background distribution using a permutation test, so that the classification into crosslink sites depends on a user-defined FDR threshold. For some datasets, even when using only sites with a reported FDR of 0, it still reports more sites than PureCLIP altogether. Beside numerical imprecision, the reason for this is most likely that varying read start counts caused by artefacts are not considered, and are also highly unlikely under the null hypothesis. In contrast, PureCLIP explicitly models the background (*non-enriched*, *non-crosslink*) and the target (*enriched*, *crosslink*) signals and will thus always classify a certain fraction of sites as background.

In practice, different objectives exist for the analysis of CLIP data. For example, for studies investigating the global positional distribution of binding signals around regulated splice sites [24, 154], a higher sensitivity is often beneficial, in particular when the number of regulated sites is low. Therefore, often raw or normalized read densities are used for the analysis. Accordingly, in such cases, methods such as iCount would likely be more suitable than PureCLIP. On the other hand, when the objective is to precisely detect interactions for individual transcripts or to reduce the impact of systematic biases, methods that are able to reliably control the number of false positives, such as PureCLIP, are essential.

Hidden Markov model

A main design decision was to use an HMM framework based on a first-order Markov chain to account for the spatial dependencies between neighbouring positions. In practice, we observed that the learned transition probabilities between the four hidden states reflect protein-specific binding and crosslinking characteristics. For example, for PUM2, which can cause crosslinks at all bases within its preferentially bound sequence motif, the probability of staying within the *enriched + crosslink* state is notably higher than for RBFOX2, for which the predominant crosslink sites are typically 4 nt apart. The latter raises the question in how far higher-order Markov chains would be more suitable for this task; however, they would also increase the computational complexity considerably. Concerning the *non-enriched* and *enriched* states, the emitted position-wise fragment densities already contain information about neighbouring positions. Although this violates the assumption that the observations are independent given the state, in practice this yields good results, as this causes the generation of larger *non-enriched* and *enriched* segments with few transitions, while the state posterior probabilities mainly depend on the corresponding fragment density emission probabilities.

From crosslink sites to binding regions

Although PureCLIP's main objective is to detect individual target-specific crosslink sites, it is sometimes desirable to identify binding regions for the investigated protein.

Instead of defining binding regions at a broader peak level, PureCLIP merges crosslink sites to binding regions based on their genomic distance, similar to CITS and iCount. Further investigation is needed to explore which distance is optimal for which proteins, or to address this task in a more systematic manner, for example by segmenting enriched regions based on *crosslink* posterior probabilities. Nevertheless, we were able to show that on simulated and experimental CLIP data PureCLIP recovers the known binding regions with higher precision compared to the other methods designed for peak calling or crosslink cluster detection. For PUM2, RBFOX2 and U2AF2, we know that they predominantly bind short, defined regions of 6 to 9 nt. Piranha, CLIPper and iCount report regions that include large flanking regions. CITS mainly calls individual sites distributed across the transcriptome, but fails to recover regions. In contrast, PureCLIP detects relatively short regions, often only a few nucleotides long.

We further conclude that for proteins binding the target RNA with multiple RBDs simultaneously or for proteins binding in clusters of multiple instances, PureCLIP is a promising method for recovering the individual bound regions.

10.1.2. Protein-binding characteristics are reflected by signal contribution

PureCLIP classifies sites as *enriched + crosslink* based on the position-wise fragment densities and read start counts. We have shown that in order to rank the identified crosslink sites, different scoring schemes perform best for different proteins. While for PUM2 a score focusing on the confidence that a site is *enriched* achieves the highest precision, for U2AF2 a score focusing on the confidence that a site is *crosslinked* performs best. These results demonstrate how for proteins with different binding and crosslinking characteristics the relative importance of the two used signals varies. As a tradeoff, we suggested a score to equally weight the confidence that a site is *enriched* and *crosslinked*, which performs reasonably well for the investigated proteins; this, however, remains to be explored on a larger scale.

Independently of the scoring scheme, PureCLIP assumes cDNA truncations for calling crosslink sites. However, certain proteins generate weak truncation patterns, for example, because they slide along the RNA with multiple RBDs instead of binding to a short, defined region. On such data, PureCLIP and other crosslink site detection and peak-calling methods will likely perform less optimally. A strength of PureCLIP is that it learns the protein-specific read start rates, both for *non-crosslink* and *crosslink* states, which can counterbalance weaker crosslinking signals to a certain degree. For PTBP1, which generates longer clusters of crosslink sites, we showed that PureCLIP captures its known binding regions upstream of regulated splice sites comparably well or better than other methods. Whether PureCLIP can also be applied to proteins that bind rather diffusely across entire transcripts such as MATR3 [24] remains to be tested.

10.1.3. Ranking of reported protein-RNA interactions

Ideally, given a method's ranking of called crosslink sites or binding regions, when increasing the score threshold and thus decreasing the number of predicted positives (#PP), we would expect increasing precision. However, in particular for the top ranking sites, the opposite is often the case, indicating highly scoring artefacts. Moreover, even for higher #PP ranges, we observed a plateau in the precision-PP curves for most of the compared methods, where the precision stays roughly constant for an increasing #PP. We observed this for the evaluation both based on known predominant binding regions and based on the bias-corrected replicate agreement. From a user's perspective, this is counter-intuitive, since one would expect more precise results when increasing the applied score threshold.

These effects are clearly reduced for crosslink sites and binding regions called by PureCLIP when correcting for biases. For the PUM2 eCLIP data, a remaining plateau can be further reduced by using the enrichment focused posterior-ratio score. Although in general we need to keep in mind that with our evaluation strategy we can only estimate the precision, the results demonstrate that likely false positives frequently occur across the entire ranking for most methods, which underlines the need to precisely model target-specific interaction footprints.

10.1.4. Recent developments

Recently, a new tool that can be used for the analysis of truncation-based CLIP data called omniCLIP [41] was published. As the name suggests, it is designed to handle different types of CLIP data. It learns the important diagnostic events from the data and uses them to support peak calling based on the position-wise read coverage. OmniCLIP models the data across replicates, while, similarly to PureCLIP, using a non-homogeneous HMM to include covariates to correct for non-specific background signals. Despite featuring similar model characteristics as PureCLIP, it has a somewhat different scope as it is designed to call broader peaks and not individual crosslink sites. Furthermore, as it is not explicitly designed for truncation-based CLIP data, it does not account for the effect of truncations on the read coverage.

10.2. Outlook

In the following, we will first describe possible future improvements of our method, and then list applications which would be interesting to explore and which would likely benefit from high-resolution binding regions.

10.2.1. Future improvements

Currently, PureCLIP allows the incorporation of covariates which influence the emission probabilities of the pulled-down fragment density or of the read start counts. Besides information from control experiments or CL-motifs, information about common background binding regions or mappability would be interesting candidates for investigation as covariates. However, it is uncertain to what degree this information is contained in input data and thus already successfully integrated. A step further would be the modelling of non-homogeneous transition probabilities between states, for example, to incorporate information about the sequence or structure binding preferences of the target protein. Furthermore, PureCLIP could be adapted to simultaneously model other types of diagnostic events to improve the analysis for CLIP protocols such as PAR-iCLIP [68] which additionally cause a higher fraction of base substitutions at crosslink sites.

Another topic worth investigating would be the parameter settings for different protein-binding characteristics, for example the optimal bandwidth and scoring scheme. Ideally, these parameters would then be derived directly from the data.

10.2.2. Applications

Binding site prediction

The number of recoverable binding sites is generally limited in CLIP experiments due to low transcript abundances or poor mappabilities within repetitive regions. However, sometimes the aim is to recover all potential binding sites of a certain protein of interest. To address this problem, methods such as GraphProt [99] and Pysster [18] were developed which learn the sequence and structure binding preferences of proteins from CLIP peaks, and then predict binding sites in silico. Although such predictions cannot account for in vivo regulations, for example by cooperative binding with other proteins, they provide a valuable supplement to experimental CLIP data. For this task, the use of high-confidence, high-resolution binding regions called by PureCLIP might improve future predictions.

Protein interactions with long non-coding RNAs

The functional roles of the diverse lncRNAs are still poorly understood. Besides their structure, expression, conservations, and interactions with other RNAs or DNAs, understanding their interactions with RBPs, both in facilitating other regulatory processes and the regulation of the lncRNAs itself, is the key to the understanding of their overall function [101]. However, the analysis of lncRNA interaction landscapes is challenging, as they are often less expressed than mRNAs [101] and thus tend to generate lower peaks in CLIP data. Consequently, when applying peak-calling methods, binding sites within such lowly abundant lncRNAs are more likely to be missed. On the other hand, in case of highly abundant RNAs, such as MALAT1 and XIST, which contain a high fraction of background signal, other methods often tend to call a large number of false positive crosslink sites or large regions. Nevertheless, it is desirable to accurately capture target-specific signals within such regions [143]. PureCLIP is designed to handle both described scenarios, and thus has the potential to improve the analysis for lncRNAs, allowing for a better functional characterization [101].

Genetic variants and protein-RNA interactions

Another interesting application would be to investigate the effect of genetic variants on protein-RNA binding. This could be interesting in the context of determining the deleteriousness of genetic variants [32] or when known disease-causing variants are given, and the goal is to understand the underlying mechanism. For each variant, one could test if it is located within a protein's binding region, for example by making use of the available eCLIP datasets for 150 RBPs, which might be disrupted. Moreover, if available or possible to generate, disease specific CLIP experiments for the candidate proteins together with RNA-seq experiments could then be further used to infer the effect on the disease-causing pathway.

For many proteins, the CLIP signals overlap due to cooperative or competitive binding at the same transcript. Long, imprecise binding regions would further impair the results, and thus methods providing high-resolution binding regions are crucial.

Allele-specific protein-RNA interactions

Recently, advances were made towards the detection of allele-specific binding from the available ENCODE eCLIP datasets. Two methods were recently published [9, 152], both based on the idea of using the ratio of allele-specific eCLIP reads to detect allele-specific binding. Both methods are based on binding regions, called with the peak-calling methods CLIPper and Piranha, respectively. PureCLIP may be a promising alternative here, not only because of its superior precision, but also because allele-specific binding events by definition cause lower peaks, and might thus be missed by other methods.

10.2.3. Differential binding analysis

Another interesting and challenging problem is the detection of differential protein-RNA binding under two or more conditions. This is crucial, for example, in order to understand the regulatory differences between healthy and diseased cells. As already mentioned, such differences can be caused by mutations interrupting binding sites, by mutations changing the binding affinities of the protein, or by differences originating from other interaction partners.

Few methods for the detection of differential protein-RNA binding have been published so far. dCLIP [147] uses a hidden Markov model on bin-wise log-fold read count enrichments between two conditions. In contrast, PEAKachu [67] does not rely on fixed bins, but instead determines peak boundaries and then uses DESeq2 [89] to detect significant count enrichments while modeling the variance between replicates and conditions. To our knowledge, no currently available method explicitly corrects for biases; however, it has been shown that, for example, differences in transcript abundances between conditions also cause differential CLIP signals [109]. Furthermore, differences in background binding might cause non-specific differential signals, which need to be distinguished from target-specific differential binding. Therefore, a modeling framework such as the one used in PureCLIP may be able to improve the precision of the detection.

A potential approach could be based on the comparison of the position-wise posterior probabilities (*enriched*, *crosslink*) computed for each condition separately. In this way, one could detect differentially bound regions and differentially crosslinked sites simultaneously, and weight these signals depending on the investigated protein. In order to additionally make use of replicate information, such an approach could be further improved by modeling the variance between replicates and across binding regions, for example by using the limma package [120, 131].

10.3. Conclusion

In this thesis we presented a new statistical model for the analysis of truncation-based CLIP data that allows for the detection of target-specific crosslink sites and binding regions more precisely than other state-of-the-art methods. PureCLIP captures protein-RNA interaction footprints, while not relying on the highest peaks alone, and being able to correct for biases, such as transcript abundances, background binding and crosslinking sequence preferences. Therefore, it provides a promising method for the analyses of the growing number of truncation-based CLIP datasets, also for proteins with lower binding affinities or proteins binding to lowly abundant RNAs, such as lncRNAs.

PureCLIP was the first method that explicitly corrects for the crosslinking sequence bias and, to our knowledge, the first that is able to include replicates for the detection of individual crosslink sites. Furthermore, it robustly reaches a high precision for a range of different parameter settings, and can be applied to proteins with different binding characteristics. It can be easily installed via Bioconda [59] and, thanks to the Freiburg Galaxy Team, accessed via the European Galaxy server [1].

The presented computational strategies to increase the numerical stability of PureCLIP (see Section 5.6.2) are an effort to allow for its application on a larger scale, including datasets with different characteristics. This could for example be different binding characteristics as well as occurrences of multi-mapping reads causing huge pile-ups of read starts. Given the specifics of the PureCLIP model, besides iCLIP and eCLIP data, it can be used to analyse data from other types of truncation-based CLIP protocols, such as irCLIP [156], FLASH [4] and miCLIP (methylation-iCLIP), a customized version of iCLIP for capturing m5C methylated sites on RNAs with single-nucleotide resolution [69]. At the time of writing this thesis, PureCLIP is already being used by multiple labs for the analysis of in-house CLIP data, and to re-analyse the 223 published ENCODE eCLIP data datasets [109].

A. Appendices

A.1. Baum-Welch algorithm

The following rearrangements are used to obtain the individual update functions from the expected log-likelihood $Q(\theta|\theta')$:

$$
\begin{aligned}
Q(\theta \mid \theta') = & \sum_{s \in \{1,\dots,l\}^T} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \left(\log \pi_{s_1} + \sum_{t=2}^{T} \log a_{s_{t-1}s_t} + \sum_{t=1}^{T} \log e_{s_t}(y_t) \right) \\
= & \sum_{s \in \{1,\dots,\ell\}^T} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \log \pi_{s_1} \\
& + \sum_{s \in \{1,\dots,\ell\}^T} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \sum_{t=2}^{T} \log a_{s_{t-1}s_t} \\
& + \sum_{s \in \{1,\dots,\ell\}^T} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \sum_{t=1}^{T} \log e_{s_t}(y_t) \\
= & \sum_{t=1}^{T} \sum_{j=1}^{\ell} \sum_{s : s_1 = j} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \log \pi_j \\
& + \sum_{t=2}^{T} \sum_{m=1}^{\ell} \sum_{n=1}^{1} \sum_{s : s_{t-1} = m, s_t = n} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \log a_{mn} \\
& + \sum_{t=1}^{T} \sum_{j=1}^{\ell} \sum_{s : s_t = j} P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{Y} = \boldsymbol{y}; \theta') \log e_j(y_t) \\
= & \sum_{j=1}^{\ell} P(S_1 = j \mid \boldsymbol{Y} = \boldsymbol{y}, \theta') \log \pi_j \\
& + \sum_{t=2}^{T} \sum_{m=1}^{\ell} \sum_{n=1}^{\ell} P(S_{t-1} = m, S_t = n \mid \boldsymbol{Y} = \boldsymbol{y}, \theta') \log a_{mn} \\
& + \sum_{t=1}^{T} \sum_{j=1}^{\ell} P(S_t = j \mid \boldsymbol{Y} = \boldsymbol{y}, \theta') \log e_j(y_t)
\end{aligned}
\tag{A.1}
$$

A.2. Model choice: gamma distribution

The gamma distribution is a popular and flexible choice to model non-negative continuous values with a right skewed distribution and thus we use it for the fragment density values. We investigated the properties of the observed fragment densities with skewness-kurtosis plots (originally proposed by Cullen and Frey [35], implemented in the *fitdistrplus* R package [38] which was used here). For this we used all sites which are used by PureCLIP for fitting the *non-enriched* and *enriched* emission probability distributions: sites with at least one read starting while discarding singleton reads. Although we do not know the empirical distributions corresponding to the assumed *non-enriched* and *enriched* components of our model, the skewness-kurtosis plots indicate that the total fragment densities within real data can be best modeled with the properties of a gamma distribution, in comparison for example to a lognormal distribution (see Figure A.1a). Furthermore, when dividing the observed values for different ranges of associated input fragment densities, the gamma distribution remains the best fitting distribution with respect to the skewness-kurtosis properties (see Figure A.1b). This shows that also for the non-homogeneous HMM the gamma GLM is likely a well suited model.

A.3. Preprocessing of iCLIP and eCLIP data

The following tools and settings were used for data preprocessing:

- **Adapter removal and filtering**

  Possible adapter contaminations at 3' ends were removed using TrimGalore [80] (v0.4.2, based on cutadapt) for iCLIP data, and by running cutadapt [98] (v1.12) twice for eCLIP data. The later was done in order to correct for possible double ligation events [143]. Reads shorter than 18bp were discarded.

- **Read mapping**

  Reads were mapped against the human genome (hg19) using STAR [39] (v2.5.1b) with setting '--alignEndsType EndToEnd', '--scoreDelOpen -1' for gap penalty, and '--outFilterMultimapNmax 1' to discard reads mapping to multiple locations. Ensembl Release 75 annotations were included to account for splice junctions.

- **PCR duplicate removal**

  To remove PCR duplicates we used UMI-tools [130] (v0.4.3) with its default setting for iCLIP data and with setting additionally '--paired' for eCLIP data.

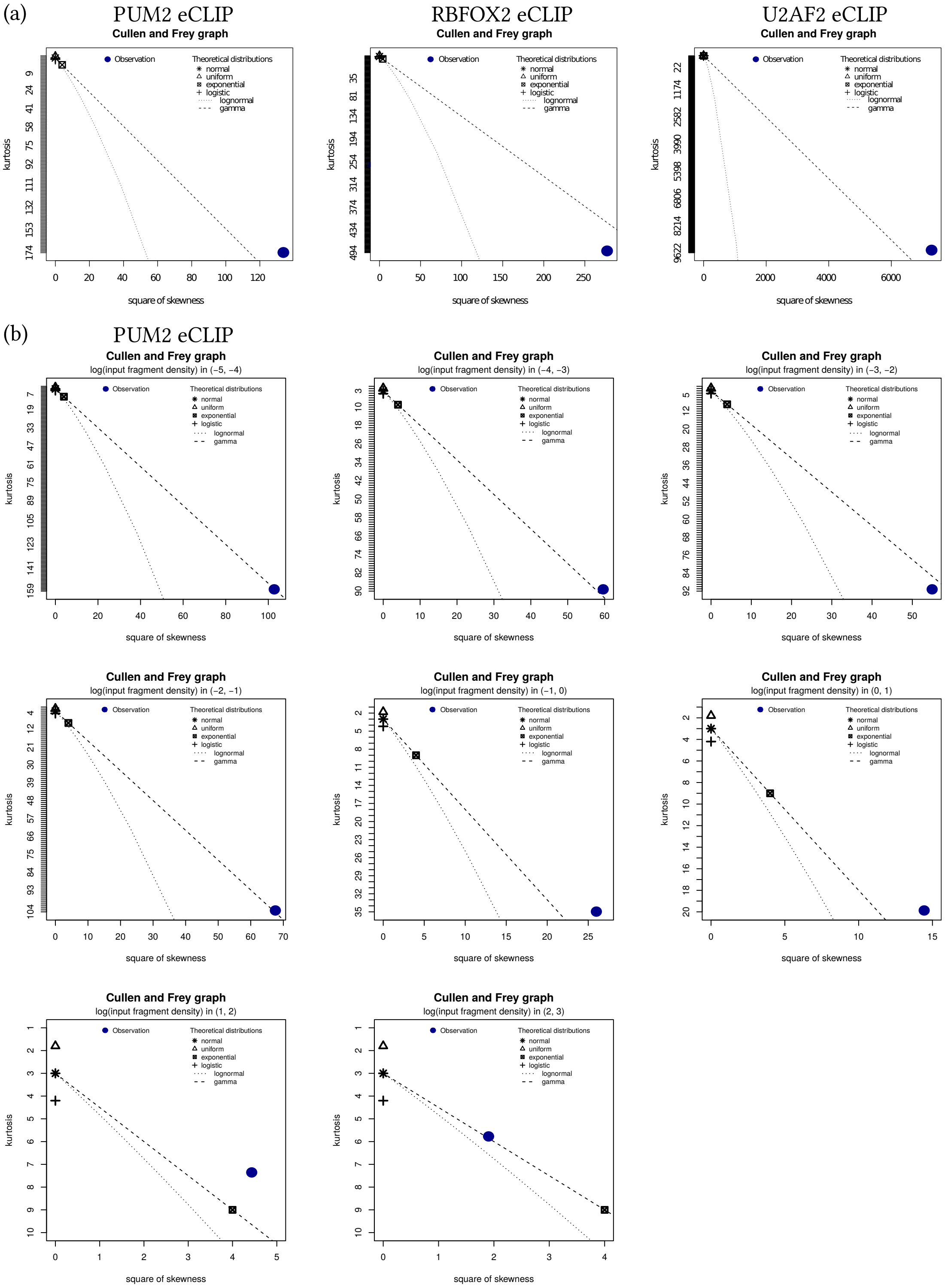


Figure A.1.: Skewness-kurtosis plots [38] for the pulled-down fragment densities from all sites that are used by PureCLIP to fit the *non-enriched* and *enriched* emission probability distributions, **a)** for different eCLIP datasets and **b)** for PUM2 eCLIP data divided for different ranges of input fragment densities.

A.4. Extension of the Nelder-Mead algorithm

We extended the Nelder-Mead algorithm to account for constraints regarding the gamma shape parameter λ, allowing for upper λ_{max} and lower boundaries λ_{min}. Recall that for n-dimensional optimization problems the algorithm uses an n-simplex with $(n+1)$ vertices as function evaluation points (see Section 4.2.8). Since we do not want to modify the GSL implementation of the core algorithm, we addressed this by returning modified function values f^* to the GSL interface for any $\lambda < \lambda_{min}$ or $\lambda > \lambda_{max}$. We generate an artificial likelihood landscape using mirrored or boundary function values in combination with penalty terms, aiming to prevent the simplex from moving to far outside of the constrained parameter space. The penalty terms are based on the distance between the current parameter value λ and the corresponding boundary. Note that we use these soft constraints to ensure that the algorithm remains able to find optima within the valid parameter space that are located close to the parameter boundaries.

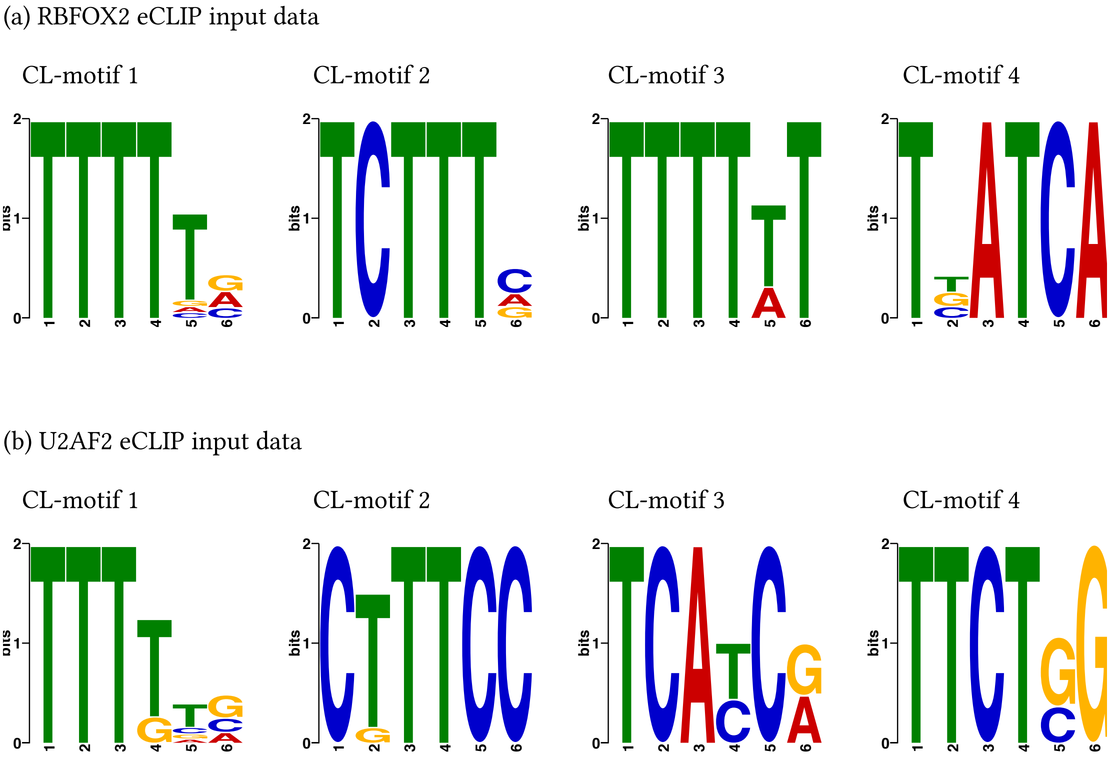
Let f be the original objective function corresponding to Equation 5.10, which is dependent on the parameters μ and λ. Algorithm A.1 depicts the computation of the modified function value f^*. Equivalently modified function values are computed for Equation 5.4.1.

```
function $f_{GSL}(\mu, \lambda)$
    $p \leftarrow 0.01$                                              ▷ Penalty factor
    $\epsilon \leftarrow 0.001$
    if $\lambda < \lambda_{min}$ then
        $d = \lambda_{min} - \lambda$                                ▷ Distance to $\lambda_{min}$
        if $f(\mu, \lambda_{min} + \epsilon) - f(\mu, \lambda_{min}) > 0$ then   ▷ $f$ is descending towards $\lambda_{min}$
            $f^* = f(\mu, \lambda_{min} + d)$                        ▷ Mirrored function value
                $+ (d \cdot f(\mu, \lambda_{min} + d) \cdot p)^2$    ▷ added penalty
        else                                                         ▷ $f$ is ascending towards $\lambda_{min}$
            $f^* = f(\mu, \lambda_{min})$                            ▷ Function value at $\lambda_{min}$
                $+ (d \cdot f(\mu, \lambda_{min}) \cdot p)^2$        ▷ added penalty
        end if
    else if $\lambda > \lambda_{max}$ then
        ...                                      ▷ Computation accordingly for the other side
    else                                                             ▷ Within valid parameter space
        $f^* = f(\mu, \lambda)$                                      ▷ Use unmodified value
    end if
    return $f^*$
end function
```

Algorithm A.1: Modified objective function used for the GSL Nelder-Mead routine to implement soft constraints.

A.5. Computation of CL-motif scores

To learn the CL-motifs on the input eCLIP data we run DREME (meme suite v4.11.3) [10] with the parameters '-norc -k 6 -4' on 10 bp windows spanning the called crosslink sites, while using 10 bp windows 20 bp upstream and downstream as control. We use FIMO (meme suite v4.11.3) [55] with the parameters (--thresh 0.01 --norc) to compute the corresponding motif matches. Figure A.2 shows the learned CL-motifs for RBFOX2 and U2AF2 eCLIP input data.


(a) RBFOX2 eCLIP input data

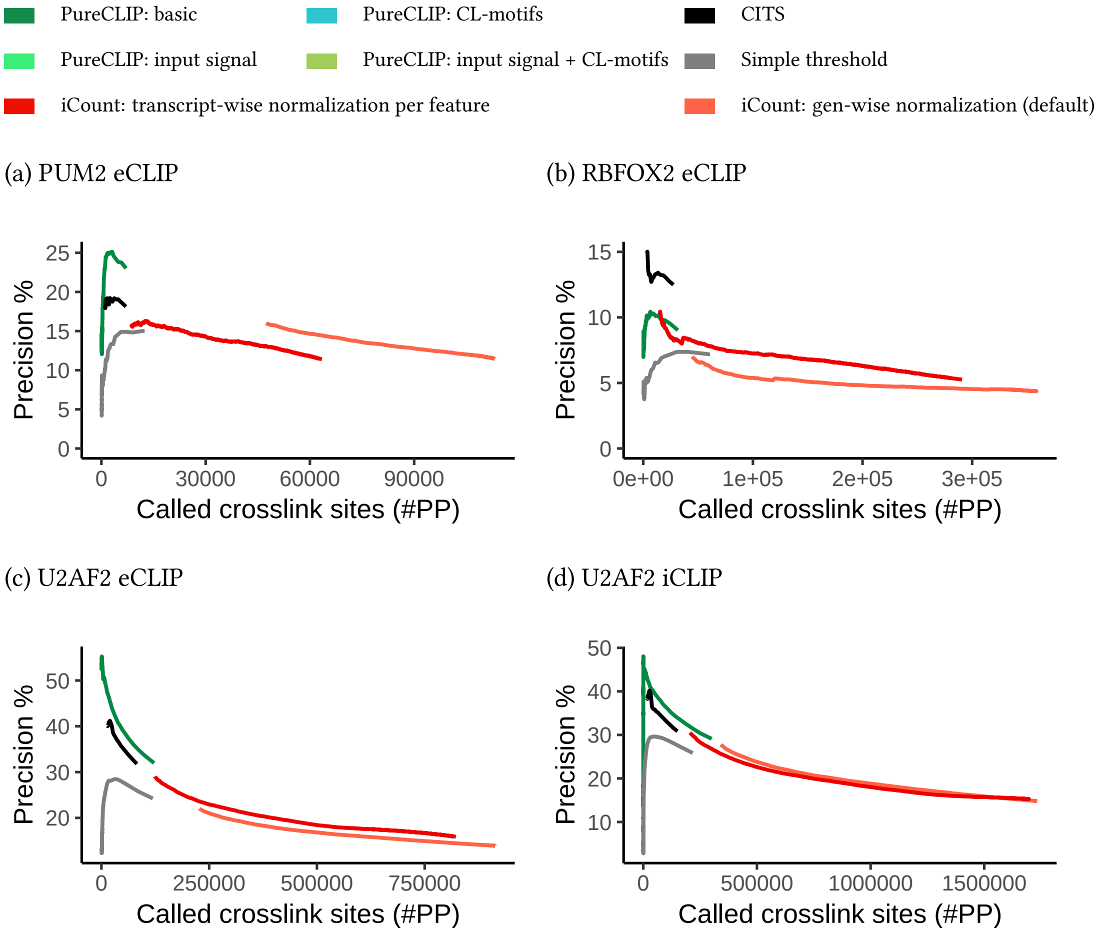
(b) U2AF2 eCLIP input data

Figure A.2.: CL-motifs as shown in Figure 7.4a, but for RBFOX2 and U2AF2 eCLIP input data.

A.6. Evaluation of called crosslink sites

Table A.1.: Number of crosslink sites reported by different methods.

	PUM2 eCLIP	RBFOX2 eCLIP	U2AF2 eCLIP
simple threshold (5)	12,449	60,891	119,195
CITS	7,045	27,446	82,454
iCount	113,189	358,987	913,179
PureCLIP: basic	7,106	31,554	122,223
PureCLIP: input	5,639	24,815	117,147
PureCLIP: CL-motifs	5,958	17,748	91,545
PureCLIP: input signal + CL-motifs	4,861	14,308	89,718



(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

(d) U2AF2 iCLIP

Figure A.3.: Precision of the called crosslink sites shown for the entire range of #PPs obtained by applying different score thresholds.

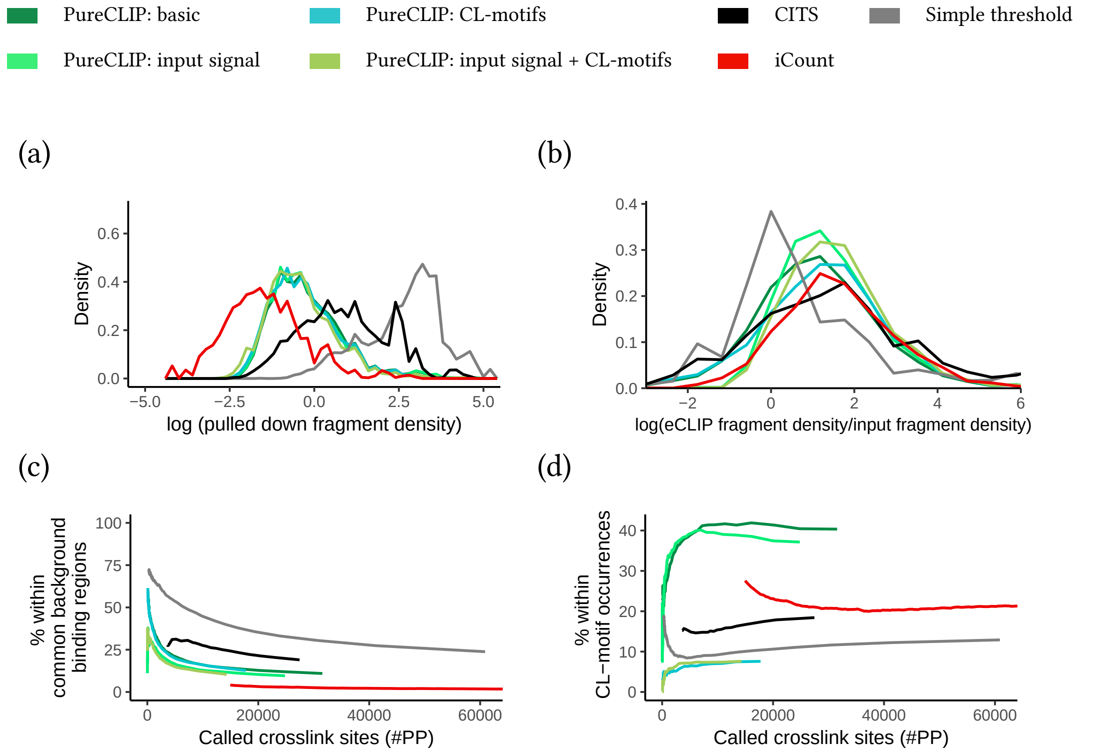


Figure A.4.: Same as Figure 8.6, but for RBFOX2 eCLIP data and using the top 5000 called crosslink sites for **a)** and **b)**.

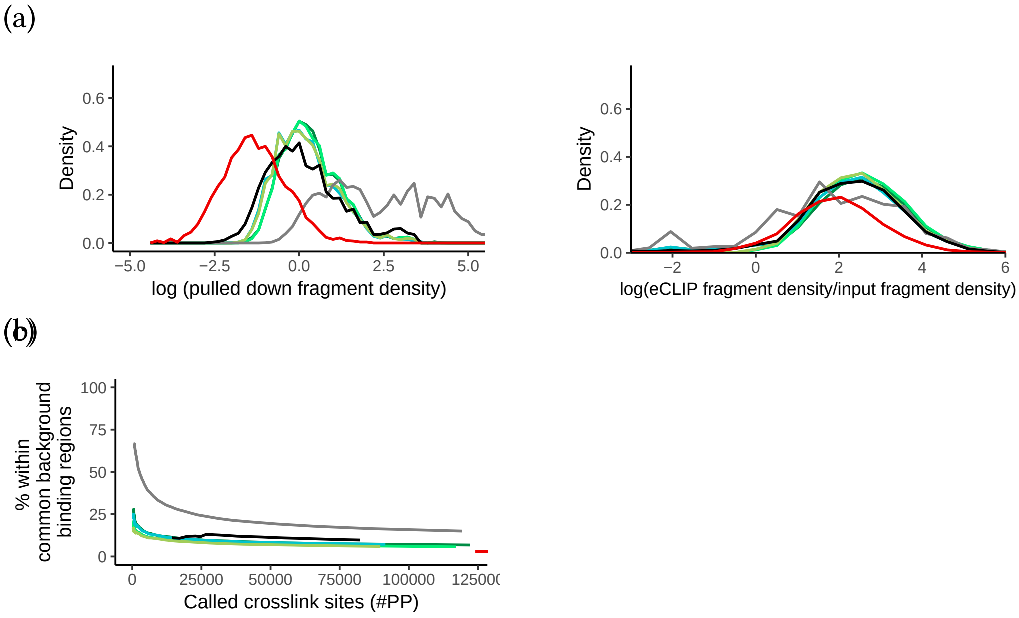


Figure A.5.: Same as Figure 8.6, but for U2AF2 eCLIP data and using the top 5000 called crosslink sites for **a)** and **b)**.

A.7. Evaluation of called binding regions

Table A.2.: Number of binding regions reported by different methods.

	PUM2 eCLIP	RBFOX2 eCLIP	U2AF2 eCLIP
Piranha	11,918	42,191	29,579
CLIPper	97,970	426,338	332,168
CITS	15,424	16,651	44,134
iCount	25,680	87,113	196,753
PureCLIP: basic	3,373	18,707	54,087
PureCLIP: input	2,528	14,607	50,404
PureCLIP: CL-motifs	2,862	13,592	44,970
PureCLIP: input signal + CL-motifs	2,223	10,949	42,702

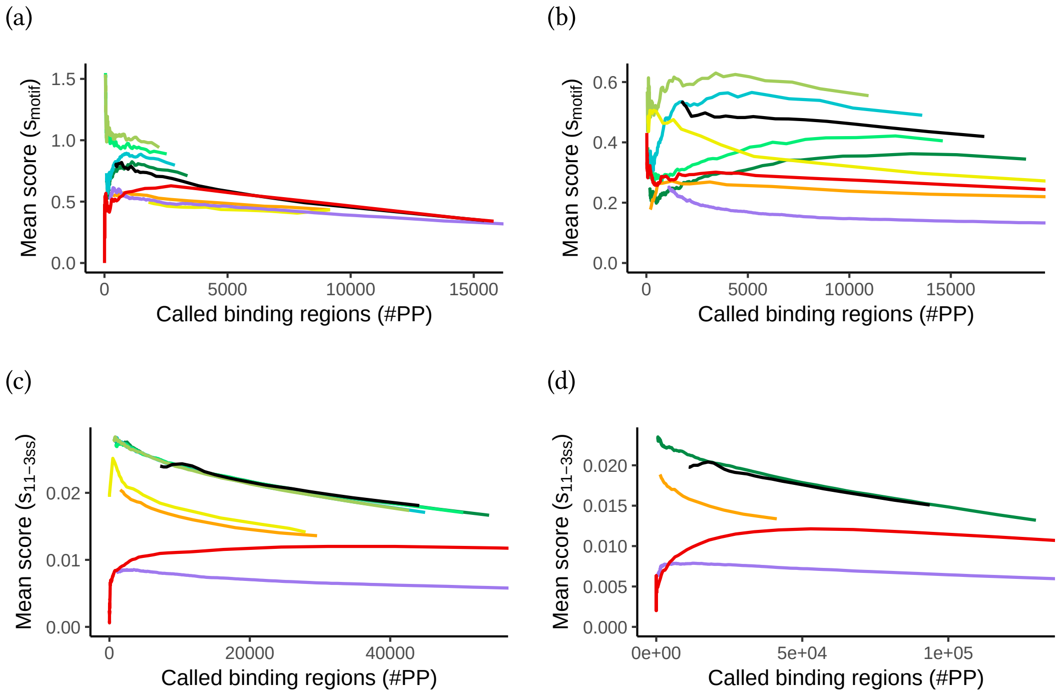


Figure A.6.: Same as Figure 8.7, but using bandwidth of 10 nt to compute region-wise scores.

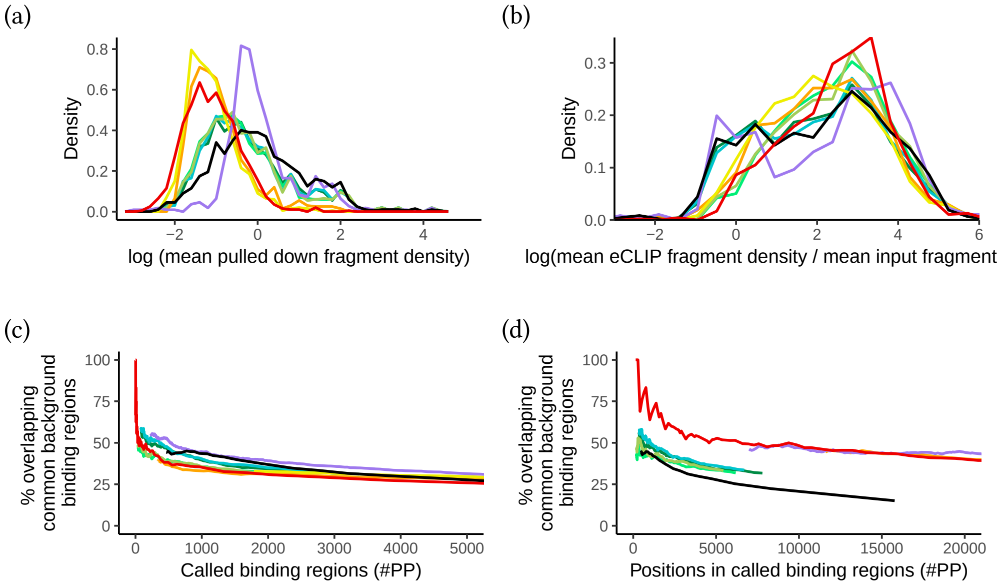
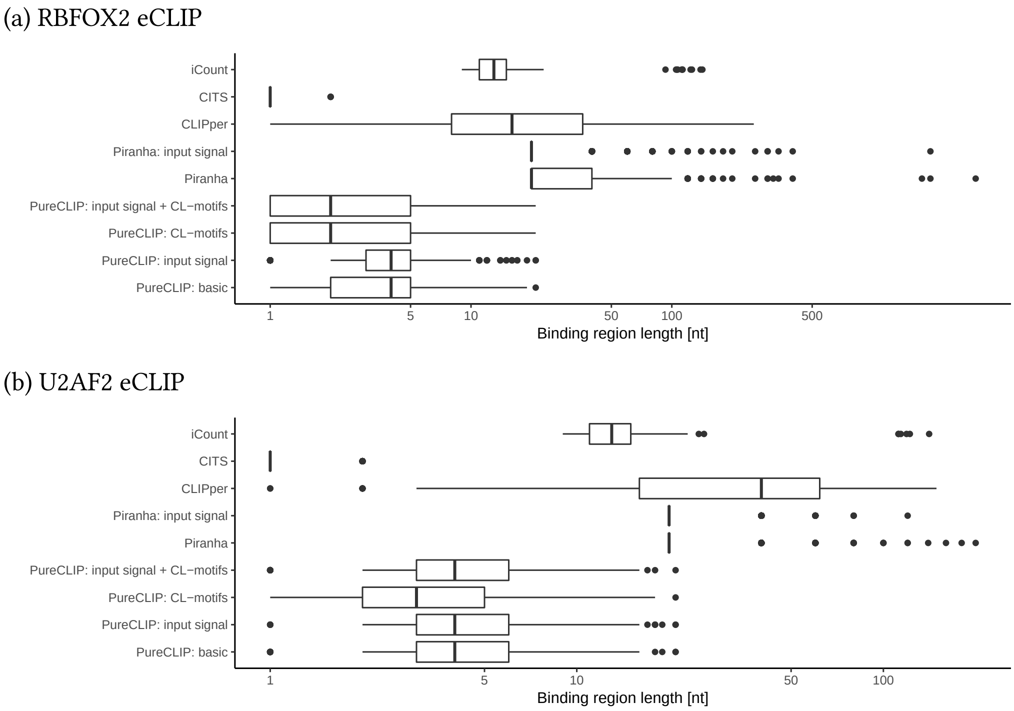


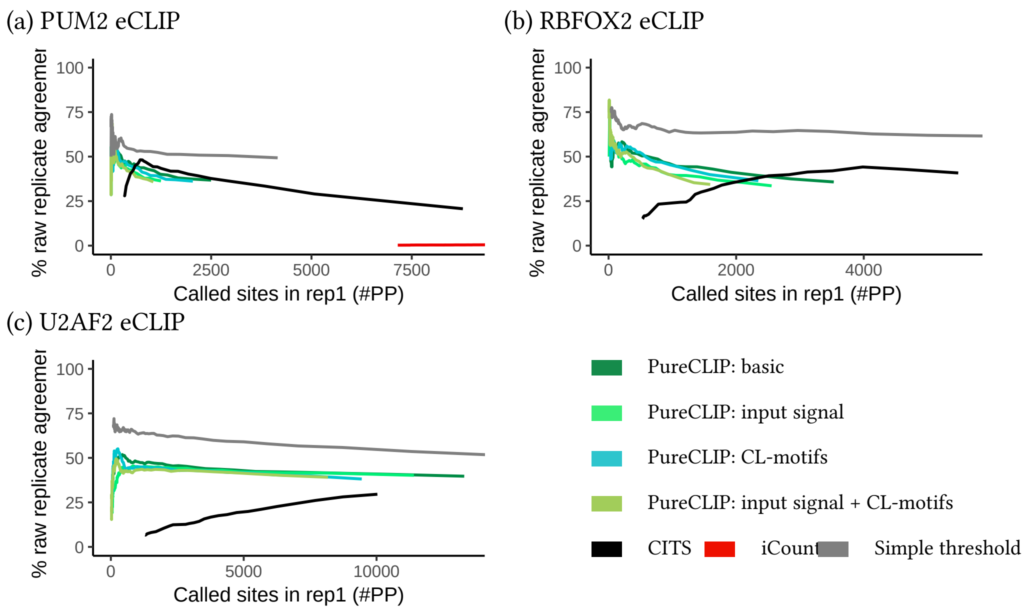
Figure A.7.: Characteristics of reported binding regions with respect to bias prone regions exemplarily shown for PUM2 eCLIP data. **a)** Distribution of log mean fragment density values and **b)** log-fold mean fragment density enrichments over the input for the top 1000 called binding regions. **c)** Fraction of called binding regions overlapping common background binding regions and **d)** fraction of positions within called binding regions overlapping common background binding regions for different #PPs.


(a) RBFOX2 eCLIP

(b) U2AF2 eCLIP

Figure A.8.: Length distributions of called binding regions.

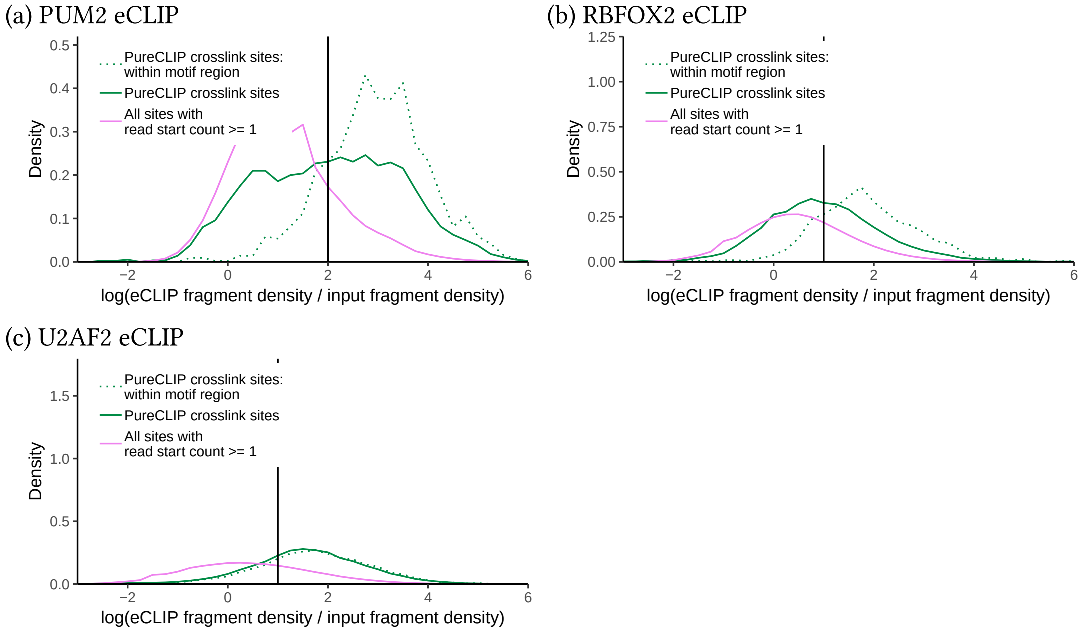
A.8. Replicate agreement


(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

Figure A.9.: Raw replicate agreement on eCLIP datasets. For each given number x of called sites in one replicate, we report the percentage that were also found within the x top ranking called sites of the other replicate.


(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

Figure A.10.: Distribution of log-fold fragment density enrichments over the input for sites called by PureCLIP, for sites called by PureCLIP and located within the target motif, and for all sites with at least one read start. Based on the distributions a threshold is set for bias correction (black vertical line).

A.9. Model choices

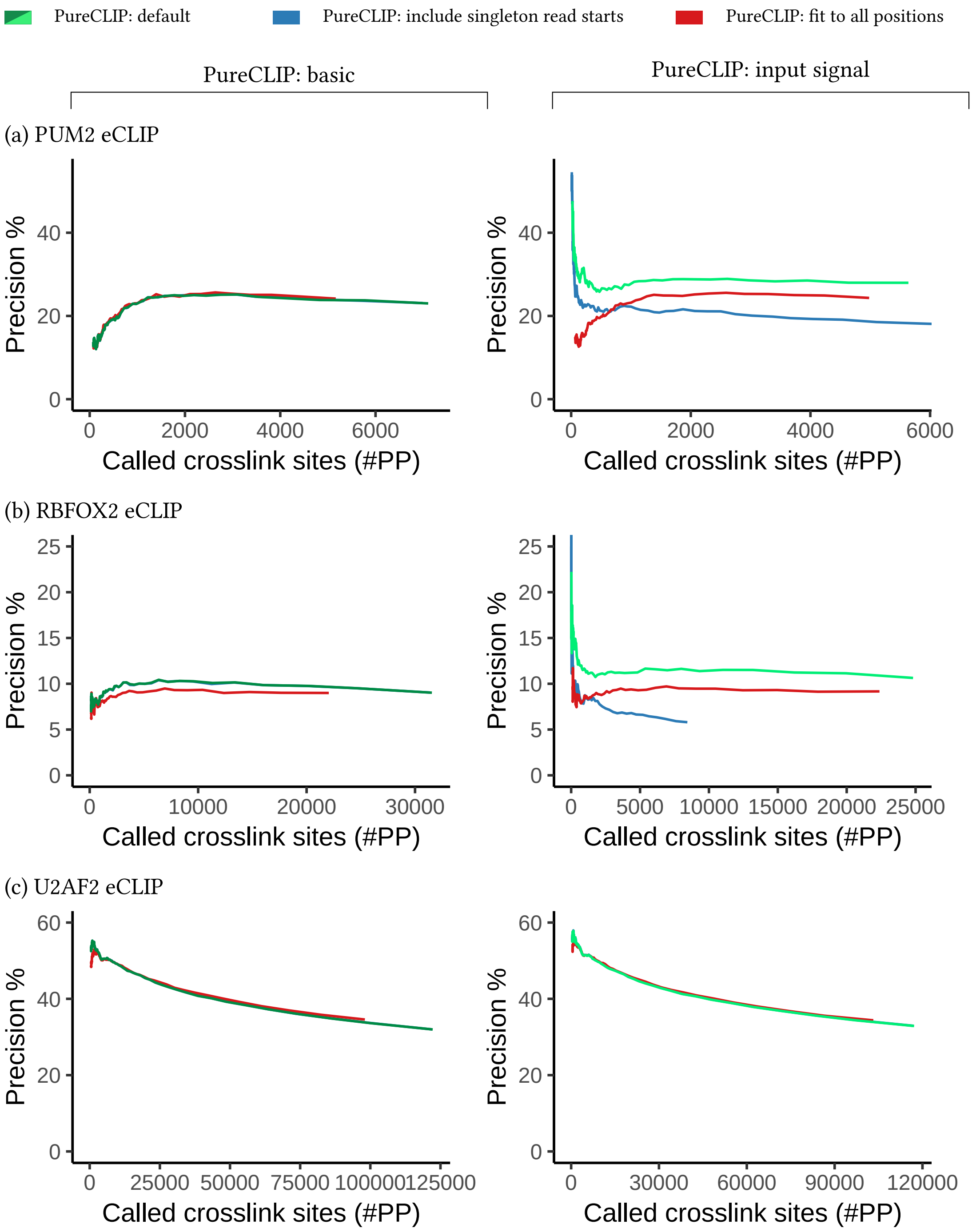


Figure A.11.: Precision of PureCLIP in default setting (using left-truncated gamma distributions fitted to sites with >= 1 read starts, discarding singleton reads) in comparison 1) to using non-truncated gamma distributions fitted to all positions, i.e. also positions with no read start, and 2) to not discarding singleton read starts.

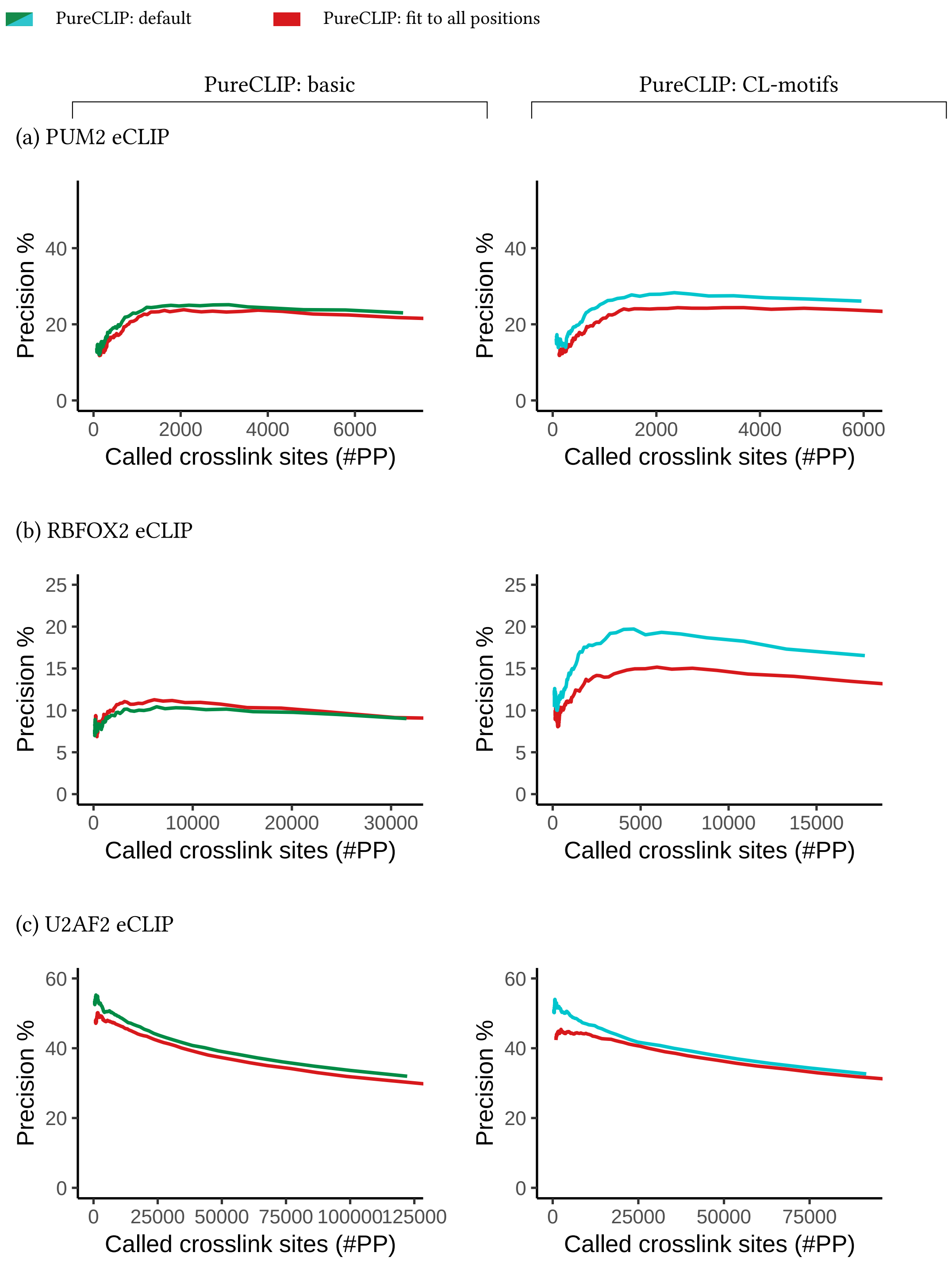
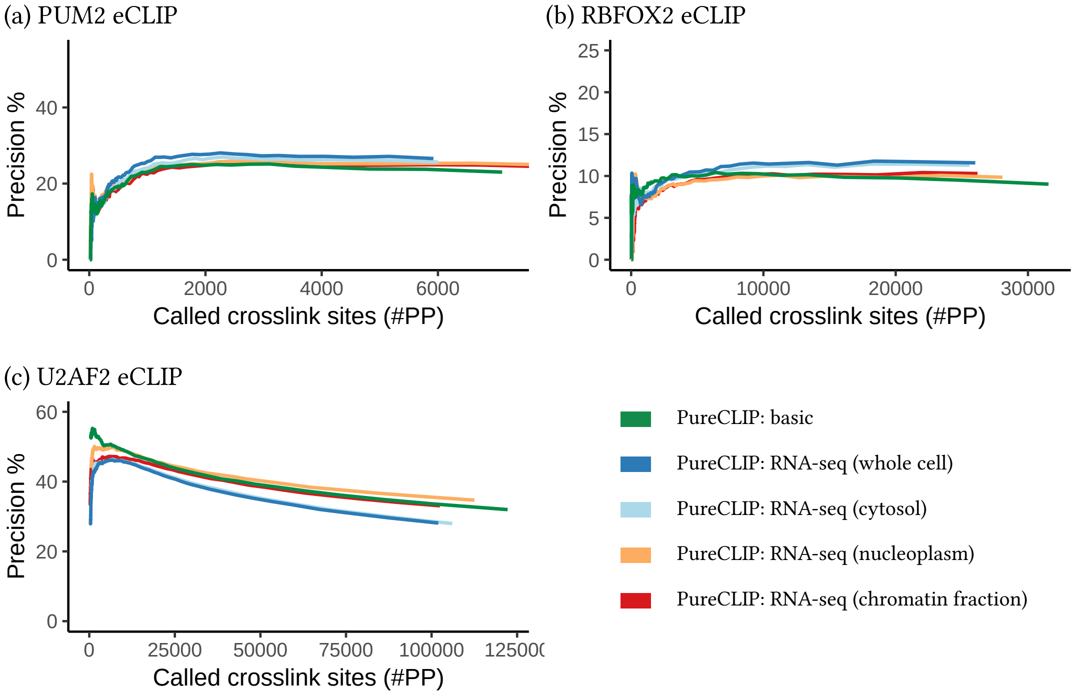


Figure A.12.: Precision of PureCLIP in default setting (using zero-truncated binomial distributions fitted to sites with >= 1 read starts) in comparison to PureCLIP when using non-truncated binomial distribution fitted to all positions, i.e. also to positions with no read start.

A.10. Incorporation of RNA-seq data


(a) PUM2 eCLIP

(b) RBFOX2 eCLIP

(c) U2AF2 eCLIP

Figure A.13.: PureCLIP's performance when including RNA-seq data from different cellular compartments.

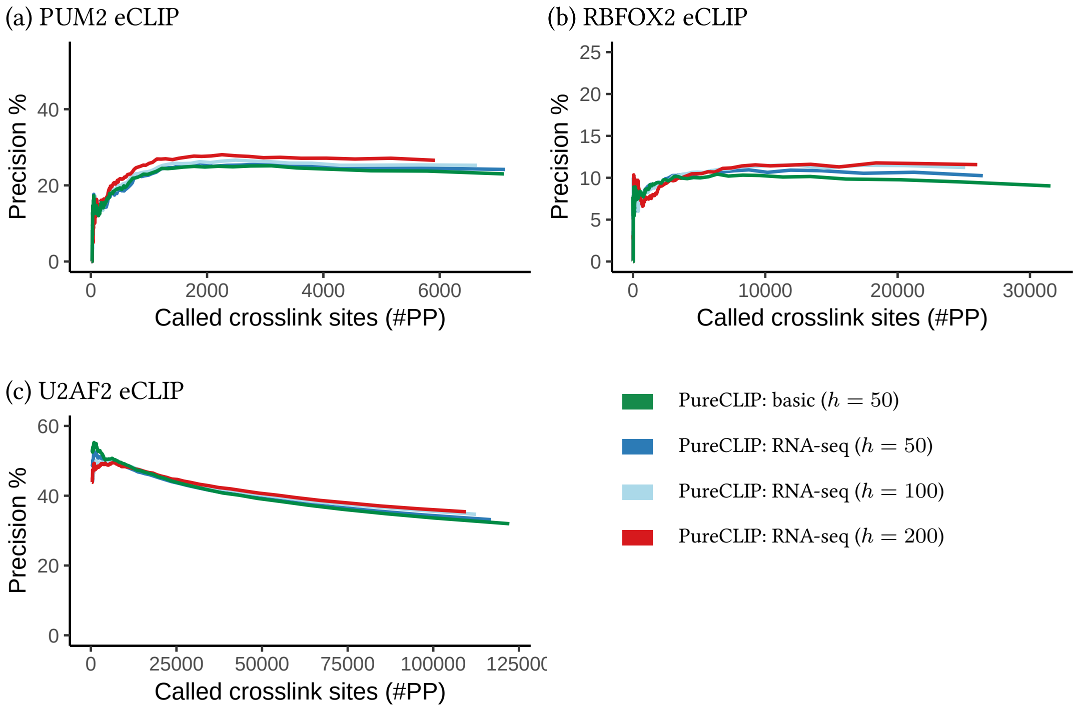


Figure A.14.: Precision of PureCLIP when including RNA-seq data from best suitable cellular compartment for different bandwidths h in comparison to PureCLIP in basic mode. **a)** and **b)** Incorporation of whole cell RNA-seq data and **c)** of nucleoplasmic RNA-seq data.

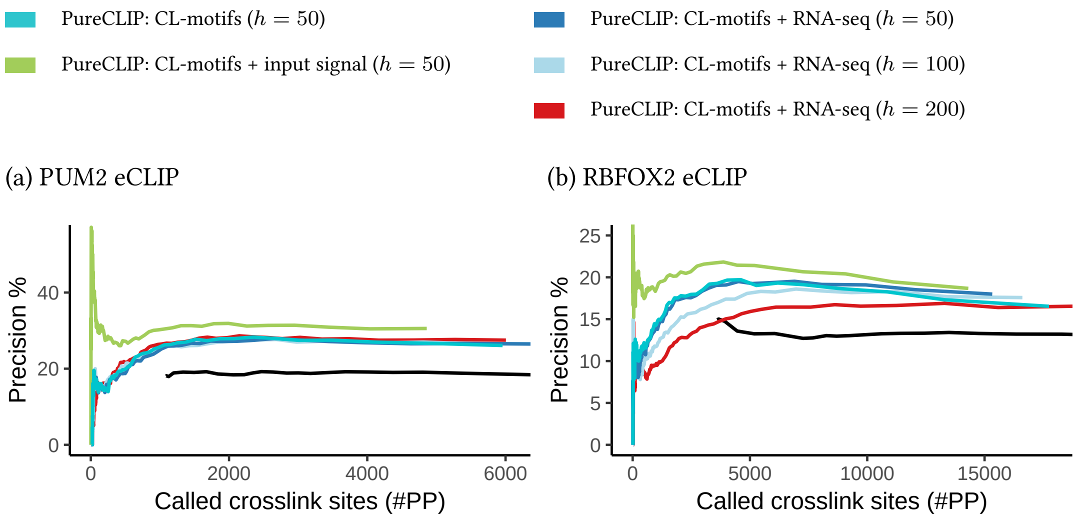


Figure A.15.: PureCLIP's performance when including whole cell RNA-seq data in combination with CL-motifs for different bandwidths h in comparison to PureCLIP in basic mode.

A.11. PTBP1 iCLIP

Additionally, since iCount calls much more sites than other methods we aimed for a comparison in a higher #PP range. The reason for this is that in previous studies iCount was used on PTBP1 CLIP data using its default sensitivity setting [24] and we do not want to exclude a for iCount potentially more optimal #PP range from the comparison. This is particular interesting on this dataset, since iCount uses a moving sum to call crosslink sites (see Sections 3.6.2) and, furthermore, normalizes these for broader regions in comparison to PureCLIP, which might be valuable for this type of data. The used default *half-window* parameter of 3 nt corresponds to the optimal parameter for PTBP1 as suggested by Haberman [60]. Consequently we expected iCount to perform better on this data in comparison to PUM2 or RBFOX2 eCLIP data. Since PureCLIP calls only 25,258 crosslink sites in its defaults setting, we run it using a setting optimized for longer crosslink clusters (see Section A.12), which increases its sensitivity and allows a comparison. We then explored the distribution of the top 100,000 called crosslink sites or regions around the 3' splice sites of silenced exons (see Figure A.16).

We observed that iCount calls the highest density of crosslink sites ~ 35 nt upstream of 3' splice sites of silenced exons, while PureCLIP shows a higher density of binding regions in this area.

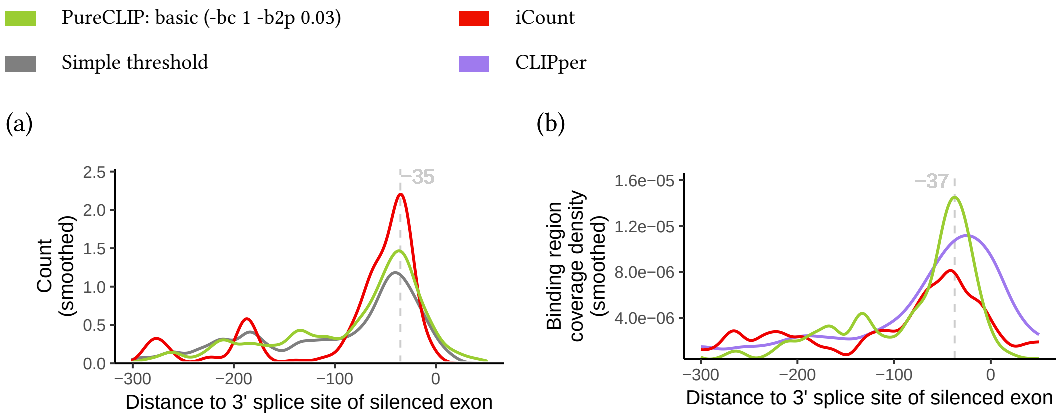


Figure A.16.: Same as Figure 9.6, but using the top 100,000 called **(a)** crosslink sites and **(b)** binding regions. Methods calling less than 100,000 crosslink sites or regions were excluded from the comparison.

A.12. PureCLIP parameters

The most important user parameters PureCLIP depends on are the following:

- **Read to use** `-ur <num>`

  Flag to define which read of a pair should be used for the analysis: 1->R1, 2->R2. Default: all.

- **Position of crosslink site** `-ctr`

  Use position of read starts as potential crosslink sites. Default: position upstream of read starts.

- **Training set** `-iv <id>[;<id>;...]`

  The set of chromosomes to use for training the HMM. Default: all.

- **KDE bandwidth** `-bw <num>`

  Bandwidth of kernel density estimation used to compute fragment densities. Default: 50.

- **Distance** `-dm <num>`

  Distance used to merge individual crosslink sites to binding regions. Default: 8.

- **Binding characteristics** `-bc <num>`

  Flag to optimize parameters according to binding characteristics of protein: e.g. for proteins causing larger crosslink clusters with relatively lower read start counts. Default: 0, i.e. the parameters are optimized for proteins binding to short defined binding regions.

- **Number of threads** `-nt <num>`

  Number of threads used for learning. Default: 1.

Bibliography

[1] E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Čech, J. Chilton, D. Clements, N. Coraor, B. A. Grüning, A. Guerler, J. Hillman-Jackson, S. Hiltemann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko, and D. Blankenberg. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research*, 46(W1): W537–W544, jul 2018. ISSN 0305-1048. doi: 10.1093/nar/gky379.

[2] T. Afroz, Z. Cienikova, A. Cléry, and F. H. Allain. One, Two, Three, Four! How Multiple RRMs Read the Genome Sequence. In *Methods in Enzymology*, pages 235–278. Elsevier, 2015. ISBN 9780128019344. doi: 10.1016/bs.mie.2015.01.015.

[3] A. A. Agrawal, E. Salsi, R. Chatrikhi, S. Henderson, J. L. Jenkins, M. R. Green, D. N. Ermolenko, and C. L. Kielkopf. An extended U2AF65-RNA-binding domain recognizes the 3' splice site signal. *Nature Communications*, 2016. ISSN 20411723. doi: 10.1038/ncomms10950.

[4] T. Aktaş, İ. A. Ilık, D. Maticzka, V. Bhardwaj, C. P. Rodrigues, G. Mittler, T. Manke, R. Backofen, and A. Akhtar. DHX9 suppresses RNA processing defects originating from the Alu invasion of the human genome. *Nature*, 544(7648):115–119, mar 2017. ISSN 0028-0836. doi: 10.1038/nature21715.

[5] C. D. Allis and T. Jenuwein. The molecular hallmarks of epigenetic control, 2016. ISSN 14710064.

[6] A. Ameur, A. Zaghlool, J. Halvardson, A. Wetterbom, U. Gyllensten, L. Cavelier, and L. Feuk. Total RNA sequencing reveals nascent transcription and widespread co-transcriptional splicing in the human brain. *Nature Structural and Molecular Biology*, 2011. ISSN 15459993. doi: 10.1038/nsmb.2143.

[7] K. Atkinson. *An Introduction to Numerical Analysis (2nd Edition)*. Wiley, 2nd edition, 1989. ISBN 9780471624899.

[8] S. D. Auweter, F. C. Oberstrass, and F. H. T. Allain. Sequence-specific binding of single-stranded RNA: Is there a code for recognition? *Nucleic Acids Research*, 2006. ISSN 03051048. doi: 10.1093/nar/gkl620.

[9] E. Bahrami-Samani and Y. Xing. Discovery of allele-specific protein-RNA interactions in human transcriptomes. *bioRxiv*, 2018. doi: 10.1101/389205.

[10] T. L. Bailey. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics*, 27(12):1653–1659, 2011.

[11] A. G. Baltz, M. Munschauer, B. Schwanhäusser, A. Vasile, Y. Murakawa, M. Schueler, N. Youngs, D. Penfold-Brown, K. Drew, M. Milek, E. Wyler, R. Bonneau, M. Selbach, C. Dieterich, and M. Landthaler. The mRNA-Bound Proteome and Its Global Occupancy Profile on Protein-Coding Transcripts. *Molecular Cell*, 46(5):674–690, jun 2012. ISSN 10972765. doi: 10.1016/j.molcel.2012.05.021.

[12] E. Behrmann, J. Loerke, T. V. Budkevich, K. Yamamoto, A. Schmidt, P. A. Penczek, M. R. Vos, J. Bürger, T. Mielke, P. Scheerer, and C. M. Spahn. Structural Snapshots of Actively Translating Human Ribosomes. *Cell*, 161(4):845–857, may 2015. ISSN 00928674. doi: 10.1016/j.cell.2015.03.052.

[13] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, G. P. Smith, J. Milton, C. G. Brown, K. P. Hall, D. J. Evers, C. L. Barnes, H. R. Bignell, J. M. Boutell, J. Bryant, R. J. Carter, R. Keira Cheetham, A. J. Cox, D. J. Ellis, M. R. Flatbush, N. A. Gormley, S. J. Humphray, L. J. Irving, M. S. Karbelashvili, S. M. Kirk, H. Li, X. Liu, K. S. Maisinger, L. J. Murray, B. Obradovic, T. Ost, M. L. Parkinson, M. R. Pratt, I. M. J. Rasolonjatovo, M. T. Reed, R. Rigatti, C. Rodighiero, M. T. Ross, A. Sabot, S. V. Sankar, A. Scally, G. P. Schroth, M. E. Smith, V. P. Smith, A. Spiridou, P. E. Torrance, S. S. Tzonev, E. H. Vermaas, K. Walter, X. Wu, L. Zhang, M. D. Alam, C. Anastasi, I. C. Aniebo, D. M. D. Bailey, I. R. Bancarz, S. Banerjee, S. G. Barbour, P. A. Baybayan, V. A. Benoit, K. F. Benson, C. Bevis, P. J. Black, A. Boodhun, J. S. Brennan, J. A. Bridgham, R. C. Brown, A. A. Brown, D. H. Buermann, A. A. Bundu, J. C. Burrows, N. P. Carter, N. Castillo, M. Chiara E. Catenazzi, S. Chang, R. Neil Cooley, N. R. Crake, O. O. Dada, K. D. Diakoumakos, B. Dominguez-Fernandez, D. J. Earnshaw, U. C. Egbujor, D. W. Elmore, S. S. Etchin, M. R. Ewan, M. Fedurco, L. J. Fraser, K. V. Fuentes Fajardo, W. Scott Furey, D. George, K. J. Gietzen, C. P. Goddard, G. S. Golda, P. A. Granieri, D. E. Green, D. L. Gustafson, N. F. Hansen, K. Harnish, C. D. Haudenschild, N. I. Heyer, M. M. Hims, J. T. Ho, A. M. Horgan, K. Hoschler, S. Hurwitz, D. V. Ivanov, M. Q. Johnson, T. James, T. A. Huw Jones, G.-D. Kang, T. H. Kerelska, A. D. Kersey, I. Khrebtukova, A. P. Kindwall, Z. Kingsbury, P. I. Kokko-Gonzales, A. Kumar, M. A. Laurent, C. T. Lawley, S. E. Lee, X. Lee, A. K. Liao, J. A. Loch, M. Lok, S. Luo, R. M. Mammen, J. W. Martin, P. G. McCauley, P. McNitt, P. Mehta, K. W. Moon, J. W. Mullens, T. Newington, Z. Ning, B. Ling Ng, S. M. Novo, M. J. O'Neill, M. A. Osborne, A. Osnowski, O. Ostadan, L. L. Paraschos, L. Pickering, A. C. Pike, A. C. Pike, D. Chris Pinkard, D. P. Pliskin, J. Podhasky, V. J. Quijano, C. Raczy, V. H. Rae, S. R. Rawlings, A. Chiva Rodriguez, P. M. Roe, J. Rogers, M. C. Rogert Bacigalupo, N. Romanov, A. Romieu, R. K. Roth, N. J. Rourke, S. T. Ruediger, E. Rusman, R. M. Sanches-Kuiper, M. R. Schenker, J. M. Seoane, R. J. Shaw, M. K. Shiver, S. W. Short, N. L. Sizto, J. P. Sluis, M. A. Smith, J. Ernest Sohna Sohna, E. J. Spence, K. Stevens, N. Sutton, L. Szajkowski, C. L. Tregidgo, G. Turcatti, S. VandeVondele, Y. Verhovsky, S. M. Virk, S. Wakelin, G. C. Walcott, J. Wang,

G. J. Worsley, J. Yan, L. Yau, M. Zuerlein, J. Rogers, J. C. Mullikin, M. E. Hurles, N. J. McCooke, J. S. West, F. L. Oaks, P. L. Lundberg, D. Klenerman, R. Durbin, and A. J. Smith. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–59, nov 2008. ISSN 0028-0836. doi: 10.1038/nature07517.

[14] J. X. Binder, S. Pletscher-Frankild, K. Tsafou, C. Stolte, S. I. O'Donoghue, R. Schneider, and L. J. Jensen. COMPARTMENTS: Unification and visualization of protein subcellular localization evidence. *Database*, 2014. ISSN 17580463. doi: 10.1093/database/bau012.

[15] A. Bird. DNA methylation patterns and epigenetic memory, 2002. ISSN 08909369.

[16] R. P. Brent. Algorithms for Minimization Without Derivatives. *IEEE Transactions on Automatic Control*, 1974. ISSN 15582523. doi: 10.1109/TAC.1974.1100629.

[17] M. Brugiolo, V. Botti, N. Liu, M. Müller-McNicoll, and K. M. Neugebauer. Fractionation iCLIP detects persistent SR protein binding to conserved, retained introns in chromatin, nucleoplasm and cytoplasm. *Nucleic Acids Research*, 2017. ISSN 13624962. doi: 10.1093/nar/gkx671.

[18] S. Budach and A. Marsico. Pysster: Classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. *Bioinformatics*, 2018. ISSN 14602059. doi: 10.1093/bioinformatics/bty222.

[19] K. Burger, B. Mühl, M. Kellner, M. Rohrmoser, A. Gruber-Eber, L. Windhager, C. C. Friedel, L. Dölken, and D. Eick. 4-thiouridine inhibits rRNA synthesis and causes a nucleolar stress response. *RNA Biology*, 10(10):1623–1630, oct 2013. ISSN 1547-6286. doi: 10.4161/rna.26214.

[20] W. A. Cantara, P. F. Crain, J. Rozenski, J. A. McCloskey, K. A. Harris, X. Zhang, F. A. P. Vendeix, D. Fabris, and P. F. Agris. The RNA modification database, RNAMDB: 2011 update. *Nucleic Acids Research*, 39(Database):D195–D201, jan 2011. ISSN 0305-1048. doi: 10.1093/nar/gkq1028.

[21] A. Castello, B. Fischer, K. Eichelbaum, R. Horos, B. M. Beckmann, C. Strein, N. E. Davey, D. T. Humphreys, T. Preiss, L. M. Steinmetz, J. Krijgsveld, and M. W. Hentze. Insights into RNA Biology from an Atlas of Mammalian mRNA-Binding Proteins. *Cell*, 149(6):1393–1406, jun 2012. ISSN 00928674. doi: 10.1016/j.cell.2012.04.031.

[22] A. Castello, B. Fischer, M. W. Hentze, and T. Preiss. RNA-binding proteins in Mendelian disease. *Trends in Genetics*, 29(5):318–327, may 2013. ISSN 01689525. doi: 10.1016/j.tig.2013.01.004.

[23] T. R. Cech and J. A. Steitz. The Noncoding RNA Revolution—Trashing Old Rules to Forge New Ones. *Cell*, 157(1):77–94, mar 2014. ISSN 00928674. doi: 10.1016/j.cell.2014.03.008.

[24] A. M. Chakrabarti, N. Haberman, A. Praznik, N. M. Luscombe, and J. Ule. Data Science Issues in Studying Protein–RNA Interactions with CLIP Technologies. *Annual Review of Biomedical Data Science*, 2018. ISSN 2574-3414. doi: 10.1146/annurev-biodatasci-080917-013525.

[25] B. Chen, J. Yun, M. S. Kim, J. T. Mendell, and Y. Xie. PIPE-CLIP: a comprehensive online tool for CLIP-seq data analysis. *Genome Biol*, 15(1):R18, 2014.

[26] X. Chen, S. A. Castro, Q. Liu, W. Hu, and S. Zhang. Practical considerations on performing and analyzing CLIP-seq experiments to identify transcriptomic-wide RNA-Protein interactions. *Methods*, 2018.

[27] Y. Chen and L. Pollack. SAXS studies of RNA: structures, dynamics, and interactions with partners. *Wiley Interdisciplinary Reviews: RNA*, 7(4):512–526, jul 2016. ISSN 17577004. doi: 10.1002/wrna.1349.

[28] C. Clerte and K. B. Hall. Characterization of multimeric complexes formed by the human PTB1 protein on RNA. *RNA*, 2006. ISSN 13558382. doi: 10.1261/rna.2178406.

[29] M. B. Coelho, J. Attig, N. Bellora, J. Konig, M. Hallegger, M. Kayikci, E. Eyras, J. Ule, and C. W. Smith. Nuclear matrix protein Matrin3 regulates alternative splicing and forms overlapping regulatory networks with PTB. *The EMBO Journal*, 2015. ISSN 0261-4189. doi: 10.15252/embj.201489852.

[30] A. C. Cohen. *Truncated and censored samples : theory and applications / A. Clifford Cohen.* Dekker, New York u.a., 1991. ISBN 0-8247-8447-2.

[31] E. P. Consortium, I. Dunham, A. Kundaje, S. F. Aldred, P. J. Collins, C. a. Davis, F. Doyle, C. B. Epstein, S. Frietze, J. Harrow, R. Kaul, J. Khatun, B. R. Lajoie, S. G. Landt, B.-K. B.-K. Lee, F. Pauli, K. R. Rosenbloom, P. Sabo, A. Safi, A. Sanyal, N. Shoresh, J. M. Simon, L. Song, N. D. Trinklein, R. C. Altshuler, E. Birney, J. B. Brown, C. Cheng, S. Djebali, X. Dong, J. Ernst, T. S. Furey, M. Gerstein, B. Giardine, M. Greven, R. C. Hardison, R. S. Harris, J. Herrero, M. M. Hoffman, S. Iyer, M. Kelllis, P. Kheradpour, T. Lassman, Q. Li, X. Lin, G. K. Marinov, A. Merkel, A. Mortazavi, S. C. J. S. L. Parker, T. E. Reddy, J. Rozowsky, F. Schlesinger, R. E. Thurman, J. Wang, L. D. Ward, T. W. Whitfield, S. P. Wilder, W. Wu, H. S. Xi, K. Y. Yip, J. Zhuang, B. E. Bernstein, E. D. Green, C. Gunter, M. Snyder, M. J. Pazin, R. F. Lowdon, L. a. L. Dillon, L. B. Adams, C. J. Kelly, J. Zhang, J. R. Wexler, P. J. Good, E. a. Feingold, G. E. Crawford, J. Dekker, L. Elinitski, P. J. Farnham, M. C. Giddings, T. R. Gingeras, R. Guigó, T. J. T. J. Hubbard, M. Kellis, W. J. Kent, J. D. Lieb, E. H. Margulies, R. M. Myers, J. a. Starnatoyannopoulos, S. a. Tennebaum, Z. Weng, K. P. White, B. Wold, Y. Yu, J. Wrobel, B. a. Risk, H. P. Gunawardena, H. C. Kuiper, C. W. Maier, L. Xie, X. Chen, T. S. Mikkelsen, S. Gillespie, A. Goren, O. Ram, X. Zhang, L. Wang, R. Issner, M. J. Coyne, T. Durham, M. Ku, T. Truong, M. L. Eaton, A. Dobin, T. Lassmann, A. Tanzer, J. Lagarde, W. Lin, C. Xue, B. a.

Williams, C. Zaleski, M. Röder, F. Kokocinski, R. F. Abdelhamid, T. Alioto, I. Antoshechkin, M. T. Baer, P. Batut, I. Bell, K. Bell, S. Chakrabortty, J. Chrast, J. Curado, T. Derrien, J. Drenkow, E. Dumais, J. Dumais, R. Duttagupta, M. Fastuca, K. Fejes-Toth, P. Ferreira, S. Foissac, M. J. Fullwood, H. Gao, D. Gonzalez, A. Gordon, C. Howald, S. Jha, R. Johnson, P. Kapranov, B. King, C. Kingswood, G. Li, O. J. Luo, E. Park, J. B. Preall, K. Presaud, P. Ribeca, D. Robyr, X. Ruan, M. Sammeth, K. S. Sandu, L. Schaeffer, L.-H. See, A. Shahab, J. Skancke, A. M. Suzuki, H. Takahashi, H. Tilgner, D. Trout, N. Walters, H. H. Wang, Y. Hayashizaki, A. Reymond, S. E. Antonarakis, G. J. Hannon, Y. Ruan, P. Carninci, C. a. Sloan, K. Learned, V. S. Malladi, M. C. Wong, G. P. Barber, M. S. Cline, T. R. Dreszer, S. G. Heitner, D. Karolchik, V. M. Kirkup, L. R. Meyer, J. C. Long, M. Maddren, B. J. Raney, L. L. Grasfeder, P. G. Giresi, A. Battenhouse, N. C. Sheffield, K. a. Showers, D. London, A. a. Bhinge, C. Shestak, M. R. Schaner, S. K. Kim, Z. Z. Z. Zhang, P. a. Mieczkowski, J. O. Mieczkowska, Z. Liu, R. M. McDaniell, Y. Ni, N. U. Rashid, M. J. Kim, S. Adar, T. Wang, D. Winter, D. Keefe, V. R. Iyer, K. S. Sandhu, M. Zheng, P. Wang, J. Gertz, J. Vielmetter, E. C. Partridge, K. E. Varley, C. Gasper, A. Bansal, S. Pepke, P. Jain, H. Amrhein, K. M. Bowling, M. Anaya, M. K. Cross, M. a. Muratet, K. M. Newberry, K. McCue, A. S. Nesmith, K. I. Fisher-Aylor, B. Pusey, G. DeSalvo, S. S. Balasubramanian, N. S. Davis, S. K. Meadows, T. Eggleston, J. S. Newberry, S. E. Levy, D. M. Absher, W. H. Wong, M. J. Blow, A. Visel, L. a. Pennachio, L. Elnitski, H. M. Petrykowska, A. Abyzov, B. Aken, D. Barrell, G. Barson, A. Berry, A. Bignell, V. Boychenko, G. Bussotti, C. Davidson, G. Despacio-Reyes, M. Diekhans, I. Ezkurdia, A. Frankish, J. Gilbert, J. M. Gonzalez, E. Griffiths, R. Harte, D. a. Hendrix, T. Hunt, I. Jungreis, M. Kay, E. Khurana, J. Leng, M. F. Lin, J. Loveland, Z. Lu, D. Manthravadi, M. Mariotti, J. Mudge, G. Mukherjee, C. Notredame, B. Pei, J. M. Rodriguez, G. Saunders, A. Sboner, S. Searle, C. Sisu, C. Snow, C. Steward, E. Tapanari, M. L. Tress, M. J. van Baren, S. Washieti, L. Wilming, A. Zadissa, Z. Zhengdong, M. Brent, D. Haussler, A. Valencia, A. Raymond, N. Addleman, R. P. Alexander, R. K. Auerbach, K. Bettinger, N. Bhardwaj, A. P. Boyle, A. R. Cao, P. Cayting, A. Charos, Y. Cheng, C. Eastman, G. Euskirchen, J. D. Fleming, F. Grubert, L. Habegger, M. Hariharan, A. Harmanci, S. Iyenger, V. X. Jin, K. J. Karczewski, M. Kasowski, P. Lacroute, H. Lam, N. Larnarre-Vincent, J. Lian, M. Lindahl-Allen, R. Min, B. Miotto, H. Monahan, Z. Moqtaderi, X. J. Mu, H. O'Geen, Z. Ouyang, D. Patacsil, D. Raha, L. Ramirez, B. Reed, M. Shi, T. Slifer, H. Witt, L. Wu, X. Xu, K.-K. Yan, X. Yang, K. Struhl, S. M. Weissman, S. a. Tenebaum, L. O. Penalva, S. Karmakar, R. R. Bhanvadia, A. Choudhury, M. Domanus, L. Ma, J. Moran, A. Victorsen, T. Auer, L. Centarin, M. Eichenlaub, F. Gruhl, S. Heerman, B. Hoeckendorf, D. Inoue, T. Kellner, S. Kirchmaier, C. Mueller, R. Reinhardt, L. Schertel, S. Schneider, R. Sinn, B. Wittbrodt, J. Wittbrodt, G. Jain, G. Balasundaram, D. L. Bates, R. Byron, T. K. Canfield, M. J. Diegel, D. Dunn, A. K. Ebersol, T. Frum, K. Garg, E. Gist, R. S. Hansen, L. Boatman, E. Haugen, R. Humbert, A. K. Johnson, E. M. Johnson, T. M. Kutyavin, K. Lee, D. Lotakis, M. T. Maurano, S. J. Neph, F. V. Neri, E. D. Nguyen, H. Qu, A. P. Reynolds, V. Roach, E. Rynes, M. E. Sanchez,

R. S. Sandstrom, A. O. Shafer, A. B. Stergachis, S. Thomas, B. Vernot, J. Vierstra, S. Vong, M. a. Weaver, Y. Yan, M. Zhang, J. a. Akey, M. Bender, M. O. Dorschner, M. Groudine, M. J. MacCoss, P. Navas, G. Stamatoyannopoulos, J. a. Stamatoyannopoulos, K. Beal, A. Brazma, P. Flicek, N. Johnson, M. Lukk, N. M. Luscombe, D. Sobral, J. M. Vaquerizas, S. Batzoglou, A. Sidow, N. Hussami, S. Kyriazopoulou-Panagiotopoulou, M. W. Libbrecht, M. a. Schaub, W. Miller, P. J. Bickel, B. Banfai, N. P. Boley, H. Huang, J. J. Li, W. S. Noble, J. a. Bilmes, O. J. Buske, A. O. Sahu, P. V. Kharchenko, P. J. Park, D. Baker, J. Taylor, and L. Lochovsky. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, sep 2012. ISSN 0028-0836. doi: 10.1038/nature11247.

[32] G. M. Cooper and J. Shendure. Needles in stacks of needles: Finding disease-causal variants in a wealth of genomic data, 2011. ISSN 14710056.

[33] D. L. Corcoran, S. Georgiev, N. Mukherjee, E. Gottwein, R. L. Skalsky, J. D. Keene, and U. Ohler. PARalyzer: definition of RNA binding sites from PAR-CLIP short-read sequence data. *Genome biology*, 12(8):R79, 2011.

[34] F. CRICK. Central Dogma of Molecular Biology. *Nature*, 227(5258):561–563, aug 1970. ISSN 0028-0836. doi: 10.1038/227561a0.

[35] A. Cullen and C. Frey. *Probabilistic Techniques in Exposure Assessment.* Springer Science & Business Media, 1999. ISBN 0306459566.

[36] T. Curk, G. Rot, Č. Gorup, J. Zmrzlikar, J. König, Y. Sugimoto, N. Haberman, G. Bobojević, C. Hauer, M. Hentze, B. Zupan, and J. Ule. *Not published at the time of writing of this thesis.* URL `https://github.com/tomazc/iCount`.

[37] L. Dagum and R. Menon. OpenMP: an industry standard API for shared-memory programming. *IEEE computational science and engineering*, 5(1):46–55, 1998.

[38] M. L. Delignette-Muller and C. Dutang. fitdistrplus : An R Package for Fitting Distributions. *Journal of Statistical Software*, 64(4), 2015. ISSN 1548-7660. doi: 10.18637/jss.v064.i04.

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

[40] D. Dominguez, P. Freese, M. S. Alexis, A. Su, M. Hochman, T. Palden, C. Bazile, N. J. Lambert, E. L. Van Nostrand, G. A. Pratt, G. W. Yeo, B. Graveley, and C. B. Burge. Sequence, Structure and Context Preferences of Human RNA Binding Proteins. *Doi.Org*, 2017. doi: 10.1101/201996.

[41] P. Drewe-Boss, H.-H. Wessels, and U. Ohler. omniCLIP: probabilistic identification of protein-RNA interactions from CLIP-seq data. *Genome biology*, 19(1):183, 2018.

[42] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis.* Cambridge University Press, Cambridge, 1998. ISBN 9780511790492. doi: 10.1017/CBO9780511790492.

[43] E.˜Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. ISSN 00034851.

[44] J. Ernst, P. Kheradpour, T. S. Mikkelsen, N. Shoresh, L. D. Ward, C. B. Epstein, X. Zhang, L. Wang, R. Issner, M. Coyne, M. Ku, T. Durham, M. Kellis, and B. E. Bernstein. Mapping and analysis of chromatin state dynamics in nine human cell types. *Nature*, 2011. ISSN 00280836. doi: 10.1038/nature09906.

[45] M. Esteller. Non-coding RNAs in human disease. *Nature Reviews Genetics*, 12 (12):861–874, dec 2011. ISSN 1471-0056. doi: 10.1038/nrg3074.

[46] J. Fallmann, S. Will, J. Engelhardt, B. Grüning, R. Backofen, and P. F. Stadler. Recent advances in RNA folding. *Journal of Biotechnology*, 261:97–104, nov 2017. ISSN 01681656. doi: 10.1016/j.jbiotec.2017.07.007.

[47] R. A. Flynn, L. Martin, R. C. Spitale, B. T. Do, S. M. Sagan, B. Zarnegar, K. Qu, P. A. Khavari, S. R. Quake, P. Sarnow, and H. Y. Chang. Dissecting noncoding and pathogen RNA-protein interactomes. *RNA*, 2015. ISSN 14699001. doi: 10.1261/rna.047803.114.

[48] M. B. Friedersdorf and J. D. Keene. Advancing the functional utility of PAR-CLIP by quantifying background binding to mRNAs and lncRNAs. *Genome biology*, 15(1):1–16, 2014.

[49] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, and R. Ulerich. *GNU Scientific Library Reference Manual.* Network Theory Ltd., United Kingdom, 3rd edition, 2009. ISBN 0954612078.

[50] S. Geisler and J. Coller. RNA in unexpected places: long non-coding RNA functions in diverse cellular contexts. *Nature Reviews Molecular Cell Biology*, 14 (11):699–712, nov 2013. ISSN 1471-0072. doi: 10.1038/nrm3679.

[51] P. L. Germain, E. Ratti, and F. Boem. Junk or functional DNA? ENCODE and the function controversy. *Biology and Philosophy*, 2014. ISSN 15728404. doi: 10.1007/s10539-014-9441-3.

[52] S. Gerstberger, M. Hafner, and T. Tuschl. A census of human RNA-binding proteins. *Nature Reviews Genetics*, 2014. ISSN 14710064. doi: 10.1038/nrg3813.

[53] Z. GHAHRAMANI. AN INTRODUCTION TO HIDDEN MARKOV MODELS AND BAYESIAN NETWORKS. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9–42, feb 2001. ISSN 0218-0014. doi: 10.1142/S0218001401000836.

[54] G. Giudice, F. Sánchez-Cabo, C. Torroja, and E. Lara-Pezzi. ATtRACT—a database of RNA-binding proteins and associated motifs. *Database*, 2016:baw035, apr 2016. ISSN 1758-0463. doi: 10.1093/database/baw035.

[55] C. E. Grant, T. L. Bailey, and W. S. Noble. FIMO: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

[56] D. Graur. An upper limit on the functional fraction of the human genome. *Genome Biology and Evolution*, 2017. ISSN 17596653. doi: 10.1093/gbe/evx121.

[57] D. Graur, Y. Zheng, N. Price, R. B. Azevedo, R. A. Zufall, and E. Elhaik. On the immortality of television sets: "Function" in the human genome according to the evolution-free gospel of encode. *Genome Biology and Evolution*, 2013. ISSN 17596653. doi: 10.1093/gbe/evt028.

[58] T. L. Griffiths, S. Goldwater, S. Goldwater, and T. Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. *ACL'07 - 45th Annual Meeting of the Association of Computational Linguistics*, 2007. ISSN 0736587X.

[59] B. Grüning, R. Dale, A. Sjödin, B. A. Chapman, J. Rowe, C. H. Tomkins-Tinch, R. Valieris, and J. Köster. Bioconda: Sustainable and comprehensive software distribution for the life sciences. *Nature Methods*, 2018. ISSN 15487105. doi: 10.1038/s41592-018-0046-7.

[60] N. Haberman. Insights into protein-RNA complexes from computational analyses of iCLIP experiments. *University College London*, (Doctoral thesis), 2017.

[61] N. Haberman, I. Huppertz, J. Attig, J. König, Z. Wang, C. Hauer, M. W. Hentze, A. E. Kulozik, H. Le Hir, T. Curk, and Others. Insights into the design and interpretation of iCLIP experiments. *Genome Biology*, 18(1):7, 2017.

[62] M. Hafner, M. Landthaler, L. Burger, M. Khorshid, J. Hausser, P. Berninger, A. Rothballer, M. Ascano, A.-C. Jungkamp, M. Munschauer, and Others. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, 141(1):129–141, 2010.

[63] A. Han, P. Stoilov, A. J. Linares, Y. Zhou, X. D. Fu, and D. L. Black. De Novo Prediction of PTBP1 Binding and Splicing Targets Reveals Unexpected Features of Its RNA Recognition and Function. *PLoS Computational Biology*, 2014. ISSN 1553734X. doi: 10.1371/journal.pcbi.1003442.

[64] J. M. Heather and B. Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1–8, jan 2016. ISSN 08887543. doi: 10.1016/j.ygeno.2015.11.003.

[65] D. Heller, R. Krestel, U. Ohler, M. Vingron, and A. Marsico. ssHMM: extracting intuitive sequence-structure motifs from high-throughput RNA-binding protein

data. *Nucleic Acids Research*, 45(19):11004–11018, nov 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx756.

[66] M. W. Hentze, A. Castello, T. Schwarzl, and T. Preiss. A brave new world of RNA-binding proteins. *Nature Reviews Molecular Cell Biology*, 19(5):327–341, jan 2018. ISSN 1471-0072. doi: 10.1038/nrm.2017.130.

[67] E. Holmqvist, P. R. Wright, L. Li, T. Bischler, L. Barquist, R. Reinhardt, R. Backofen, and J. Vogel. Global RNA recognition patterns of post-transcriptional regulators Hfq and CsrA revealed by UV crosslinking *in vivo*. *The EMBO Journal*, 2016. ISSN 0261-4189. doi: 10.15252/embj.201593360.

[68] I. Huppertz, J. Attig, A. D'Ambrogio, L. E. Easton, C. R. Sibley, Y. Sugimoto, M. Tajnik, J. König, and J. Ule. iCLIP: Protein-RNA interactions at nucleotide resolution. *Methods*, 2014. ISSN 10462023. doi: 10.1016/j.ymeth.2013.10.011.

[69] S. Hussain, A. A. Sajini, S. Blanco, S. Dietmann, P. Lombard, Y. Sugimoto, M. Paramor, J. G. Gleeson, D. T. Odom, J. Ule, and M. Frye. NSun2-mediated cytosine-5 methylation of vault noncoding RNA determines its processing into regulatory small RNAs. *Cell Reports*, 2013. ISSN 22111247. doi: 10.1016/j.celrep.2013.06.029.

[70] W. James Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler. The human genome browser at UCSC. *Genome Research*, 2002. ISSN 10889051. doi: 10.1101/gr.229102.

[71] R. E. Jewett. Characteristics of physician's assistant programs. *Journal of medical education*, 50(12 pt 2):92–6, dec 1975. ISSN 0022-2577. doi: 10.1023/A:1007425814087.

[72] D. S. Johnson, A. Mortazavi, R. M. Myers, and B. Wold. Genome-Wide Mapping of in Vivo Protein-DNA Interactions. *Science*, 316(5830):1497–1502, jun 2007. ISSN 0036-8075. doi: 10.1126/science.1141319.

[73] M. I. Jordan, Z. Ghahramani, and L. K. Saul. Hidden Markov Decision Trees. *Advances in Neural Information Processing Systems*, 1997. doi: 10.1007/978-94-011-5014-9_5.

[74] G. Jordison. Molecular Biology of the Gene, 1965. ISSN 03024598.

[75] W. Kassuhn, U. Ohler, and P. Drewe. Cseq-simulator: A data simulator for CLIP-Seq eperiments. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 21:433–444, 2016.

[76] D. V. Kliche, P. L. Smith, and R. W. Johnson. L-moment estimators as applied to gamma drop size distributions. *Journal of Applied Meteorology and Climatology*, 2008. ISSN 15588424. doi: 10.1175/2008JAMC1936.1.

[77] T. Komatsu, S. Yokoi, K. Fujii, M. Mito, Y. Kimura, S. Iwasaki, and S. Nakagawa. UPA-Seq: Prediction of functional lncRNAs using differential sensitivity to UV crosslinking. *RNA*, 2018. ISSN 14699001. doi: 10.1261/rna.067611.118.

[78] J. König, K. Zarnack, G. Rot, T. Curk, M. Kayikci, B. Zupan, D. J. Turner, N. M. Luscombe, and J. Ule. iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. *Nature structural & molecular biology*, 17(7): 909–915, 2010.

[79] S. Krakau, H. Richard, and A. Marsico. PureCLIP: Capturing target-specific protein-RNA interaction footprints from single-nucleotide CLIP-seq data. *Genome Biology*, 2017. ISSN 1474760X. doi: 10.1186/s13059-017-1364-2.

[80] F. Krueger. Trim Galore! v0.4.0., 2015. URL `https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/`.

[81] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, 9(1):112–147, jan 1998. ISSN 1052-6234. doi: 10.1137/S1052623496303470.

[82] N. Lambert, A. Robertson, M. Jangi, S. McGeary, P. A. Sharp, and C. B. Burge. RNA Bind-n-Seq: Quantitative Assessment of the Sequence and Structural Binding Specificity of RNA Binding Proteins. *Molecular Cell*, 2014. ISSN 10974164. doi: 10.1016/j.molcel.2014.04.016.

[83] F. C. Lee and J. Ule. Advances in CLIP Technologies for Studies of Protein-RNA Interactions. *Molecular Cell*, 69(3):354–369, feb 2018. ISSN 10972765. doi: 10.1016/j.molcel.2018.01.005.

[84] M. R. Lerner and J. A. Steitz. Antibodies to small nuclear RNAs complexed with proteins are produced by patients with systemic lupus erythematosus. *Proceedings of the National Academy of Sciences*, 76(11):5495–5499, nov 1979. ISSN 0027-8424. doi: 10.1073/pnas.76.11.5495.

[85] D. D. Licatalosi, A. Mele, J. J. Fak, J. Ule, M. Kayikci, S. W. Chi, T. A. Clark, A. C. Schweitzer, J. E. Blume, X. Wang, J. C. Darnell, and R. B. Darnell. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature*, 456 (7221):464–469, nov 2008. ISSN 0028-0836. doi: 10.1038/nature07488.

[86] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data.* John Wiley & Sons, 1987. ISBN 0471802549.

[87] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 2011. ISSN 17487188. doi: 10.1186/1748-7188-6-26.

[88] M. T. Lovci, D. Ghanem, H. Marr, J. Arnold, S. Gee, M. Parra, T. Y. Liang, T. J. Stark, L. T. Gehman, S. Hoon, and Others. Rbfox proteins regulate alternative mRNA splicing through evolutionarily conserved RNA bridges. *Nature structural & molecular biology*, 20(12):1434–1442, 2013.

[89] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 2014. ISSN 1474760X. doi: 10.1186/s13059-014-0550-8.

[90] G. Lu and T. M. Hall. Alternate modes of cognate RNA recognition by human PUMILIO proteins. *Structure*, 2011. ISSN 09692126. doi: 10.1016/j.str.2010.12.019.

[91] J. B. Lucks, S. A. Mortimer, C. Trapnell, S. Luo, S. Aviran, G. P. Schroth, L. Pachter, J. A. Doudna, and A. P. Arkin. Multiplexed RNA structure characterization with selective 2'-hydroxyl acylation analyzed by primer extension sequencing (SHAPE-Seq). *Proceedings of the National Academy of Sciences*, 108(27):11063–11068, jul 2011. ISSN 0027-8424. doi: 10.1073/pnas.1106501108.

[92] A. Lukashin. GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Research*, 26(4):1107–1115, feb 1998. ISSN 13624962. doi: 10.1093/nar/26.4.1107.

[93] K. E. Lukong, K.-w. Chang, E. W. Khandjian, and S. Richard. RNA-binding proteins in human genetic disease. *Trends in Genetics*, 24(8):416–425, aug 2008. ISSN 01689525. doi: 10.1016/j.tig.2008.05.004.

[94] B. M. Lunde, C. Moore, and G. Varani. RNA-binding proteins: modular design for efficient function. *Nature Reviews Molecular Cell Biology*, 8(6):479–490, jun 2007. ISSN 1471-0072. doi: 10.1038/nrm2178.

[95] H. Mamitsuka. Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models. *Proteins*, 1998. ISSN 0887-3585. doi: 10.1002/(SICI)1097-0134(19981201)33:4<460::AID-PROT2>3.0.CO;2-M.

[96] A. Mammana and H. R. Chung. Chromatin segmentation based on a probabilistic model for read counts explains a large portion of the epigenome. *Genome Biology*, 2015. ISSN 1474760X. doi: 10.1186/s13059-015-0708-z.

[97] A. Marsico, M. R. Huska, J. Lasserre, H. Hu, D. Vucicevic, A. Musahl, U. A. Orom, and M. Vingron. PROmiRNA: A new miRNA promoter recognition method uncovers the complex regulation of intronic miRNAs. *Genome Biology*, 2013. ISSN 1474760X. doi: 10.1186/gb-2013-14-8-r84.

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

[99] D. Maticzka, S. J. Lange, F. Costa, and R. Backofen. GraphProt: Modeling binding preferences of RNA-binding proteins. *Genome Biology*, 2014. ISSN 1474760X. doi: 10.1186/gb-2014-15-1-r17.

[100] K. I. M. McKinnon. Convergence of the Nelder–Mead Simplex Method to a Nonstationary Point. *SIAM Journal on Optimization*, 9(1):148–158, jan 1998. ISSN 1052-6234. doi: 10.1137/S1052623496303482.

[101] T. R. Mercer and J. S. Mattick. Structure and function of long noncoding RNAs in epigenetic regulation, 2013. ISSN 15459993.

[102] M. J. Moore, C. Zhang, E. C. Gantman, A. Mele, J. C. Darnell, and R. B. Darnell. Mapping Argonaute and conventional RNA-binding protein interactions with RNA at single-nucleotide resolution using HITS-CLIP and CIMS analysis. *Nature protocols*, 9(2):263–293, 2014.

[103] K. P. Murphy. *Machine Learning: A Probablistic Perspective.* MIT Press, 2012. ISBN 978-0262018029.

[104] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, jan 1965. ISSN 0010-4620. doi: 10.1093/comjnl/7.4.308.

[105] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Statistical Models in S*, 1972. ISSN 19395108. doi: 10.1201/9780203738535.

[106] S. Niranjanakumari. Reversible cross-linking combined with immunoprecipitation to study RNA–protein interactions in vivo. *Methods*, 26(2):182–190, feb 2002. ISSN 10462023. doi: 10.1016/S1046-2023(02)00021-X.

[107] J. Nocedal, S. J. Wright, and SpringerLink (Online service). *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN 978-0-387-30303-1. doi: 10.1007/978-0-387-40065-5.

[108] J. H. Noh, K. M. Kim, W. G. McClusky, K. Abdelmohsen, and M. Gorospe. Cytoplasmic functions of long noncoding RNAs. *Wiley Interdisciplinary Reviews: RNA*, 9(3):e1471, may 2018. ISSN 17577004. doi: 10.1002/wrna.1471.

[109] E. L. V. Nostrand, P. Freese, G. A. Pratt, X. Wang, X. Wei, S. M. Blue, D. Dominguez, N. A. L. Cody, S. Olson, B. Sundararaman, R. Xiao, L. Zhan, C. Bazile, L. P. B. Bouvrette, J. Chen, M. O. Duff, K. Garcia, C. Gelboin-Burkhart, A. Hochman, N. J. Lambert, H. Li, T. B. Nguyen, T. Palden, I. Rabano, S. Sathe, R. Stanton, A. L. Louie, S. Aigner, J. Bergalet, B. Zhou, A. Su, R. Wang, B. A. Yee, X.-D. Fu, E. Lecuyer, C. B. Burge, B. Graveley, and G. W. Yeo. A Large-Scale Binding and Functional Map of Human RNA Binding Proteins. *bioRxiv*, 2017. doi: 10.1101/179648.

[110] A. F. Palazzo and E. S. Lee. Non-coding RNA: what is functional and what is junk? *Frontiers in Genetics*, 6, jan 2015. ISSN 1664-8021. doi: 10.3389/fgene.2015.00002.

[111] M. Pertea, A. Shumate, G. Pertea, A. Varabyou, Y.-C. Chang, A. K. Madugundu, A. Pandey, and S. Salzberg. Thousands of large-scale RNA sequencing experiments yield a comprehensive new human gene list and reveal extensive transcriptional noise. *bioRxiv*, 2018. doi: 10.1101/332825.

[112] C. Plaschka, P.-C. Lin, and K. Nagai. Structure of a pre-catalytic spliceosome. *Nature*, may 2017. ISSN 0028-0836. doi: 10.1038/nature22799.

[113] X. C. Quek, D. W. Thomson, J. L. Maag, N. Bartonicek, B. Signal, M. B. Clark, B. S. Gloss, and M. E. Dinger. lncRNAdb v2.0: expanding the reference database for functional long noncoding RNAs. *Nucleic Acids Research*, 43(D1):D168–D173, jan 2015. ISSN 1362-4962. doi: 10.1093/nar/gku988.

[114] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[115] F. A. Ran, P. D. Hsu, J. Wright, V. Agarwala, D. A. Scott, and F. Zhang. Genome engineering using the CRISPR-Cas9 system. *Nature Protocols*, 8(11):2281–2308, oct 2013. ISSN 1754-2189. doi: 10.1038/nprot.2013.143.

[116] D. Ray, H. Kazan, E. T. Chan, L. P. Castillo, S. Chaudhry, S. Talukder, B. J. Blencowe, Q. Morris, and T. R. Hughes. Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nature Biotechnology*, 2009. ISSN 10870156. doi: 10.1038/nbt.1550.

[117] K. Reinert, T. H. Dadi, M. Ehrhardt, H. Hauswedell, S. Mehringer, R. Rahn, J. Kim, C. Pockrandt, J. Winkler, E. Siragusa, G. Urgese, and D. Weese. The SeqAn C++ template library for efficient sequence analysis: A resource for programmers. *Journal of Biotechnology*, 2017. ISSN 18734863. doi: 10.1016/j.jbiotec.2017.07.017.

[118] J. S. Reuter and D. H. Mathews. RNAstructure: Software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, 2010. ISSN 14712105. doi: 10.1186/1471-2105-11-129.

[119] P. H. Reyes-Herrera, C. A. Speck-Hernandez, C. A. Sierra, and S. Herrera. BackCLIP: A tool to identify common background presence in PAR-CLIP datasets. *Bioinformatics*, 31(22):3703–3705, 2015. ISSN 14602059. doi: 10.1093/bioinformatics/btv442.

[120] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth. Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 2015. ISSN 13624962. doi: 10.1093/nar/gkv007.

[121] J. R. Sanford, P. Coutinho, J. A. Hackett, X. Wang, W. Ranahan, and J. F. Caceres. Identification of Nuclear and Cytoplasmic mRNA Targets for the Shuttling Protein SF2/ASF. *PLoS ONE*, 3(10):e3369, oct 2008. ISSN 1932-6203. doi: 10.1371/journal.pone.0003369.

[122] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, dec 1977. ISSN 0027-8424. doi: 10.1073/pnas.74.12.5463.

[123] B. Schäling. *The boost C++ libraries.* XML Press, Laguna Hills, CA, 2nd edition, 2014. ISBN 1937434362.

[124] A. Shah, Y. Qian, S. M. Weyn-Vanhentenryck, and C. Zhang. CLIP Tool Kit (CTK): a flexible and robust pipeline to analyze CLIP sequencing data. *Bioinformatics*, page btw653, 2016.

[125] M. D. SHETLAR, J. CARBONE, E. STEADY, and K. HOM. PHOTOCHEMICAL ADDITION OF AMINO ACIDS AND PEPTIDES TO POLYURIDYLIC ACID. *Photochemistry and Photobiology*, 39(2):141–144, feb 1984. ISSN 0031-8655. doi: 10.1111/j.1751-1097.1984.tb03419.x.

[126] C. R. Sibley. *Individual Nucleotide Resolution UV Cross-Linking and Immunoprecipitation (iCLIP) to Determine Protein–RNA Interactions*, pages 427–454. Springer New York, New York, NY, 2018. ISBN 978-1-4939-7213-5. doi: 10.1007/978-1-4939-7213-5_29.

[127] M. D. Simon. Capture hybridization analysis of RNA targets (CHART). *Current Protocols in Molecular Biology*, 101:21.25.1–21.25.16, 2013. doi: 10.1002/0471142727.mb2125s101.

[128] G. Singh, E. P. Ricci, and M. J. Moore. RIPiT-Seq: A high-throughput approach for footprinting RNA:protein complexes. *Methods*, 65(3):320–332, feb 2014. ISSN 10462023. doi: 10.1016/j.ymeth.2013.09.013.

[129] C. A. Sloan, E. T. Chan, J. M. Davidson, V. S. Malladi, J. S. Strattan, B. C. Hitz, I. Gabdank, A. K. Narayanan, M. Ho, B. T. Lee, L. D. Rowe, T. R. Dreszer, G. Roe, N. R. Podduturi, F. Tanaka, E. L. Hong, and J. M. Cherry. ENCODE data at the ENCODE portal. *Nucleic Acids Research*, 44(D1):D726–D732, jan 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv1160.

[130] T. Smith, A. Heger, and I. Sudbery. UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. *Genome Research*, 27(3):491–499, mar 2017. ISSN 1088-9051. doi: 10.1101/gr.209601.116.

[131] G. K. Smyth. Limma: linear models for microarray data BT - Bioinformatics and Computational Biology Solutions Using R and Bioconductor. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, 2005. ISSN 1544-6115. doi: citeulike-article-id:5722720.

[132] L. Song and G. E. Crawford. DNase-seq: A High-Resolution Technique for Mapping Active Gene Regulatory Elements across the Genome from Mammalian Cells. *Cold Spring Harbor Protocols*, 2010(2):pdb.prot5384–pdb.prot5384, feb 2010. ISSN 1559-6095. doi: 10.1101/pdb.prot5384.

[133] Y. Sugimoto, J. König, S. Hussain, B. Zupan, T. Curk, M. Frye, and J. Ule. Analysis of CLIP and iCLIP methods for nucleotide-resolution studies of protein-RNA interactions. *Genome biology*, 13(8):1–13, 2012.

[134] F. R. Sutandy, S. Ebersberger, L. Huang, A. Busch, M. Bach, H.-S. Kang, J. Fallmann, D. Maticzka, R. Backofen, P. F. Stadler, K. Zarnack, M. Sattler, S. Legewie, and J. König. In vitro iCLIP-based modeling uncovers how the splicing factor U2AF2 relies on regulation by cofactors. *Genome Research*, 28(5):699–713, may 2018. ISSN 1088-9051. doi: 10.1101/gr.229757.117.

[135] S. Tenenbaum. Ribonomics: identifying mRNA subsets in mRNP complexes using antibodies to RNA-binding proteins and genomic arrays. *Methods*, 26(2): 191–198, feb 2002. ISSN 10462023. doi: 10.1016/S1046-2023(02)00022-1.

[136] S. A. Tenenbaum, C. C. Carson, P. J. Lager, and J. D. Keene. Identifying mRNA subsets in messenger ribonucleoprotein complexes by using cDNA arrays. *Proceedings of the National Academy of Sciences*, 97(26):14085–14090, dec 2000. ISSN 0027-8424. doi: 10.1073/pnas.97.26.14085.

[137] T. Treiber, N. Treiber, and G. Meister. Regulation of microRNA biogenesis and its crosstalk with other cellular pathways, 2018. ISSN 14710080.

[138] C. Tuerk and L. Gold. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science*, 1990. ISSN 00368075. doi: 10.1126/science.2200121.

[139] M. Uhl, T. Houwaart, G. Corrado, P. R. Wright, and R. Backofen. Computational analysis of CLIP-seq data. *Methods (San Diego, Calif.)*, 118-119:60–72, 2017. ISSN 1095-9130 (Electronic). doi: 10.1016/j.ymeth.2017.02.006.

[140] J. Ule. CLIP Identifies Nova-Regulated RNA Networks in the Brain. *Science*, 302 (5648):1212–1215, nov 2003. ISSN 0036-8075. doi: 10.1126/science.1090095.

[141] P. J. Uren, E. Bahrami-Samani, S. C. Burns, M. Qiao, F. V. Karginov, E. Hodges, G. J. Hannon, J. R. Sanford, L. O. F. Penalva, and A. D. Smith. Site identification in high-throughput RNA–protein interaction data. *Bioinformatics*, 28(23):3013–3020, 2012.

[142] T. Uzawa, A. Yamagishi, and T. Oshima. Polypeptide Synthesis Directed by DNA as a Messenger in Cell-Free Polypeptide Synthesis by Extreme Thermophiles, Thermus thermophilus HB27 and Sulfolobus tokodaii Strain 7. *Journal of Biochemistry*, 131(6):849–853, jun 2002. ISSN 0021-924X. doi: 10.1093/oxfordjournals.jbchem.a003174.

[143] E. L. Van Nostrand, G. A. Pratt, A. A. Shishkin, C. Gelboin-Burkhart, M. Y. Fang, B. Sundararaman, S. M. Blue, T. B. Nguyen, C. Surka, K. Elkins, R. Stanton, F. Rigo, M. Guttman, and G. W. Yeo. Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). *Nature Methods*, 13(6):508–514, jun 2016. ISSN 1548-7091. doi: 10.1038/nmeth.3810.

[144] E. L. Van Nostrand, C. Gelboin-Burkhart, R. Wang, G. A. Pratt, S. M. Blue, and G. W. Yeo. CRISPR/Cas9-mediated integration enables TAG-eCLIP of endogenously tagged RNA binding proteins. *Methods*, 118-119:50–59, apr 2017. ISSN 10462023. doi: 10.1016/j.ymeth.2016.12.007.

[145] E. L. Van Nostrand, T. B. Nguyen, C. Gelboin-Burkhart, R. Wang, S. M. Blue, G. A. Pratt, A. L. Louie, and G. W. Yeo. *Robust, Cost-Effective Profiling of RNA Binding Protein Targets with Single-end Enhanced Crosslinking and Immunoprecipitation (seCLIP)*, pages 177–200. Springer New York, New York, NY, 2017. ISBN 978-1-4939-7204-3. doi: 10.1007/978-1-4939-7204-3_14.

[146] E. L. Van Nostrand, A. A. Shishkin, G. A. Pratt, T. B. Nguyen, and G. W. Yeo. Variation in single-nucleotide sensitivity of eCLIP derived from reverse transcription conditions. *Methods*, 126:29–37, aug 2017. ISSN 10462023. doi: 10.1016/j.ymeth.2017.08.002.

[147] T. Wang, Y. Xie, and G. Xiao. dCLIP: a computational approach for comparative CLIP-seq analyses. *Genome Biol*, 15(1):R11, 2014.

[148] J. Watson and F. Crick. Molecular structure of nucleic acids. *Nature*, 1953.

[149] S. M. Weyn-Vanhentenryck, A. Mele, Q. Yan, S. Sun, N. Farny, Z. Zhang, C. Xue, M. Herre, P. A. Silver, M. Q. Zhang, and Others. HITS-CLIP and integrative modeling define the Rbfox splicing-regulatory network linked to brain development and autism. *Cell reports*, 6(6):1139–1152, 2014.

[150] E. C. Wheeler, E. L. Van Nostrand, and G. W. Yeo. Advances and challenges in the detection of transcriptome-wide protein–RNA interactions. *Wiley Interdisciplinary Reviews: RNA*, 2018. ISSN 17577012. doi: 10.1002/wrna.1436.

[151] C. Willyard. New human gene tally reignites debate. *Nature*, 558(7710):354–355, jun 2018. ISSN 0028-0836. doi: 10.1038/d41586-018-05462-w.

[152] E.-W. Yang, J. H. Bahn, E. Yun-Hua Hsiao, B. X. Tan, Y. Sun, T. Fu, B. Zhou, E. L. Van Nostrand, G. A. Pratt, P. Freese, X. Wei, G. Quinones-Valdez, A. E. Urban, B. R. Graveley, C. B. Burge, G. W. Yeo, and X. Xiao. Allele-specific binding of RNA-binding proteins reveals functional genetic variants in the RNA. *bioRxiv*, 2018. doi: 10.1101/396275.

[153] Y.-C. T. Yang, C. Di, B. Hu, M. Zhou, Y. Liu, N. Song, Y. Li, J. Umetsu, and Z. Lu. CLIPdb: a CLIP-seq database for protein-RNA interactions. *BMC Genomics*, 16 (1):51, 2015. ISSN 1471-2164. doi: 10.1186/s12864-015-1273-2.

[154] B. Yee, G. Pratt, B. Graveley, E. Van Nostrand, and G. Yeo. RBP-Maps enables robust generation of splicing regulatory maps. *RNA*, 2018. ISSN 1355-8382. doi: 10.1261/rna.069237.118.

[155] K. Zarnack, J. König, M. Tajnik, I. Martincorena, S. Eustermann, I. Stévant, A. Reyes, S. Anders, N. M. Luscombe, and J. Ule. Direct competition between hnRNP C and U2AF65 protects the transcriptome from the exonization of Alu elements. *Cell*, 152(3):453–466, 2013.

[156] B. J. Zarnegar, R. A. Flynn, Y. Shen, B. T. Do, H. Y. Chang, and P. A. Khavari. irCLIP platform for efficient characterization of protein–RNA interactions. *Nature Methods*, 2016. ISSN 1548-7091. doi: 10.1038/nmeth.3840.

[157] C. Zhang and R. B. Darnell. Mapping in vivo protein-RNA interactions at single-nucleotide resolution from HITS-CLIP data. *Nature biotechnology*, 29(7):607–614, 2011.

[158] Z. Zhang and Y. Xing. CLIP-seq analysis of multi-mapped reads discovers novel functional RNA regulatory sites in the human transcriptome. *Nucleic Acids Research*, 45(16):9260–9271, sep 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx646.

[159] J. Zhao, T. K. Ohsumi, J. T. Kung, Y. Ogawa, D. J. Grau, K. Sarma, J. J. Song, R. E. Kingston, M. Borowsky, and J. T. Lee. Genome-wide Identification of Polycomb-Associated RNAs by RIP-seq. *Molecular Cell*, 40(6):939–953, dec 2010. ISSN 10972765. doi: 10.1016/j.molcel.2010.12.011.

[160] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1): 133–148, 1981. ISSN 0305-1048. doi: 10.1093/nar/9.1.133.

List of Figures

List of Tables

Zusammenfassung

Interaktionen zwischen Proteinen und RNAs spielen eine wichtige Rolle in allen posttranskriptionalen regulatorischen Prozessen. Die in den letzten Jahren entwickelten CLIP-seq Technologien haben die Hochdurchsatz-Detektion von Protein-RNA Interaktionen möglich gemacht. Anreicherungen alignierter Reads sowie Transitionen oder Deletionen einzelner Basen können dabei genutzt werden, um auf die Binderegionen rückzuschließen. Durch die Erfassung des hohen Anteils von cDNAs, die an der Protein-RNA Crosslink-Stelle trunkiert wurden, kann des Weiteren eine Auflösung bis hin zu einzelnen Nukleotiden erreicht werden.

Die steigende Anzahl publizierter Datensätze sowie Weiterentwicklungen der Verfahren erfordern maßgeschneiderte, computergestützte Analysemethoden. Existierende Methoden sind bislang nicht in der Lage, die Besonderheiten der cDNA-Trunkierungsmuster und mögliche Biase durch unspezifische Hintergrund-Binde-Ereignisse oder Crosslink-Sequenz-Präferenzen zu modellieren.

In dieser Arbeit stellen wir PureCLIP vor, eine neue Methode basierend auf einem Hidden Markov Modell, welche simultan die Detektion von Peaks und individuellen Crosslink-Positionen durchführt. Zusätzlich können externe Daten zur Korrektur unspezifischer Hintergrundsignale und des Crosslink Bias integriert werden. Um die Methode zu evaluieren haben wir drei Strategien entworfen. Zunächst haben wir einen Workflow für die Simulation von iCLIP Daten entwickelt, welcher, ausgehend von echten RNA-seq Daten und bekannten Binderegionen, die experimentellen Schritte des iCLIP Protokolls einschließlich der Generierung von Hintergrundsignalen imitiert. Als zweites haben wir experimentelle iCLIP und eCLIP Datensätze von Proteinen verwendet, deren prädominante Binderegionen bekannt sind. Schließlich haben wir als drittes die Übereinstimung von detektierten Bindestellen zwischen Replikaten zur Evaluation verwendet, unter der Annahme, dass Protein-spezifische Signale zwischen den Replikaten reproduzierbar sind.

Sowohl auf simulierten als auch auf experimentellen Daten zeigt sich, dass PureCLIP präziser in der Detektion von Crosslink-Positionen ist als andere Methoden. Insbesondere durch die Integration von Input Kontrolldaten und Crosslink assoziierten Motiven ist PureCLIP bis zu 13% präziser als andere Methoden und erreicht eine um bis zu 20% höhere Übereinstimmung zwischen Replikaten. Unsere Methode kann außerdem detektierte Crosslink-Positionen auf Basis ihrer Distanz zu Binderegionen zusammenfassen. Wir zeigen auch hier, dass die resultierenden Regionen bekannte Binderegionen mit einer hohen Präzision wiedergeben.

Darüber hinaus demonstrieren wir, dass unsere Methode für zahlreiche unterschiedliche Konfigurationen und auch für Proteine mit unterschiedlichen Bindeeigenschaften eine hohe Präzision erreicht. Als Letztes haben wir die Methode dahingehend erweitert, dass mehrere Replikate gleichzeitig integriert werden können und zeigen, dass dadurch die Präzision weiter gesteigert werden kann. PureCLIP und die zugehörige Dokumentation sind öffentlich verfügbar unter `https://github.com/skrakau/PureCLIP`.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe. Ich erkläre weiterhin, dass ich die vorliegende Arbeit oder deren Inhalt nicht in einem früheren Promotionsverfahren eingereicht habe.

Berlin, den 15. Januar 2019
Sabrina Krakau