CSEQ-SIMULATOR: A DATA SIMULATOR FOR CLIP-SEQ EXPERIMENTS

WANJA KASSUHN
Max Delbrück Center for Molecular Medicine, Berlin Institute for Medical Systems Biology
13125 Berlin, Germany
Email: wanja.kassuhn@mdc-berlin.de

UWE OHLER*
Max Delbrück Center for Molecular Medicine, Berlin Institute for Medical Systems Biology
13125 Berlin, Germany
Email: uwe.ohler@mdc-berlin.de

PHILIPP DREWE*
Max Delbrück Center for Molecular Medicine, Berlin Institute for Medical Systems Biology
13125 Berlin, Germany
Email: philipp.drewe@mdc-berlin.de

CLIP-Seq protocols such as PAR-CLIP, HITS-CLIP or iCLIP allow a genome-wide analysis of protein-RNA interactions. For the processing of the resulting short read data, various tools are utilized. Some of these tools were specifically developed for CLIP-Seq data, whereas others were designed for the analysis of RNA-Seq data. To this date, however, it has not been assessed which of the available tools are most appropriate for the analysis of CLIP-Seq data. This is because an experimental gold standard dataset on which methods can be accessed and compared, is still not available. To address this lack of a gold-standard dataset, we here present Cseq-Simulator, a simulator for PAR-CLIP, HITS-CLIP and iCLIP-data. This simulator can be applied to generate realistic datasets that can serve as surrogates for experimental gold standard dataset. In this work, we also show how Cseq-Simulator can be used to perform a comparison of steps of typical CLIP-Seq analysis pipelines, such as the read alignment or the peak calling. These comparisons show which tools are useful in different settings and also allow identifying pitfalls in the data analysis.

1. Introduction

RNA-binding proteins (RBPs) play a central role in post-transcriptional gene regulation (e.g. in splicing, RNA-degradation or translation). However, the mechanisms by which RBPs regulate RNA-processing are still poorly understood. This is partially due to the challenges in quantifying protein-RNA interactions. Therefore, the recent introduction of cross-linking immunoprecipitation-high-throughput sequencing (CLIP-Seq) protocols that allow measuring protein-RNA interactions at a nucleotide level, such as PAR-CLIP [1], HITS-CLIP [2] or iCLIP [3], present a great advance as they allow getting an accurate picture of the RBP binding-landscape.

The approach of the CLIP-protocols is, to first UV-crosslinking RBPs to their bound RNA [4]. Subsequently, the RNAs are fragmented and the protein-RNA complexes are immunoprecipitated, in order to extract the complexes that involve the RBP of interest. Next,

[*] To whom correspondence should be addressed.

the RBP is digested using Proteinase K, but typically leaving cross-linked amino acids at the crosslinking site. Finally, the RNA-fragments are reverse transcribed to produce a cDNA library that can, at the end, be sequenced. The amino acids that are still linked to the cross-linking sites can introduce errors during the reverse transcription in the cDNA (diagnostic events) at the cross-linking site. For PAR-CLIP, these errors are predominately Thymine to Cytosine conversions (T-C conversion), whereas short deletions are introduced in HITS-CLIP and truncations in iCLIP experiments. As the diagnostic events occur at the crosslinking site, the events can be used to infer with single nucleotide-resolution the interaction site.

After sequencing the library, the resulting reads are aligned to the genome. A difference of CLIP-Seq reads and RNA-Seq reads is, however, that CLIP-Seq reads tend to be shorter than RNA-Seq reads (typically around 25 bp) and that they additionally can contain diagnostic events. Consequently, these two differences make alignment of CLIP-Seq reads more challenging than the alignment of RNA-Seq reads. To our knowledge there exists no spliced-alignment tool that is specifically design to align this data. Therefore, various aligners for gapped or ungapped alignments such as Bowtie2 [10], BWA [11], BWA-PSSM [12], STAR [18] or TopHat2 [19] are used to map the reads. The aligned reads can then be used in order to determine the sites of protein-RNA interactions. For this, sites where the CLIP-Seq reads are enriched are identified (peak calling). More sophisticated approaches, such as PARalyzer [5] or wavClusteR [7], make additionally use of the diagnostic events in order go get more accurate predictions.

However, for the read alignment and the subsequent peak calling, a systematics evaluation of the tools to perform the analyses has not been performed yet. This is partially due to the fact that there is no dataset available for which the ground truth is known and on the basis of which the tools can be benchmarked. A potential surrogate for such a dataset could be a realistic simulated dataset. However, there exist only simulators for RNA-Seq data (e.g. Flux Simulator [8]) but to our knowledge, there does not exist a realistic simulator for CLIP-Seq protocols.

In this work, we therefore present the CLIP-sequencing Simulator (Cseq-Simulator), a software to simulate data for various CLIP-protocols to address this lack of CLIP-Seq data simulators. We show that our simulation pipeline can be used to generate CLIP-Seq datasets that have the same characteristics as real datasets. Furthermore, we exemplify how this simulator can be used to assess the performance of various alignment tools for CLIP-Seq data. Finally, we study how the choice of the alignment algorithms influences the peak calling and identify potential pitfalls in the CLIP-Seq data analysis.

2. Material and Methods

2.1. *Read simulation approach*

Simulated data can provide a useful approximation to real dataset in cases where experimental determination of the ground truth on a large scale is infeasible. However, for the simulation to be useful, it is critical that it has the same characteristics as real datasets. Otherwise, the insights gained on the simulated data may not be transferable to real data. A challenge in the data simulation is, however, that the underlying processes are often only partially understood.

Thus, assumptions on the modelled processes have to be made, which may not be valid and can result in differences between simulated and real data.

In the Cseq-Simulator, we mimic key steps of the CLIP-Seq protocols in order to simulate CLIP-Seq data that is as realistic as possible. This is done in the following manner (see Fig. 1):

First, we determine the transcriptomic RNA-binding site of the RBP of interest. To this end, we use a position weight matrix (PWM) of the RBP of interest in order to predict its binding sites using FIMO [9]. The binding sites are called on the positive strand of annotated transcripts. As an alternative to the prediction of binding sites the user may also provide a list of transcriptomic binding sites. This can be useful when the RBP has an unspecific sequence motif, binding depends not only on the sequence or a high-quality set of experimentally determined binding sites is available.

After determining the binding sites, we simulated the raw reads (i.e. the reads without the diagnostic events). The PAR-CLIP, iCLIP and HITS-CLIP protocol share many steps with the standard RNA-Seq protocol. Therefore, we use components of the Flux Simulator [8], a simulator that has been shown to generate realistic RNA-Seq data, for simulation of steps of CLIP-Seq protocols that are similar to the RNA-Seq protocol. Specifically, we use the Flux-Simulator to first simulate the transcript abundances if the abundances are not provided by the user. As the transcripts that are not bound by the RBP are not of interest for the simulation, we set their expression to zero and readjust the other transcripts in order to speed up the simulation. This is done such that the overall number of transcripts remains constant.

Next, we use Flux Simulator to generate a library based on the transcript abundances. Then, we remove all the fragments in the library that do not contain a RBP binding site. This yields a library of RNA-fragments that have a RBP-binding site. Subsequently, we use the Flux Simulator to simulate the library amplification and sequencing of the library. This results in the raw reads. The advantage of using Flux simulator is that effects such as PCR-duplicates and sequencing errors can be simulated.

Finally, in order to generate the CLIP-Seq reads, we induce the diagnostic events (e.g. T-C conversions, deletions and truncations) in the raw reads. To this end, we sample the diagnostic events in the reads according to user specified distribution (diagnostic event profile) that is centred on the binding site. The resulting CLIP-Seq reads are returned in the FASTA-format.

2.2. *Dataset generation*

For the CLIP-Seq reads generation, we used our pipeline (see Sec. 2.1). We simulated reads for the GRCh38 *human* genome using the GENCODE release 21 gene annotation. To call Pumilio homolog 2 (PUM2) binding sites, we used the PWM that we obtained from [5]. For the read simulation we used the T-C conversion event profile of PUM2 from [5]. For the simulation of deletions, we assumed a uniform diagnostic event profile at all locations in the read that were a Thymine. To simulate truncations, we assumed that they occur at random at distance. However, we only introduced the truncation, if the location that was sampled had a distance of at least 4 bp the binding site.

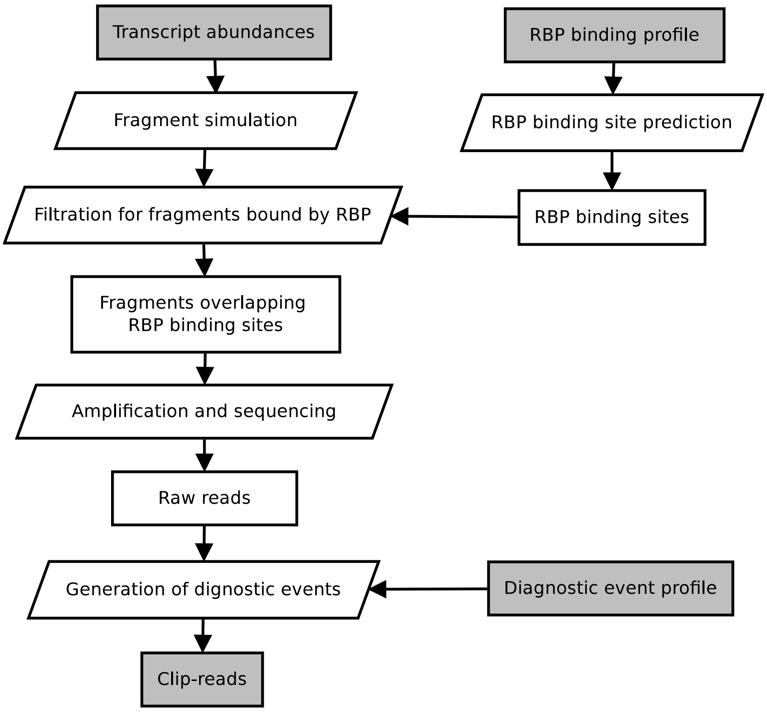
Transcript abundances
Fragment simulation
Filtration for fragments bound by RBP
Fragments overlapping RBP binding sites
Amplification and sequencing
Raw reads
Generation of dignostic events
Clip-reads
RBP binding profile
RBP binding site prediction
RBP binding sites
Diagnostic event profile

Fig. 1. Shown is a flow chart of the Cseq-Simulator read simulation. Shown in dark grey are the input and output data.

2.3. *Alignment algorithms*

In this study, we used the following aligners to align the CLIP-Seq reads to the hg38 *human* genome: Bowtie2 [10], BWA [11], BWA-PSSM [12], HISAT [15], PalMapper [16], Segemehl [17], STAR [18] and TopHat2 [19]. The tools have been selected to cover the commonly used short read alignment tools for CLIP-Seq data and aligners that are well suited for CLIP-Seq data alignment. For all the tools, we allowed in general for two mismatches and one indel during the alignment. To have a comparison that is less affected by the appropriateness of default parameters for CLIP-Seq data, we contacted the authors to obtain optimised parameter settings. For the tools where the authors did not reply, we used the parameters that were recommended by experts working with PAR-CLIP. For BWA-PSSM we used the error-profile that was provided for PAR-CLIP-Seq data. A list of the non-default parameters is given below:

Bowtie2: `-f -p 1 -L 15 -N 1 --very-sensitive --end-to-end`
BWA: `-k 1 -n 3 -t 1`
BWA-PSSM: `-l 15 -m 400`
HISAT: `-f -p 1 --mp 3,1 --pen-cansplice 0 --known-splicesites-infile splicesites.txt --max-intronlen 10000`
PalMapper: `-M 2 -n 3 -l 10 -E 3 -m 3 -S -min-spliced-segment-len 6 -include-unmapped-reads -report-gff-init annotation.gff3 -qpalma parameter.qpalma -I 10000 -no-ss-pred`
Segemehl: `-S -D 2 -M 3 -Z`

STAR:

```
--alignIntronMax 1 --sjdbGTFfile annotation.gtf --outSAMunmapped
"Within" --outFilterMultimapNmax 3 --outFilterMismatchNmax 2
--seedSearchStartLmax 6 --winAnchorMultimapNmax 10000 --alignEndsType
EndToEnd
```

TopHat2:

```
--report-secondary-alignments --read-mismatches 2 --read-edit-dist
3 --min-anchor-length 10 --splice-mismatches 1 --max-intron-length
10000 --no-coverage-search --segment-mismatches 1 --max-multihits 3
--segment-length 10 --no-convert-bam
```

2.4. *Alignment evaluation*

To evaluate the alignment of a set of reads, we determined for each read whether the read was mapping to multiple locations (multimapping) or to only one position. If the latter was the case we further determined whether the alignment was correct (i.e. it was mapped to the read‘s origin) or whether its mapping location was incorrect (mismapped).

2.5. *Alignment filtering*

For the filtering of multimappers, we only kept the best alignment for a read when the second best alignment had more than one mismatch more than the best alignment. Otherwise, all alignments for the respective read are discarded. In the later case, the read was treated as an unaligned read in the alignment evaluation. If read aligned only once, we kept it.

2.6. *Peak caller*

To call peaks from the CLIP-Seq reads, we used three tools: wavClusteR [7], PARalyzer [5] and BMix [**?**]. The tool Piranha [6] was not included in our evaluation as it could not be applied to all alignments. As the peak calling tools had different requirements to the input SAM-format, we standardized the SAM-files such that they were accepted by all peak callers. This was done by discarding all unmapped reads and alignments with ”MD”-tags that included other operations than nucleotide substitutions. The peak calling tools were run with their default parameters. We defined the called peaks to be correct when they entirely overlapped the RBP binding sites.

3. Results

3.1. *Read generation*

We generated reads for PAR-CLIP, HITS-CLIP and iCLIP experiments of PUM2 using the Cseq-Simulator as described below. For the read simulation, we used all transcriptomic PUM2 binding-sites that were predicted by FIMO (FDR≤ 0.1). Of the 23362 detected binding sites, 5233 were in transcripts that were simulated to be expressed. To simulate the reads, we first simulated the raw reads without diagnostic events for seven different read length: 14, 16, 18, 20, 24, 28 and 32. Overall, this resulted between 0.66×10^6 and 2.74×10^6 reads per library (see Tab. 1). We used these reads as templates to simulate three different groups of reads: Reads with T-C conversions, deletions or truncations. This resulted in seven sets of reads for each type of diagnostic event. From the reads for which we simulated T-C conversions, 75% had at least one diagnostic event. For the reads with simulated deletions and truncations between 85% and 91% resp. 15% and 69% had a diagnostic event. The high variation in the fraction of reads having a truncation was due to the fact that we set a boundary around the motif where truncation sites where there were no truncations. Therefore, many short reads were not truncated.

To determine whether the simulated PUM2-dataset had a realistic diagnostic event distribution, we analysed the diagnostic event distribution for the simulated data. For this, we compared the fraction of reads that had a T-C conversion at a given position relative to the predicted binding site with the diagnostic event profile from [5], which was used for the simulation (See Fig. 2). Overall, we found that the two profiles were very similar. We noticed, however, that there were subtle differences at the positions where the motif indicated a high preference for A. We believe, that these differences are due to the fact that the binding site prediction did not predict binding sites with a T at these positions. Consequently, a T-C conversion could not be simulated.

Table 1. Library sizes for the different read lengths.

Read length (bp)	*14*	*16*	*18*	*20*	*24*	*28*	*32*
Number of reads ($\times 10^6$)	0.66	1.13	1.16	1.35	1.92	2.22	2.74

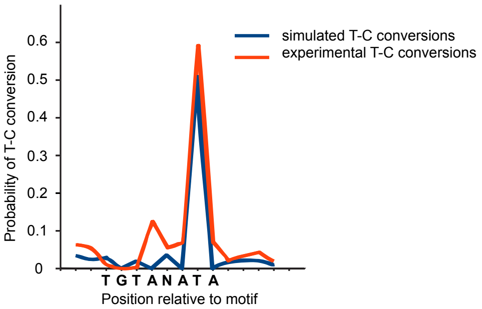

Fig. 2. Shown is T-C diagnostic event profile that was used for the simulation (red) and the fraction of reads that have a T-C conversion at a given position (blue) relative to the motif (bold letters).

3.2. *Assessment of alignment tools for CLIP-Seq data*

Short read alignment tools are used in most bioinformatics pipelines for the analysis of CLIP-Seq data. However, the influence of the choice of the aligner on the outcome is an aspect that has received little attention. Here, we exemplify how we can use the Cseq-Simulator to asses the performance of aligners for PAR-CLIP, HITS-CLIP and iCLIP data. For this, we aligned reads for a selection of commonly used short read aligners, namely Bowtie2, BWA, BWA-PSSM, HISAT, PalMapper, Segemehl, STAR and TopHat2. We then analysed different aspects of the alignment tools.

We first studied the sensitivity of the aligners, i.e. how many of the alignments are correct for reads with T-C conversions, deletions and truncations. To have a fair comparison between aligners that can produce split-alignments and the other methods, we only considered the reads that were unspliced in this analysis. We found, as it was expected, that the sensitivity of the aligners increased as the reads got longer (see Fig. 3). Moreover, we found that the sensitivity decreased as the number of diagnostic events increased.

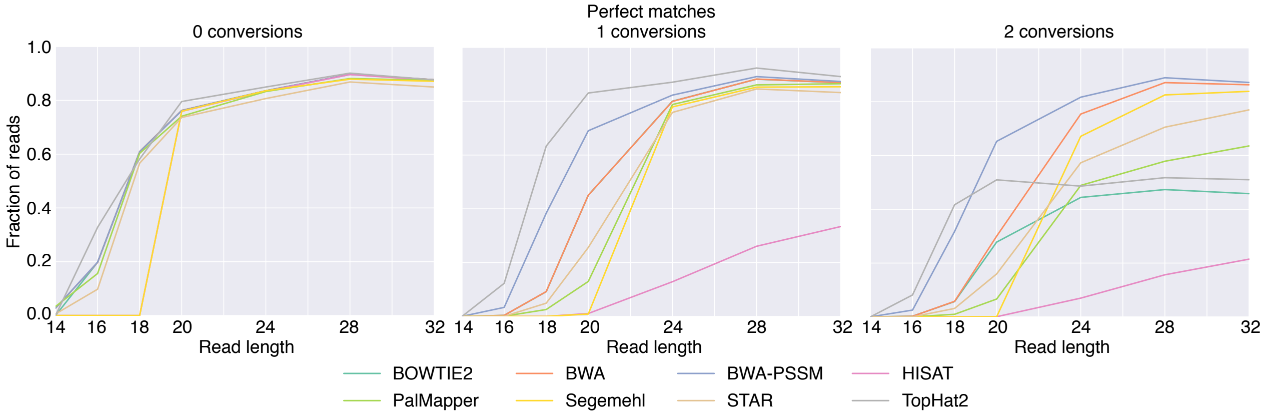

Fig. 3. Shown is the fraction of unspliced reads with 0 (left), 1 (middle) and 2 (right) conversions that map perfectly.

For reads with T-C conversions, we found that TopHat2 and BWA-PSMM had the highest sensitivity, although the sensitivity of TopHat2 for reads having two mismatches plateaued as the reads got longer. Furthermore, we found that the performance of HISAT was suboptimal for reads with mismatches. This was not surprising as it was developed to work with longer reads. We assume that the good performance of TopHat2 can be attributed to the strategy of TopHat2, to align reads to the transcriptome first and only perform alignment of reads to the whole genome when no good transcriptomic hit was found. This reduces the number of potential mapping locations significantly, thus also reducing the number of misalignments. To confirm that the high sensitivity of TopHat2 for short read lengths can indeed be attributed to the TopHat2 alignment strategy, we ran TopHat2 without providing a gene annotation (data not shown). This forced TopHat2 to align to the whole genome. We did this for the 16 bp long read-dataset. By doing this, the number of unspliced perfectly mapping reads dropped by 96%, showing that the transcriptome alignment was indeed responsible for the good performance on the short libraries. This suggests that the two-step alignment strategy might also be promising for CLIP-Seq data alignment pipelines that are using other aligners than TopHat2.

Next, we analysed the performance of the aligners for reads with simulated deletions (see Fig. 4). As we expected, the aligners achieved the same sensitivity on the reads without deletions as on the reads without T-C conversion. This was expected because we used the same reads as basis for the simulation of all three types of diagnostic events. For the reads with a deletion, we found that all algorithms could align less than 10% of the reads of length 20 and shorter to the correct location. For the reads that were 24 bp and longer, Segemehl had the highest sensitivity. The sensitivity of the other tools was considerably lower than in the T-C conversion setting. We assume that the high sensitivity of Segemehl may be the

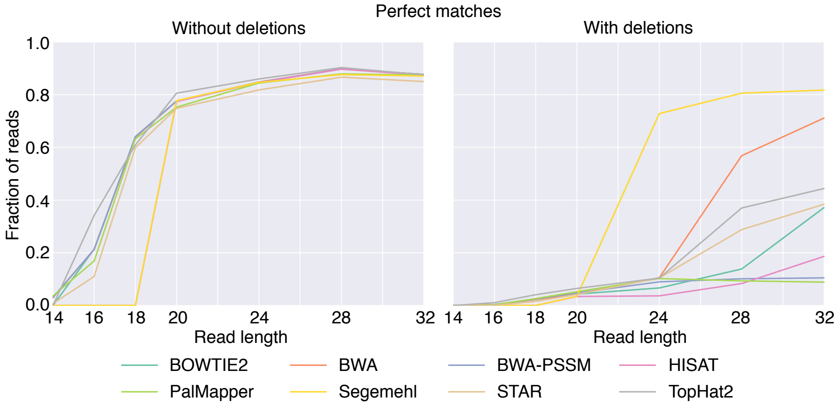

Fig. 4. Shown is the fraction of unspliced reads without and with deletions (left and right, respectively) that map perfectly.

consequence of the strategy to search, already at the seeding stage, for matches with deletions and insertions.

After this, we analysed the performance of the aligners for reads with truncations (see Fig. 5). For these, the performance of all the alignment tools on the reads with truncations reflected their performance on the reads without diagnostic events. This is because the libraries with truncations were basically a mixture of libraries for shorter read lengths without diagnostic event.

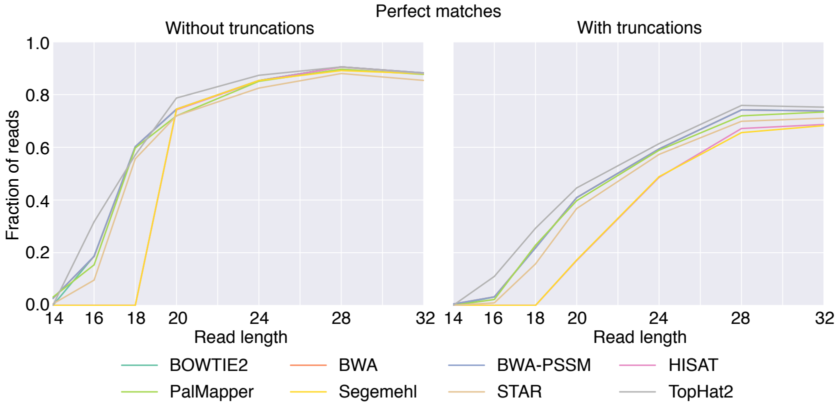

Fig. 5. Shown is the fraction of unspliced reads without and with truncations (left and right, respectively) that map perfectly.

In order to picture the overall performance of the selected alignment tools, we determined the fraction of reads that could not be mapped (unmapped), that mapped to multiple loci (multimapping), mapped to the wrong locus (errors) and both for spliced and unspliced reads, the reads that were correctly mapping (correct unspliced resp. spliced mapping). We performed this analysis for the reads with T-C conversions of length 32. Overall, we found that the fraction of reads in the different categories, varied substantially between the aligners (see Fig. 6). We found for example that the there were differences in the specificity of aligners.

Both BWA and BWA-PSSM could align a large fraction of the reads (95.6% resp. 97.4%). However, a substantial fraction of these alignments (9.0% resp. 10.3%) was mapping to the wrong location. In contrast, aligners such as STAR and TopHat2 were more conservative (i.e. did report more of the reads as unmapped): STAR and TopHat2 mapped 88.9% resp. 89.8% of the reads from which only 1.5% resp. 4.5% were mismapped.

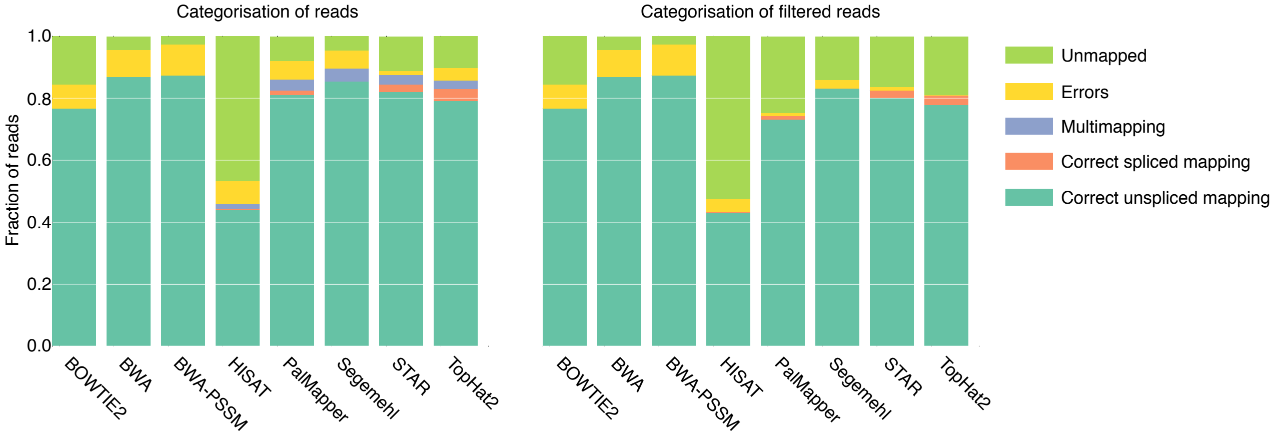

Fig. 6. Shown is the fraction of reads that were unmapped (light green), errors (yellow), multimapping (blue), correct spliced alignments (orange) and correct unspliced alignments (turquoise). Shown on right are the results for the T-C conversion dataset for reads of length 32. Shown on the left are the results for the dataset after filtering for multimappers.

3.3. *Alignment filtering*

A post-processing step that is commonly performed after the alignment, is removal of reads that map to multiple loci. The rational behind this is that for multimapping reads at least one alignment is wrong. This means that in the multimapping reads at least 50% map to a wrong location, thus their removal typically increases the quality of the alignment.

In order to understand how such filtering affects the CLIP-Seq data analysis, we first studied the influence of read-filtering on different categories of the alignments (as defined in the previous section). To this end, we filtered the reads of length 32 with T-C conversions for multimappers (see Sec. 2.5). We found, that the filtering affected the different alignment categories differently (see Fig. 6). Bowtie2, BWA and BWA-PSSM were not affected as with the parameters used, they only reported one alignment. For the other alignment tools, we observed that the filtering reduced the number of perfect matches and more strongly also the numbers of errors. This difference in reduction between the perfectly mapping reads and wrongly mapping reads was most striking for TopHat2, where only 5.8% of the correct mappings were removed but 93.8% of the errors. Overall, the filtering increased the specificity of the alignments. This suggests that filtering for multimappers is beneficial in settings where a high specificity is required.

3.4. *Peak calling*

The second step in the typical analysis of CLIP-Seq data analysis pipelines is the determination of binding sites by a peak calling. To analyse how the peak calling depends on the alignment tool that was used to align the reads, we applied PARalyzer, wavClusteR and BMix to all alignments of the 32 bp long reads with T-C conversions. We found that the number of peaks that were reported by the tree methods varied between the different alignments (see Fig. 7) and that PARalyzer had in general the lowest false positive rate. In our comparison wavClusteR detected between 4080 and 5867 peaks of which between 58% and 87% overlapped the true binding sites (n=5233). PARalyzer detected between 3336 and 4770 peaks of which between 74% and 95% overlapped the true binding sites and BMix detected between 3534 and 5948 peaks of which between 66% and 89% overlapped the true binding sites. Furthermore, we found that the fraction of true positives in the intersection of the peaks of all programs was in general higher than the fraction of true positives in the calls for each program (data not shown).

This shows that the choice of the alignments tool has a profound influence on the number of peaks that are called and suggest that it is important to use the same alignment strategy when comparing the performance of peak callers. We further investigated whether the difference in the number of peaks was due to the different numbers of reads that were aligned. Therefore, we evaluated the peak calling when the same number of reads was used from each aligner (the number of alignments in the smallest library). In order to exclude confounding of the result by the number of multimappers, we used the reads that have been filtered for multimappers.

We found that the number of clusters still showed a large variation (see Fig. 7). For wavClusteR, the number of peaks varied by 837 clusters, for PARalyzer by 973 and for BMix by 1093 clusters. This suggests that there are also systematic differences between results of the alignment tools. Furthermore, we found that filtering increased the fraction of true positives for all libraries for all tools.

4. Software

We have released the read simulation pipeline in a tool called Cseq-Simulator. This tools can be used under the GNU general public licence v.3. The tool can be obtained at: `https://ohlerlab.mdc-berlin.net/software/Cseq-Simulator_\%28Crosslinked-sequence_Simulator\%29_129/`

5. Discussion

In this work, we have presented Cseq-Simulator, a simulator for different types of CLIP-Seq data such as PAR-CLIP, HITS-CLIP or iCLIP. This simulator allows generation of datasets with known ground truth that exhibits several characteristics of real data, e.g. the read length or the diagnostic event profiles. In order to achieve a high resemblance of simulated and real data, we model different steps of the CLIP-Seq protocol and build on components of an existing RNA-Seq read simulator. For the binding site that are used for the simulation we provide two

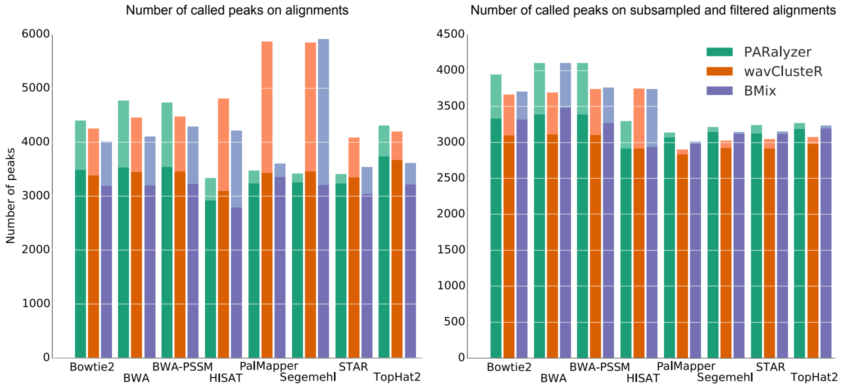

Fig. 7. Number of peaks that are called by PARalyzer, wavClusteR and BMiX. Light colors indicate the number of peaks that are called and dark colors the number true peaks that are called (n=5233). Shown on the right are the number of peaks that are called for reads with T-C conversions of length 32. Shown on the left are the number of peaks the multimapper-filtered and subsampled reads with T-C conversions of length 32.

options: (1) Prediction of the binding sites using a PWM. We expect that this provides a good approximation to RBP-binding in the case where the binding is mainly determined by the sequence. (2) A user provided list of binding site, which allows users to provide binding sites that are experimentally determined or derived using other models. We believe that the second option is particularly helpful when the RBP has a low sequence specificity or binding depends also on the secondary structure. Overall, Cseq-simulator allows modelling many aspects of CLIP-Seq datasets and can therefore be applied to simulate data for a broad range of RBPs and CLIP-Seq protocols.

Additionally, we have exemplified here, how simulated dataset can be used to assess the steps of a typical CLIP-Seq analysis pipelines. These analyses were performed for the read alignment, the peak calling and the interdependence of the two. In this assessment of the tools, we have made several interesting observations: For example, we have found that there was no best alignment tools for all CLIP-Seq data and that there was also a significant variation in the sensitivity and specificity of the alignments. When we compared PARalayzer, wavClusteR and BMix, we have observed that the number of peaks that were discovered, strongly depended on the choice of the alignment tool and that this was not only due to the different number of aligned reads. Overall, these observations show the potential of the Cseq-Simulator to inform decision on which tools to use for an CLIP-Seq data analysis.

A shortcoming of the benchmarking that we have carried out in this study is that we have mostly relied on default parameters for the tools. Therefore, the results might not reflect the optimal performance of the tools when tuned to a specific task. We would like to mention, however, that the simulated data is also valuable resource to improve the performance of the respective tools for CLIP-Seq data.

Another important point to mention is that, as the exact properties of CLIP-Seq data have not been characterised entirely, our simulations may not capture all aspects of this data. We did for example not simulate any biases. Therefore, the insights that have been gained on the basis of simulated data might not be entirely transferable to real data. However, we believe that independent of this shortcoming, important pitfalls in the data analysis can be identified, which could otherwise not be identified. In the future, we plan to extend Cseq-Simulator in order to also simulate stochastic binding and biases, e.g. the ones introduced by the choice of the restriction enzymes.

6. Acknowledgment

We would like to thank Neelanjan Mukherjee, Hans-Hermann Wessels and Alina-Cristina Munteanu and the reviewers of the manuscript for helpful discussions, comments and questions.

References

[1] M. Hafner, M. Landthaler, L. Burger, M. Khorshid, J. Hausser, P. Berninger, A. Rothballer, M. Ascano, A.-C. Jungkamp, M. Munschauer, A. Ulrich, G. S. Wardle, S. Dewell, M. Zavolan and T. Tuschl, *J Vis Exp* (2010).

[2] 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, *Nature* **456**, 464 (Nov 2008).

[3] J. Konig, K. Zarnack, G. Rot, T. Curk, M. Kayikci, B. Zupan, D. J. Turner, N. M. Luscombe and J. Ule, *J Vis Exp* (2011).

[4] J. König, K. Zarnack, N. M. Luscombe and J. Ule, *Nat Rev Genet* **13**, 77 (Feb 2011).

[5] D. L. Corcoran, S. Georgiev, N. Mukherjee, E. Gottwein, R. L. Skalsky, J. D. Keene and U. Ohler, *Genome Biol* **12**, p. R79 (2011).

[6] 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, *Bioinformatics* **28**, 3013 (Dec 2012).

[7] F. Comoglio, C. Sievers and R. Paro, *BMC Bioinformatics* **16**, p. 32 (2015).

[8] T. Griebel, B. Zacher, P. Ribeca, E. Raineri, V. Lacroix, R. Guigó and M. Sammeth, *Nucleic Acids Res* **40**, 10073 (Nov 2012).

[9] C. E. Grant, T. L. Bailey and W. S. Noble, *Bioinformatics* **27**, 1017 (Apr 2011).

[10] B. Langmead and S. L. Salzberg, *Nat Methods* **9**, 357 (Apr 2012).

[11] H. Li and R. Durbin, *Bioinformatics* **26**, 589 (Mar 2010).

[12] P. Kerpedjiev, J. Frellsen, S. Lindgreen and A. Krogh, *BMC Bioinformatics* **15**, p. 100 (2014).

[13] G. G. Faust and I. M. Hall, *Nat Methods* **9**, 1159 (Dec 2012).

[14] T. D. Wu and S. Nacu, *Bioinformatics* **26**, 873 (Apr 2010).

[15] D. Kim, B. Langmead and S. L. Salzberg, *Nat Methods* **12**, 357 (Apr 2015).

[16] G. Jean, A. Kahles, V. T. Sreedharan, F. De Bona and G. Rätsch, *Curr Protoc Bioinformatics* **Chapter 11**, p. Unit 11.6 (Dec 2010).

[17] S. Hoffmann, C. Otto, S. Kurtz, C. M. Sharma, P. Khaitovich, J. Vogel, P. F. Stadler and J. Hackermüller, *PLoS Comput Biol* **5**, p. e1000502 (Sep 2009).

[18] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson and T. R. Gingeras, *Bioinformatics* **29**, 15 (Jan 2013).

[19] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley and S. L. Salzberg, *Genome Biol* **14**, p. R36 (2013).