Chapter 2
Computational Analysis of PAR-CLIP Data

2.1 About CLIP

Currently, three techniques are available to study the binding of RNA-binding proteins (RBPs) to RNA at high resolution with high throughput methods: HITS-CLIP/CLIP-seq [15, 20], iCLIP [10], and PAR-CLIP [6]. All of them rely on the formation of covalent bonds between RBP and the bound RNA (commonly termed *crosslinking*), purification of the RBP-RNA complexes by immuno-precipitation (*CLIP*: **C**ross**L**inking and **I**mmuno-**P**recipitation), and high-throughput sequencing of RNA fragments from these complexes (reviewed in [16]). UV light is employed to induce crosslinking. In the case of PAR-CLIP, additional photo-activatable ribonucleoside analogues are incorporated into cellular RNA, which greatly enhance crosslinking efficacy and allow to use lower energy UV light, also reducing the damage to the fragile RNA molecules [6]. A crucial step in the experimental procedure is the generation of cDNA from the RBP-bound RNA fragments by reverse transcription (Fig. 2.1). The presence of a covalent bond between ribonucleic- and amino-acid alters the chemical and steric properties of the base and influences the activity of the reverse transcriptase enzyme. While normally undesirable, the artifacts in the cDNA sequence induced by such RNA lesions can be harnessed to pinpoint the exact nucleotide position of the crosslink. The prevalence and kind of such artifacts differs substantially between the three CLIP methods: While Zhang and Darnell show that HITS-CLIP induces predominantly nucleotide deletions [22] (albeit at a low frequency, data not shown), the iCLIP protocol assumes that the reverse transcription reaction in most cases will stop at the site of the lesion and consequently the end of CLIP reads demarcate the crosslink site. In the PAR-CLIP protocol, the thionucleoside analogues lead to a very high rate (up to ∼80 %) of mismatch mutations: U is read as C when using 4-thio-uridine, G is read as A when using 6-thio-guanosine as RNA label (Fig. 2.2).

M. Jens, *Dissecting Regulatory Interactions of RNA and Protein*, Springer Theses,
DOI: 10.1007/978-3-319-07082-7_2,

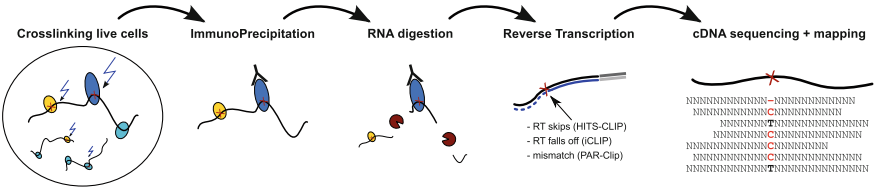


Fig. 2.1 RNA-protein crosslinking and immunoprecipitation (CLIP) methods. Current experimental methods to study the interactions of RNA-binding proteins (RBPs) induce covalent bonds (*red crosses*) between RNA (*black line*) and bound proteins (*filled ellipses*) by UV light (*blue arrows*) while the cells are intact to stabilize endogenous complexes. The efficacy of UV crosslinking can be enhanced by labeling cellular RNAs with photoactivatable ribonucleoside analogues (PAR-CLIP). Immunoprecipitation of a particular RBP enriches the RNA-RBP complexes of interest. Efficient and specific recovery of the bound RNA requires partial digestion of long mRNAs into fragments that are amenable to preparation of a cDNA sequencing library. Digestion typically involving the single strand specific endonucleases RNase-T1 or RNase-I (pacmans). For large scale analysis, a cDNA library of the bound RNA fragments is prepared for high-throughput sequencing. Due to chemical alterations, the reverse-transcriptase enzyme RT is prone to introduce errors into the DNA strand when it encounters crosslinked RNA bases. As such errors indicate the exact position of the crosslink, the downstream computational analysis benefits from extracting the signature of these events. In the case of HITS-CLIP, the RT is expected to read through the crosslink-induced RNA lesion, occasionally skipping the base and introducing deletions. In PAR-CLIP, the nucleoside analogue introduces specific mutations at a high rate (U is read as C), while in iCLIP, the RT is supposed to stop at the crosslink and the ends of cDNA reads demarcate the site of crosslinking

2.2 Computational Pipelines for PAR-CLIP Analysis

The sequencing data produced from a PAR-CLIP experiment contain information about the binding of a specific RBP on a transcriptome-wide scale, but to yield useful biological insight they need to be processed, filtered, annotated and comprehensively analyzed. In 2011 no software package to perform these analyses was publicly available. In the meantime a number of approaches have been published, each of them designed with a particular focus.

CLIPz [9] emphasizes collaborative analysis and represents itself as a web-based service that requires uploading all data to a CLIPz server. This is very convenient for experimenters but not the ideal choice for large data sets or if more control over the analysis is required.

PARalyzer [4] has roots in a HuR PAR-CLIP analysis which was published back-to-back [17] with our own work [12]. It performs read-mapping, cluster scoring and annotation, similar to our own work, but does not provide adaptive quality filtering and has no built-in support for consensus clusters or infrastructure to manage larger projects. A distinct feature is a kernel-density based segmentation of large clusters of overlapping reads into smaller binding sites based on the conversion profile. While typically only few clusters are large (Fig. 3.2c), they may contain high affinity binding sites (Neel Mukherjee, personal communication).

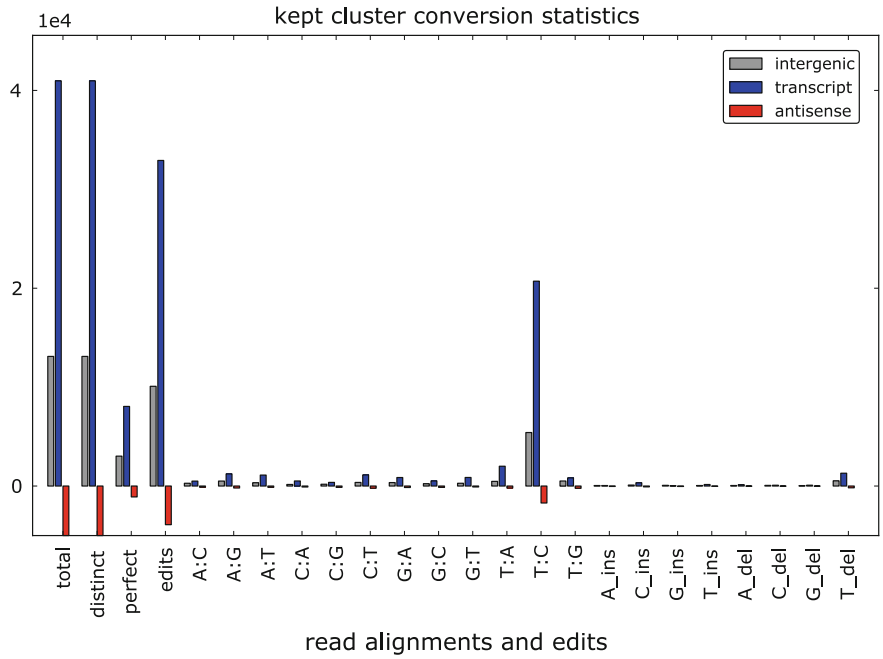


Fig. 2.2 Characteristic nucleotide conversions induced by efficient crosslinking in PAR-CLIP. Unique alignments of collapsed (distinct) read sequences from 4SU PAR-CLIP of HuR in SILAC medium cultured cells. Alignments are annotated against reference transcripts and classified as 'transcript' (*blue*), when aligning sense to known mRNAs, 'antisense' (*red*), when aligning exclusively antisense to known transcripts, or 'intergenic' (*gray*) when falling outside of known transcripts. Antisense alignments are considered as decoy hits and represented as negative numbers for illustrative purposes. A large fraction of reads aligns with edits (indel or mismatch). Out of all possible edit operations, mismatches of a 'T' in the reference and a 'C' in the read are strongly enriched, followed by less abundant T mutations and T deletions, indicating effective crosslinking of the 4SU

wavClustR [18] is available as a package for R and arbitrarily treats PAR-CLIP read clusters as signals that can be projected onto wavelet functions. While this was demonstrated to work for the analysis of MOV10 PAR-CLIP data, the scoring/filtering scheme is ad-hoc and not adaptive to the experiment.

2.3 The Rajewsky Lab Pipeline

Starting with the HuR project [12] our own computational analysis pipeline was developed to carry out the processing steps outlined in Fig. 2.3, either by use of publicly available tools or custom scripts. After publication of [12] the pipeline was subsequently improved, extended and partially modularized into a library. The majority of the code is written in Python [21], using the numerical Python

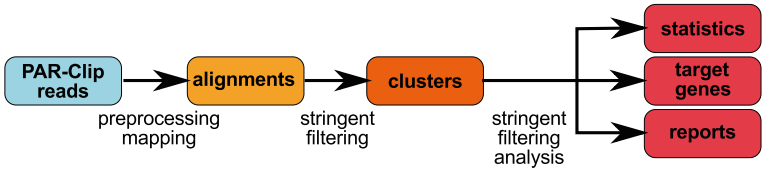


Fig. 2.3 A computational pipeline for PAR-CLIP analysis. Experimentally produced sequencing reads are cleaned of adapter sequences, length filtered, and collapsed into distinct (unique) read sequences. Reads are aligned to the reference sequence and unique, quality filtered alignments are grouped into clusters. Clusters are scored by various metrics and annotated against reference transcripts. Using decoys, the false discovery rate (FDR) can be optimized by selecting appropriate cutoffs on cluster quality scores until the FDR falls below a desired limit such that each resulting, FDR-filtered cluster confidently contains at least one binding site of the RBP. The resulting set is subjected to statistical analyses and target gene calling. Extensive diagnostic output is generated in the form of diagrams and HTML for each stage of the pipeline. The report pages interface with the UCSC genome browser [8] to facilitate exploration of the resulting binding sites

extension [3]. Time-critical parts are implemented in C++ [19] and imported/wrapped with boost.python [1]. Read alignments are stored in the SAM/BAM format and processed with samtools [14] and the respective Python bindings provided in the pysam module [7].

The computational pipeline was also used to re-analyze published human PAR-CLIP datasets and the results made publicly accessible via the genome browser and database of RNA interactions **doRiNA** [2] at http://dorina.mdc-berlin.de, hosted by the *Berlin Institute for Medical Systems Biology* (BIMSB).

The rest of this section presents the different stages of the processing and filtering in more detail.

2.3.1 CLIP Read Pre-processing

As the protocol involves partial digestion with RNase-T1 to focus sequencing on the RBP-bound RNA fragments, CLIP reads are typically short ($\sim 20nt$). Analogous to small RNA sequencing, the $3'$adapter sequence that is ligated to the RNA fragments for cDNA generation will many times appear in the $3'$ part of sequencing reads and requires trimming before reads can undergo further processing. The pipeline delegates this task to the FAR/FLEXBAR tool [5]. After adapter removal, reads shorter than 15 nucleotides are discarded. Remaining reads are collapsed into distinct sequences (each unique sequence is counted only once) by a Perl script contributed by Sebastian Mackowiak. This drastically reduces disk usage and the runtime of all downstream analyses. It also eliminates the impact of PCR artifacts that sometimes can lead to amplification of individual sequences.

2.3.2 Alignment to the Reference Sequence

Originally all reads were aligned with BWA [13]. More recently, support has been added for Bowtie2 [11], which allows more fine-grained control of alignment parameters and provides alignment scores in the output.

2.3.3 Clustering of Aligned Reads

Reads that align to contiguous stretches of the reference sequence are grouped into clusters. Clusters with only single reads are discarded. The algorithm makes use of the fact that the alignments are sorted on genomic coordinates and thus requires very little memory. Genomic strands are analyzed separately, as PAR-CLIP sequencing is strand specific.

2.3.4 Consensus Rules

Where available, biological replicates can be combined to yield consensus clusters. The current implementation allows to specify the minimal number of replicates that are required to support a given cluster by either read alignments or, more strictly, by characteristic conversions. For example, a consensus set could be built by demanding that clusters contain reads from 2/3 replicates (see Fig. 2.6). The application of such consensus rules typically reduce the FDR, prior to any additional, downstream filtering.

2.3.5 Annotation and Quality Scoring of Clusters

Clusters are annotated against a database of known transcripts and categorized as aligning either to

- a known transcript in sense orientation
- a known transcript in antisense orientation
- an unannotated region
- a region with overlapping transcription.

This distinction is exploited later on in downstream filtering steps. Additionally, a number of quality scores is computed for each cluster that derives from the set of aligning reads and their characteristic properties

1. number of (unique, distinct) read alignments
2. number of characteristic mismatches (T to C, G to A, . . .)

3. length of the cluster
4. maximum *uniqueness* of all read alignments
5. cumulative *uniqueness* of all read alignments
6. entropy score over read start/end positions
7. entropy score over read sequence variability.

Uniqueness refers to the margin between the reported, best alignment of a read and the second best alignment considered by the read mapper. In the case of BWA, this is a binary flag, indicating whether additional alignments exist with one additional edit operation (sometimes referred to as the *hull* of an alignment) or not. In the case of Bowtie2, uniqueness refers to the difference between the alignment score of the best and the second best alignment. Bowtie2 employs more complex scoring of edit operations, taking into account base quality scores and using affine gap penalties. The uniqueness scores computed from Bowtie2 alignments consequently offer a broader dynamic range.

2.3.6 False Discovery Rate Estimation

As PAR-CLIP reads are typically short and their sequence is mutated by effects of the crosslinking and other lesions induced during UV irradiation or RNA processing, they can not be expected to always align correctly to the reference sequence. A certain fraction will produce false alignments, leading to false-positive binding sites.

However, as the bound RNA fragments derive from biological, naturally occurring RNA and the sequencing strategy preserves strand information, we can expect the true-positives to align predominantly sense to known transcripts. Clusters aligning antisense to known transcripts, on the other hand, can be regarded as false-positives. In a few cases they may be true-positives, derived from un-annotated antisense transcripts, but these will be rare and typically much less abundant (Fig. 2.4). Consequently, treating antisense aligning clusters as false-positives, is a conservative assumption because it can only over estimate the number of false-positives produced by alignment artifacts.

Furthermore, as the aggregate amount of sense and antisense sequence is identical (ambiguous cases are put aside), one can regard the reverse complement of all transcripts, as a fair decoy database. In the absence of any real biological signal in the PAR-CLIP data we may expect an equal number of clusters to hit sense and antisense. Such a decoy database, therefore introduces a simple estimator of the false discovery rate (FDR) in the set of all PAR-CLIP read clusters:

$$\widehat{FDR} = \frac{\#decoy + 1}{\#sense + \#decoy + 2}\,. \tag{2.1}$$

The pseudo counts correspond to the bayesian estimate of $\widehat{FDR} = 50\,\%$ in the absence of any data.

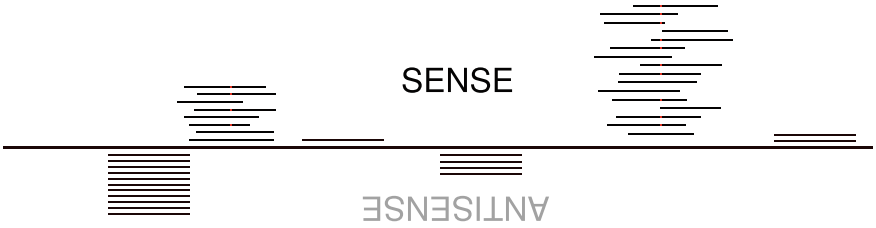


Fig. 2.4 Decoy (antisense) hits, allow to estimate and optimize the false-discovery rate of PAR-CLIP experiments. If antisense clusters are indeed false-positives, they represent mapping or PCR artifacts. As such, they can be expected to display a lower variability of read alignments, less margin between the score of the reported alignment and the second-best alternative (uniqueness), less conversions, less reads, or shorter reads than the true interactions. Each of these features can be measured as a cluster quality score and tested for discriminative power in distinguishing 'transcript' from 'decoy' on the basis of the score distribution. This allows to select an optimal score and reduce the FDR, by eliminating 'transcript' aligning clusters with similar properties as 'decoy' clusters and thereby enriching for true positives

2.3.7 Adaptive Cluster Filtering

With the FDR estimator at hand, it is possible to assess the effect of filtering the cluster set by setting thresholds on their quality scores. If the antisense clusters indeed represent mapping artifacts, the corresponding quality score distribution should differ from the sense clusters, which supposedly contain the true-positives. This would allow to find cutoffs that deplete false-positives more strongly than true-positives and improve the FDR. It is important to bear in mind, that mapping artifacts may also align sense to known transcripts. Utilizing the antisense clusters to select cutoffs will arguably serve to also deplete the false-positives among the sense aligning clusters. On the other hand, the filtering should discard as little real data as possible.

To find the best compromise, the pipeline code iterates over each of the aforementioned cluster quality scores and estimates the FDR at each quality score that actually appears in the data, effectively probing the whole range of possible cutoffs (Fig. 2.5). If a score cutoff serves to reduce the FDR below a desired limit (the default is $\widehat{FDR} \leq 5\,\%$) it is recorded, together with the number of sense clusters that surpass the cutoff and would be retained. Out of all score/cutoff combinations that satisfy the FDR limit, the one preserving the largest number of sense aligning clusters is chosen. After the cutoff is applied, remaining decoy clusters are discarded and a cluster set is reported that can be expected to satisfy the FDR constraint.

This procedure provides two major benefits. First, the filtering automatically adapts to the quality of the data and does not rely on empirically chosen *magic numbers* as cutoffs. Over digested, or strongly PCR amplified libraries generated from too little input material, or poor crosslinking efficiency will be ruthlessly filtered down to the strongest signals that are hopefully still biologically meaningful, while good experiments with high quality reads will undergo relaxed filtering and can yield a large number of clusters. This has been seen to work in practice on at

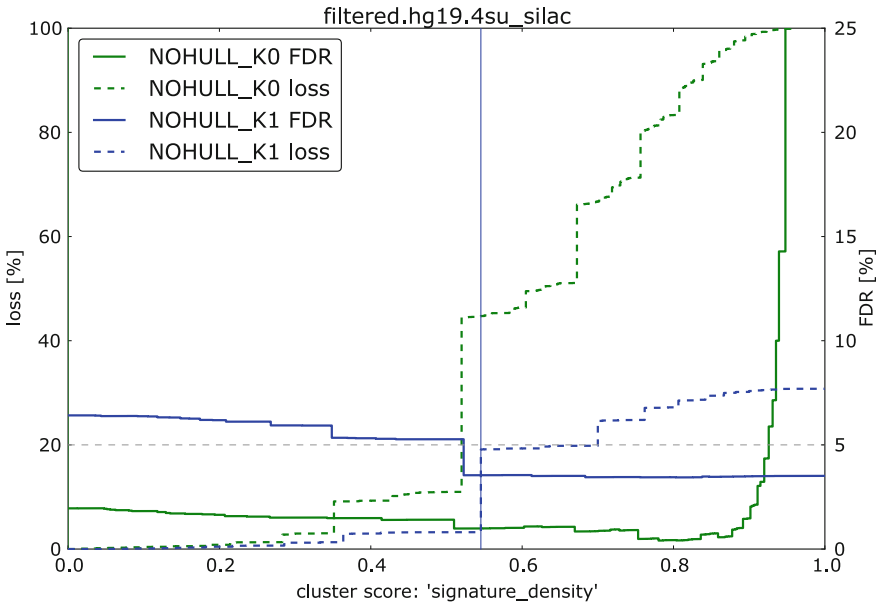
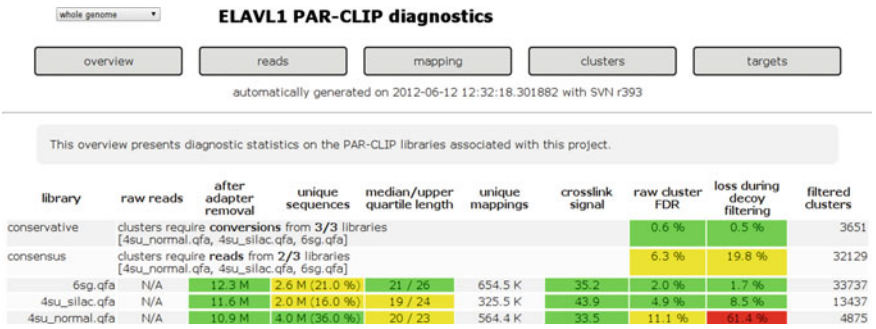


Fig. 2.5 Adaptive score/cutoff selection to control the FDR The false discovery rate (FDR, *solid lines*) and the fraction of total clusters that do not meet the cutoff (loss, *dashed lines*) are plotted as a function of score cutoff. The scoring function 'signature_density' denotes the proportion of reads in a cluster that carry a mutation consistent with crosslinking (the experiment shown was performed with 4SU, so T-C mismatches and T deletions count as *signature*). Additionally the clusters are broken down into *uniqueness* and edit distance categories extracted from the output of BWA. *Green* clusters contain at least one read which aligns perfectly and without additional alignments at edit-distance 1 ('NOHULL_K0'). *Blue* the best read aligns with one edit operation, but without additional alignments at edit-distance 2 ('NOHULL_K1'). *Blue vertical line* the selected cutoff which satisfies FDR $< 5\,\%$ for the complete set. Generally, the FDR decreases with higher cutoffs, while the fraction of 'lost' clusters increases. Perfect matches in a cluster (*green*, 'K0') indicate lower FDR, except at very high conversion densities, which may stem from actual T to C mutations in the HEK293 cells or other sources of false mappings

least a dozen PAR-CLIP libraries, with the stringency of the filtering usually corresponding to the experimenters assessment of how well the experiment went. Second, the selected scoring function allows to draw conclusions about the quality of the experiment. If the algorithm finds that filtering by "signature_density" (the fraction of reads with characteristic mutations) gives optimal results, crosslinking most likely was effective and useful to delineate direct binding from artifacts. On the other hand, "length" or "uniqueness" indicate over digestion resulting in very short reads that just barely align uniquely to the reference and consequently provide little headroom for the extra mutations introduced by crosslinking. Again, this usually matches the observations of experimenters. An overview (Fig. 2.6), as well as more in-depth diagnostic output, are generated as HTML.



library	raw reads	after adapter removal	unique sequences	median/upper quartile length	unique mappings	crosslink signal	raw cluster FDR	loss during decoy filtering	filtered clusters
conservative	clusters require **conversions** from **3/3** libraries [4su_normal.qfa, 4su_silac.qfa, 6sg.qfa]						0.6 %	0.5 %	3651
consensus	clusters require **reads** from **2/3** libraries [4su_normal.qfa, 4su_silac.qfa, 6sg.qfa]						6.3 %	19.8 %	32129
6sg.qfa	N/A	12.3 M	2.6 M (21.0 %)	21 / 26	654.5 K	35.2	2.0 %	1.7 %	33737
4su_silac.qfa	N/A	11.6 M	2.0 M (16.0 %)	19 / 24	325.5 K	43.9	4.9 %	8.5 %	13437
4su_normal.qfa	N/A	10.9 M	4.0 M (36.0 %)	20 / 23	564.4 K	33.5	11.1 %	61.4 %	4875

Fig. 2.6 HTML overview of a pipeline run. Multiple PAR-CLIP experiments can be combined in a project and analyzed together. Support for consensus clusters (read support from multiple, independent libraries) is built in. The colors are chosen to empirically match the quality metrics and highlight possible problems. The '6sg' and especially the '4su_silac' experiments have a small fraction of distinct reads ('unique sequences'), indicating a reduction of cDNA diversity by PCR amplification. The 'median/upper length' of adapter removed reads is overall in the $20+$ *nt* range, with the 6SG experiment having longer reads. 'crosslink signal' is the percentage of read mutations that matches the crosslink signature. The 'raw cluster FDR' is the false-discovery rate estimated by treating antisense clusters as decoy hits. Consensus building clearly serves to reduce the FDR, by comparison to the individual libraries. 'Loss during decoy filtering' is the percentage of clusters that align antisense or do not meet the selected minimal cutoff. For '4su_normal' the selected cutoff was rather high and eliminated 61 % of clusters. For unknown reasons, this library had a high ratio of conversions among decoy-aligned reads (not shown, but part of diagnostic output)

2.3.8 Possible Improvements

While the FDR limiting code works well in practice, combinations of scores offer the possibility to enhance both sensitivity and specificity of the filtering (Fig. 2.5 demonstrates that clusters with perfect matches would require no filtering at all, but the large number of 'K1' clusters require a cutoff that removes $\sim$50 % of 'K0' clusters). An option currently explored for future improvements of the filtering is a principle component analysis over all computed cluster scores and the monitoring of false positives due to mapping (random sequence decoys), but also high abundance RNAs (tRNA, rRNA, motivated by Neel Mukherjee, personal communication). Determining a hyperplane that distinguishes between true and false positives represents a multi-dimensional generalization of the current filtering scheme by a cutoff. However, a balance must be found between accurate filtering and the number of parameters to estimate from the data, otherwise the empirically derived FDR may become misleading due to over fitting.

2.3.9 Target Gene Calling

As the FDR filtered cluster set is already annotated against a set of known transcripts, it is only a matter of counting to produce a list of genes that are targeted by the studied

RBP. The quantitative information contained in the PAR-CLIP clusters that fall into a gene (number of clusters, number of characteristic conversions, number of reads) is preserved and combined with the gene structure, counting clusters separately that fall into introns or exons, and furthermore distinguish between 5′UTR, CDS, 3′UTR, where applicable. The resulting table of target genes can be sorted by each combination of cluster score and gene segment (Fig. 2.7). The different rankings allow to test predictions about the functional consequences of binding, when intersected with perturbation data.

2.3.10 HuR Analysis as an Example

The following python code is sufficient to create a directory structure with Makefiles that carry out the complete analysis from reads to clusters and target gene tables, including HTML reports. The use of Makefiles simplifies the addition of new data sets and limits updates to files that depend on changed input.

```
from sequence_data.parclip.setup import Project

P = Project(
    "/data/BIO2/pcp/projects/elavl1",
    protein_name="ELAVL1",
    n_threads=4,
    max_edits=1,
    system='hg19',
)

lane1 = "/data/deep_seq3/solexa/100507_HWUSI-EAS1620_0005_61C42_CHEN"
lane2 = "/data/deep_seq3/solexa/100730_HWUSI-EAS1620_0014_624KR_HUEBNER-MIXED"
lane3 = "/data/deep_seq3/solexa/100730_HWUSI-EAS1620_0014_624KR_HUEBNER-MIXED"

P.add_reads("4su_normal",lane1+"/Data/Intensities/BaseCalls/s_3_1_*_qseq.txt")
P.add_reads("4su_silac" ,lane2+"/Data/Intensities/BaseCalls/s_3_1_*_qseq.txt")
P.add_reads("6sg"       ,lane3+"/Data/Intensities/BaseCalls/s_4_1_*_qseq.txt")

P.consensus_clusters(
    "consensus",
    ["4su_normal","4su_silac","6sg"],
    min_support=2,
    require_conversions=False
)

P.consensus_clusters(
    "conservative",
    ["4su_normal","4su_silac","6sg"],
    min_support=3,
    require_conversions=True
)

P.generate()
```

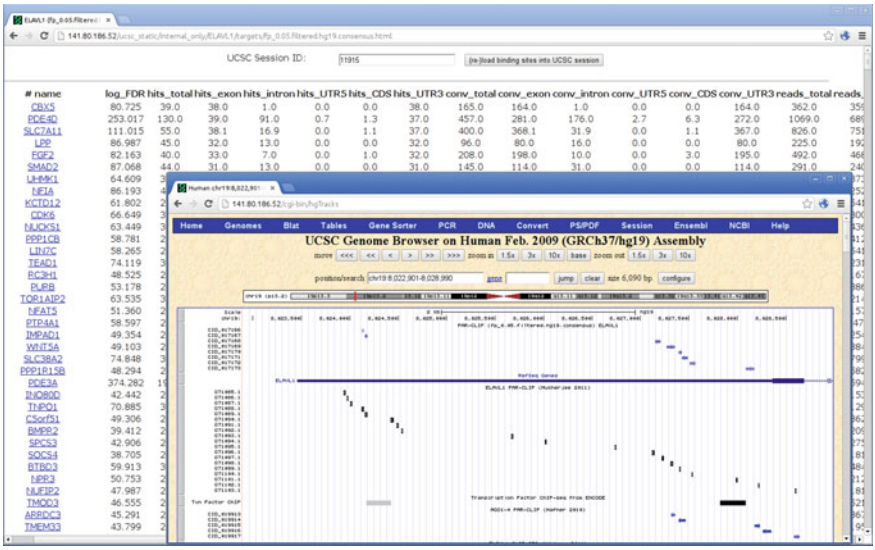


Fig. 2.7 Sortable target gene list, connected to the UCSC genome browser. *Left*, *background* a table of genes with PAR-CLIP clusters can be ranked by a click on the table header (number of clusters, conversions, reads—in introns, exons, 5′UTR, CDS, 3′UTR, total. In the screenshot the table is sorted by the number of 3′UTR clusters in descending order). The page connects to a local mirror of the UCSC genome browser [8] (hosted by the BIMSB [2]) to upload binding sites and remote-control the session. This conveniently allows to explore binding sites of interesting targets. *Right, foreground*) UCSC genome browser view of the HuR (ELAVL1) 3′UTR. Clusters are represented as colored blocks (*red* + strand, *blue* − strand) and anchor positions with most conversion events indicated by a *vertical line*. The data can be viewed in the context of other data sets available in doRiNA [2], and arbitrary custom tracks that can be visualized by the UCSC browser (sequencing coverage, etc.)

2.3.11 Statistical Binding Site Analysis

PAR-CLIP experiments can yield tens of thousands of FDR filtered clusters. As each cluster represents at least one in vivo binding site of the studied RBP, the cluster set can be thought of as samples from the ensemble of bound states of the RBP. Statistical analysis of the cluster set can therefore shed light on the characteristics of the recognition process carried out by the RNA binding domains of the RBP. In the next section, average profiles of binding site conservation and secondary structure will be computed, as well as an analysis of sequence specificity.

References

1. D. Abrahams, U. Koethe, R.W. Grosse-Kunstleve et al., The Boost Python Library. Comput. Softw. (2002). http://www.boost.org/libs/python
2. G. Anders, S.D. Mackowiak, M. Jens, J. Maaskola, A. Kuntzagk, N. Rajewsky, M. Landthaler, C. Dieterich, doRiNA: a database of RNA interactions in post-transcriptional regulation. Nucleic Acids Res. **40**(D1), D180–D186 (2012)
3. D. Ascher, P.F. Dubois, K. Hinsen, J. Hugunin, T. Oliphant et al., Numerical python (2001). http://sourceforge.net/projects/numpy
4. D.L. Corcoran, S. Georgiev, N. Mukherjee, E. Gottwein, R.L. Skalsky, J.D. Keene, U. Ohler et al., PARalyzer: definition of RNA binding sites from PAR-CLIP short-read sequence data. Genome Biol. **12**(8), R79 (2011)
5. M. Dodt, J.T. Roehr, R. Ahmed, C. Dieterich, FLEXBARG–flexible barcode and adapter processing for next-generation sequencing platforms. Biology **1**(3), 895–905 (2012)
6. M. Hafner, M. Landthaler, L. Burger, M. Khorshid, J. Hausser, P. Berninger, A. Rothballer, M. Ascano Jr, A.-C. Jungkamp, M. Munschauer, A. Ulrich, G.S. Wardle, S. Dewell, M. Zavolan, T. Tuschl, Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. Cell **141**(1), 129–141 (2010)
7. A. Heger et al., Pysam (2009). http://code.google.com/p/pysam
8. A.S. Hinrichs, D. Karolchik, R. Baertsch, G.P. Barber, G. Bejerano, H. Clawson, M. Diekhans, T.S. Furey, R.A. Harte, F. Hsu, J. Hillman-Jackson, R.M. Kuhn, J.S. Pedersen, A. Pohl, B.J. Raney, K.R. Rosenbloom, A. Siepel, K.E. Smith, C.W. Sugnet, A. Sultan-Qurraie, D.J. Thomas, H. Trumbower, R.J. Weber, M. Weirauch, A.S. Zweig, D. Haussler, and W.J. Kent. The UCSC genome browser database: update 2006. Nucleic Acids Res. 34(Database issue), D590–598 (2006).
9. M. Khorshid, C. Rodak, M. Zavolan, CLIPZ: a database and analysis environment for experimentally determined binding sites of RNA-binding proteins. Nucleic Acids Res. **39**(suppl 1), D245–D252 (2011)
10. J. König, K. Zarnack, G. Rot, T. Curk, M. Kayikci, B. Zupan, D.J. Turner, N.M. Luscombe, J. Ule, iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. Nat. Struct. Mol. Biol. **17**(7), 909–915 (2010)
11. B. Langmead, S.L. Salzberg, Fast gapped-read alignment with Bowtie 2. Nat. Methods **9**(4), 357–359 (2012)
12. S. Lebedeva, M. Jens, K. Theil, B. Schwanhäusser, M. Selbach, M. Landthaler, N. Rajewsky, Transcriptome-wide analysis of regulatory interactions of the RNA-binding protein HuR. Mol. Cell 43(3), 340–352 (2011). ISSN 1097–2765. doi:10.1016/j.molcel.2011.06.008. http://www.sciencedirect.com/science/article/pii/S1097276511004229
13. H. Li, R. Durbin, Fast and accurate short read alignment with burrows-wheeler transform. Bioinformatics **25**(14), 1754–1760 (2009)
14. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin et al., The sequence alignment/map format and samtools. Bioinformatics **25**(16), 2078–2079 (2009)
15. 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, R.B. Darnell, HITS-CLIP yields genome-wide insights into brain alternative RNA processing. Nature **456**(7221), 464–469 (2008)
16. Miha Milek, Emanuel Wyler, Markus Landthaler, *Transcriptome-wide analysis of protein-RNA interactions using high-throughput sequencing, in Seminars in Cell & Developmental Biology* (Elsevier, Amsterdam, 2011)
17. N. Mukherjee, D.L. Corcoran, J.D. Nusbaum, D.W. Reid, S. Georgiev, M. Hafner, M. Ascano, T. Tuschl, U. Ohler, J.D. Keene, Integrative regulatory mapping indicates that the RNA-binding protein HuR couples pre-mRNA processing and mRNA stability. Mol. cell **43**(3), 327–339 (2011)

18. C. Sievers, T. Schlumpf, R. Sawarkar, F. Comoglio, R. Paro, Mixture models and wavelet transforms reveal high confidence RNA-protein interaction sites in MOV10 PAR-CLIP data. Nucleic Acids Res. **40**(20), e160–e160 (2012)
19. B. Stroustrup, The C++ programming language (Addison-Wesley Longman Publishing Co., Inc, 1997).
20. J. Ule, K.B. Jensen, M. Ruggiu, A. Mele, A. Ule, R.B. Darnell, CLIP identifies nova-regulated RNA networks in the brain. Science **302**(5648), 1212–1215 (2003)
21. G. Van Rossum, F.L Drake Jr., Python reference manual. Centrum voor Wiskunde en Informatica (1995). http://www.python.org
22. C. Zhang, R.B. Darnell, Mapping in vivo protein-RNA interactions at single-nucleotide resolution from HITS-CLIP data. Nat. Biotech. **29**(7), 607–614 (2011)

http://www.springer.com/978-3-319-07081-0

Dissecting Regulatory Interactions of RNA and Protein
Combining Computation and High-throughput
Experiments in Systems Biology

Jens, M.

2014, XVIII, 99 p. 38 illus., 19 illus. in color., Hardcover
ISBN: 978-3-319-07081-0