Published online XX XXX XXXX *Nucleic Acids Research, XXXX, Vol. XX, No. XX* ***1–43***
doi:10.1093/nar/gkn000

SUPPLEMENTARY DATA
RIPSeeker: a statistical package for identifying protein-associated transcripts from RIP-seq experiments

Yue Li [1,2], Dorothy Yanling Zhao [2,3] Jack Greenblatt [2,3,4] and Zhaolei Zhang [1,2,3,4,*]

[1]Department of Computer Science, [2]The Donnelly Centre, University of Toronto, Toronto, ON, M5S 3E1, Canada, [3]Department of Molecular Genetics, University of Toronto, Toronto, ON, M5S 1A4, Canada and [4]Banting and Best Department of Medical Research, University of Toronto, Toronto, ON, M5S 3E1, Canada

Received October 11, 2012; Revised January 20, 2013; Accepted X XX, XXXX

QUALITY CONTROL OF RAW READ LIBRARY

Current sequencing machines produce reads with low accuracy and are prone to various contaminants. Proper preprocessing is necessary for reliable analyses. Raw RIP-seq read libraries were subject to a series of quality control (QC). First, an automated report of basic statistics for each raw read library was generated using FastQC program (`http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/`). In particular, FastQC reports "per base sequence quality" and "overrepresented sequences". The former indicates the averaged Phred quality at each individual base position of the reads. Sequencing quality usually drops off as reads go from 5′-end toward 3′-end. Poor quality reads will unlikely map to the reference genome with reasonable number of mismatches allowed (default: 2 mismatches). The latter reports enriched sequences according to a list of contaminants provided by the user. We compiled a custom list of common contaminants including Illumina adapter and primer sequences for single and paired-end sequencing, species specific (human or mouse) mitochondria genome, ribosomal and actin RNA, and phi X genome. Comparison of read library against this list will reveal the common contaminants and facilitate further experimental design and formulation of filtering strategy described next.

To improve the subsequent alignment quality, we devised an automated filtering program consisting of the following preprocessing steps. Reads containing non-determinant nucleotides ('N') are filtered out using `fastx_clipper` from FASTX-Toolkit (`http://hannonlab.cshl.edu/fastx_toolkit/index.html`). Bases lower than a defined Phred quality threshold (default: 20) at the 3′ end are trimmed off from each read using `cutadapt` (`http://code.google.com/p/cutadapt/`) (1). Next, known Illumina primers and adaptor sequences are clipped off from each read by `cutadapt`, which computes sensitive semi-global alignments of all the reads against all the primer/adaptor sequences, allowing gapped and mismatched alignments. Finally, filtered reads are aligned against the aforementioned custom contaminant list using Bowtie to further filter contaminant (aligned) reads (2).

ALIGNMENT OF FILTERED RIP-SEQ READ LIBRARY TO REFERENCE GENOME

TopHat (version 1.4.1) was used to align RIP-seq library to mouse (mm9 build) or human (hg19) reference genome (3). Following the default setting, multihits mapped to more than 20 distinct loci were discarded. To account for splicing junction, each read is cut up into segments of half of its total length `$s` with option `--segment-length`. In addition, a list of exon annotations from Ensembl (v65) was supplied as a parameter option (`--GTF`) to TopHat. Based on this annotation, the program will build a set of known splice junctions for each gene and attempt to align reads to these junctions even if they would not normally be covered by the initial mapping. This will presumably increase the sensitivity of the mapper. For the strand-specific sequencing, the library type (`--libType`) was set to "fr-secondstrand" consistent with the RIP-seq experiments where the second strand of the cDNA (i.e., the opposite strand of the PRC2-bound RNA) was sequenced. To account for PCR artefacts, only reads aligned to the distinct genome coordinates are retained. This post-alignment filtering is achieved using `samtools rmdup` from Samtools (4) and `MarkDuplicates` from Picard (`picard.sourceforge.net/command-line-overview.shtml`). Table S1 presents the basic statistics from the above analysis.

Bowtie (version 0.12.7) was used to align reads from PAR-CLIP library to human (hg19) reference genome following the alignment setting recommended in the README file of the PARalyzer program (version 1.1) (5). Specifically, two mismatches (`-v 2`) and up to 10 multihits per read (`-m 10`) are allowed, and only the best alignment (`--best`) is reported. Alignment in both BOWTIE and SAM formats are produced as input to PARalyzer and other peak callers, respectively (described below).

*To whom correspondence should be addressed. Tel: +1 (416) 946-0924; Fax: +44 000 0000000; Email: zhaolei.zhang@utoronto.ca

CHIP-SEQ PROGRAM SETTINGS

Alignment conversion

The alignment files in BAM (TopHat outputs) were converted to SAM using `samtools view` and to BED using `bamToBed` from BEDTools (6) as compatible input formats for the ChIP-seq algorithms QuEST and HPeak, respectively. For strand-specific library, as a fair comparison with the proposed program RIPSeeker, alignments on + and - strand from the BAM, SAM, and BED files were extracted and provided as separate inputs to MACS, QuEST, and HPeak respectively in order to force each peak caller to model read alignments in the strand-specific manner. The strand information is crucial to the identification of strand-specific lncRNA. Notably, both *strand-specific* sequencing experiments generated sequences from the opposite strand of the protein-bound RNA. Accordingly, the strand signs in the alignment files were switched (i.e. + to -, - to +) before providing to the peak callers. For the PAR-CLIP data, the alignment files in SAM were converted to BAM files as input to MACS and RIPSeeker and further converted to BED file as input to HPeak.

QuEST

QuEST (version 2.4) (7) was provided with required SAM formatted alignment files of RIP and control. The program allows user to select from three types of ChIP experiments including "TF", "RNAPII-like", and "Histone ChIP" in terms of the range and size of the read-enriched regions. For our purpose, we select "RNAPII-like" option for both RIP-seq datasets to reflect the median length of ∼200 bp for the majority of the known lncRNA based on Ensembl annotation (v65). (Figure S6). QuEST also requires a genome table specifies the sequence length on each chromosome in a space-delimited (not tab) format. Such table was downloaded from UCSC Genome Browser for both mouse and human and formatted in the expected way. The options are piped to QuEST prompt to run jobs on the server without user-interaction. All other settings are set by default.

MACS

MACS (version 1.4.1) was provided with BAM alignment files of RIP and control. The "–nomodel" and "–shiftsize=1" options were set to disallow MACS to estimate fragment size and to minimize the shift distance (–shiftsize must be nonzero as required by the program). Unlike double-stranded DNA in ChIP-seq, RIP library contain the single-stranded RNA, which does not have symmetrical enrichment on both strands to enable the estimation of fragment size and localization of "binding site". Please referred to (8) and the online manual (`http://liulab.dfci.harvard.edu/MACS/`) for more details. The genome size was set to be species-specific (`--gsize mm` for mouse and `--gsize hs` for human) for PRC2 and the other RIP-seq data for human, respectively. Option `--call-subpeaks` was enabled to refine the MACS peaks and split the wide peaks into smaller subpeaks.

HPeak

HPeak 3.0 (`https://sourceforge.net/projects/hpeak/files/latest/download`) was provided with BED formatted files with species option (`-sp`) set to "mouse" and "human" for PRC2 and the other RIP-seq data for human, respectively. The minimum (`-fmin`) and maximum (`-fmax`) fragment size was set to 200 and 1200 nt to be consistent with the selected fragment size in the RIP-seq experiments.

RNA-SEQ PROGRAM CUFFLINKS AND CUFFDIFF SETTINGS

Cufflinks (version 1.3.0) was applied to individual BAM files from **Alignment of Filtered RIP-seq Read Library to Reference Genome** to first assemble transcripts from each alignments separately. For each assembly tasks, Ensembl (v65) annotation files for mouse and human was provided to Cufflinks as the parameters for `--GTF-guide` option for the PRC2 and other human RIP-seq data, respectively. With the annotation guide, Cufflinks generates faux read alignments from the known exons and compare these artificial read coverage with real read alignment coverage to improve detection accuracy for novel transcripts (9). The options `--multi-read-correct` and `--frag-bias-correct` were also activated to improve prediction accuracy. For strand-specific RIP-seq data (PRC2 and CCNT1 Biorep1, Table S1), the option `--library-type` was set to "fr-secondstrand" otherwise to "fr-unstranded" (Table S1).

The assembled transcripts for the RIP-seq data (including the control) from Cufflinks were merged with the known transcripts from Ensembl (v65) via option `--ref-gtf` into a single GTF file using `cuffmerge`. The resulting GTF transcript file was provided to `cuffdiff` for differential analysis between RIP and control library (each having two biological replicates). To maximize prediction accuracy, options `--frag-bias-correct` (requiring fasta genome sequence) and `--multi-read-correct` were turned on. Only transcripts with default significant positive log2(fold-change) of normalized RPKM in RIP versus control comparison were retained as predicted protein-intersecting RNAs.

BIOCONDUCTOR PACKAGE DESEQ SETTINGS AND RESULTS

As another RNA-seq comparison strategy, we applied Bioconductor package DESeq (10) to the read count table computed over all of the transcripts from the latest Ensembl for human or Ensembl 65 (mm9) for mouse for RIP and control library. DESeq is

applicable only to RIP-seq libraries as PAR-CLIP does not have control library. To have a stable variance (dispersion) estimate, we applied DESeq to only the transcripts with read count larger than 5 in RIP or control. Similar to Cuffdiff, we compared gene expression of RIP with control to identify transcripts with fold-change greater than 1 and BH-adjusted p-value (or FDR) $<= 0.1$. To estimate dispersions, we experimented with different parameter settings to maximize the yield of the significant transcripts above the cutoff: `estimateDispersions( cds , method="pooled", sharingMode="fit-only", fitType="local")` (the default `fitType="parametric"` fitting dispersion with gamma GLM failed to fit the data).

For PRC2 data, 6400 out of the 95883 Ensembl transcripts for mouse have read count greater than 1. With the most sensitive empirical settings for DESeq, only one transcript for PRC2 dataset (ensembl id: ENSMUST00000124738; external gene id: Gm12992, a predicted gene) with a positive fold-change of 3.64 passed the BH-adjust p-value cutoff (adjusted p-value < 0.02). The known PRC2 RNA-interactors Meg3, Xist, Tsix and all have adjusted p-value equal to 1 and smallest unadjusted p-values (for multiple transcripts) 0.95, 0.89, and 0.65, respectively. For CCNT1, no transcript passed the same 0.1 cutoff. Similar results were observed for the ENCODE data, where only one transcript ENST00000261254 from PABPC1 versus T7Tag comparison in K562 have passed the 0.1 adjusted p-value cutoff and have positive fold-change of 3. *Due to small fraction of positive hits for all test RIP-seq datasets, we omitted the comparison with DESeq in the main text.*

Although DESeq is a powerful tool and has been successfully applied in many RNA-seq analyses, its dispersion estimation may be over-stringent to RIP-seq analyses. Comparing with RNA-seq analyses, the insufficient power of DESeq in RIP-seq analysis is likely due to the sparseness of the RIP-seq read count input data. The first problem might be potentially alleviated by further restricting hypothesis testings on transcripts with greater minimum read count as suggested in the DESeq vignette. In our case, however, testing on only 1555 and 718 transcripts with at least 5 and 10 read counts, respectively, in RIP or control data for PRC2 did not change the statistical conclusion. The same applied to the other RIP-seq test data. This also underscores the limitation of the current RIP-seq protocol. More effective IP followed deeper sequencing may lead to better results when applying DESeq-like approach. Notably, DESeq (and Cuffdiff) is non-applicable to the PAR-CLIP data, which does not have an external control library.

PAR-CLIP PROGRAM PARALYZER SETTINGS

For each PAR-CLIP library, the `sample.ini` file required by PARalyzer (version 1.1) (5) was prepared with identical setting as in the default `Sample.ini` file provided by the program package. The required `hg19.2bit` file for human genome was downloaded from UCSC genome browser.

SUPPLEMENTARY TABLES

Table S1. Source information and mapping statistics for the RIP-seq and PAR-CLIP data used in the paper

Cell line	Protein	Platform	Sample	GEO Accession	Strand-specific	Read Length (nt)	Pilot Reads	Reads After Filter	Total Mapped	Distinct Mapped
mESC	PRC2	RIP-seq	biorep1	GSE17064	Yes	36	14,359,505	5,292,794	1,839,385	1,022,474
mESC	PRC2	RIP-seq	biorep2	GSE17064	Yes	36	6,410,602	2,021,856	1,598,060	442,030
mESC	PRC2	RIP-seq	mutant	GSE17064	Yes	36	6,861,940	2,030,415	549,315	208,445
HEK293	CCNT1	RIP-seq	wildtype	GSM1057803	Yes	59	775,582	737,905	53,950	5,853
HEK293	GFP	RIP-seq	control	GSM1057804	Yes	59	773,785	717,692	55,062	4,556
HEK293	CCNT1	RIP-seq	wildtype	GSM1057805	No	128	1,647,641	674,583	39,300	26,859
HEK293	GFP	RIP-seq	control	GSM1057806	No	128	2,369,271	1,912,105	2,658,747	45,024
GM12878	ELAVL1	RIP-seq	Biorep1	SRR504447	No	36	37,263,588	32,601,518	32,528,042	13,261,848
GM12878	ELAVL1	RIP-seq	Biorep2	SRR504448	No	36	37,414,489	31,912,714	31,601,964	6,132,389
GM12878	PABPC1	RIP-seq	Biorep1	SRR504445	No	36	34,885,904	33,458,725	39,368,686	8,351,529
GM12878	PABPC1	RIP-seq	Biorep2	SRR504446	No	36	33,969,717	32,636,482	38,209,503	4,549,408
GM12878	RIP input	RIP-seq	Biorep1	SRR504457	No	36	36,111,283	12,778,363	12,057,948	5,506,941
GM12878	RIP input	RIP-seq	Biorep2	SRR504458	No	36	32,508,482	8,051,545	6,505,188	2,913,744
GM12878	T7Tag	RIP-seq	Biorep1	SRR504455	No	36	35,560,522	18,710,540	18,268,309	5,735,953
GM12878	T7Tag	RIP-seq	Biorep2	SRR504456	No	36	20,625,454	10,870,584	10,885,726	5,784,151
K562	ELAVL1	RIP-seq	Biorep1	SRR504453	No	36	24,253,488	22,326,300	22,239,698	6,161,580
K562	ELAVL1	RIP-seq	Biorep2	SRR504454	No	36	24,201,745	21,184,619	20,988,413	4,685,302
K562	PABPC1	RIP-seq	Biorep1	SRR504451	No	36	21,970,069	20,899,928	28,123,613	3,440,796
K562	PABPC1	RIP-seq	Biorep2	SRR504452	No	36	21,951,428	20,993,334	26,887,104	5,973,108
K562	RIP input	RIP-seq	Biorep1	SRR504449	No	36	20,351,980	7,766,898	7,102,232	4,529,832
K562	RIP input	RIP-seq	Biorep2	SRR504450	No	36	24,019,308	7,288,617	5,439,098	1,348,233
K562	T7Tag	RIP-seq	Biorep1	SRR504459	No	36	22,795,411	12,112,270	12,743,129	4,914,470
K562	T7Tag	RIP-seq	Biorep2	SRR504460	No	36	28,776,896	24,358,117	24,355,967	6,493,841
HEK293	PUM2	PAR-CLIP	TechRep1	SRR048967	No	32	5,104,559	3,471,160	500,288	500,288
HEK293	PUM2	PAR-CLIP	TechRep2	SRR048968	No	32	5,351,797	2,162,508	385,669	385,669
HEK293	QKI	PAR-CLIP	TechRep1	SRR048969	No	32	3,682,206	2,441,584	132,334	132,334
HEK293	QKI	PAR-CLIP	TechRep2	SRR048970	No	32	2,845,295	1,919,805	110,904	110,904
HEK293	QKI	PAR-CLIP	TechRep3	SRR048971	No	32	5,402,812	2,843,006	106,222	106,222
HEK293	QKI	PAR-CLIP	TechRep4	SRR048972	No	32	5,023,532	2,433,413	15,743	15,743

Technical replicates were pooled after the alignments and subject to preprocessing. Please refer to **Quality Control of Raw Read Library** and **Alignment of Filtered RIP-seq Read Library to Reference Genome** for preprocessing and alignment procedures applied on each datasets. The RIP-seq data for CCNT1 and GFP control were generated in-house and deposited in GEO subsequently.

Table S2. Top 10 Molecular Function (MF) GO terms enriched for genes associated with RIPSeeker predictions on - strand of the PRC2 Biorep1 dataset.

go.id	go.term	Definition	Ontology	pvalue	count In Dataset	count In Genome	total term In Dataset	total term In Genome	BH adjusted pvalue
GO:0(	protein binding	Interacting selectively and non-covalently with any protein or protein complex (a complex of two or more proteins that may include other nonprotein molecules).	MF	4.46E-21	2594	5788	68028	174997	1.17E-17
GO:0(	nucleic acid binding	Interacting selectively and non-covalently with any nucleic acid.	MF	3.85E-16	1197	2565	68028	174997	5.04E-13
GO:0(	enzyme binding	Interacting selectively and non-covalently with any enzyme.	MF	1.02E-14	532	1053	68028	174997	8.93E-12
GO:0(	RNA binding	Interacting selectively and non-covalently with an RNA molecule or a portion thereof.	MF	1.92E-14	392	745	68028	174997	1.25E-11
GO:0(	nucleotide binding	Interacting selectively and non-covalently with a nucleotide, any compound consisting of a nucleoside that is esterified with (ortho)phosphate or an oligophosphate at any hydroxyl group on the ribose or deoxyribose.	MF	2.46E-13	1027	2213	68028	174997	1.29E-10
GO:0(	binding	The selective, non-covalent, often stoichiometric, interaction of a molecule with one or more specific sites on another molecule.	MF	4.28E-13	4580	10871	68028	174997	1.60E-10
GO:0(	small molecule binding	Interacting selectively and non-covalently with a small molecule, any low molecular weight, monomeric, non-encoded molecule.	MF	4.29E-13	1094	2375	68028	174997	1.60E-10
GO:0(	purine ribonucleoside triphosphate binding	Interacting selectively and non-covalently with a purine ribonucleoside triphosphate, a compound consisting of a purine base linked to a ribose sugar esterified with triphosphate on the sugar.	MF	1.86E-10	801	1731	68028	174997	5.43E-08
GO:0(	purine ribonucleotide binding	Interacting selectively and non-covalently with a purine ribonucleotide, any compound consisting of a purine ribonucleoside that is esterified with (ortho)phosphate or an oligophosphate at any hydroxyl group on the ribose moiety.	MF	1.87E-10	815	1764	68028	174997	5.43E-08
GO:0(	ribonucleotide binding	Interacting selectively and non-covalently with a ribonucleotide, any compound consisting of a ribonucleoside that is esterified with (ortho)phosphate or an oligophosphate at any hydroxyl group on the ribose moiety.	MF	2.13E-10	815	1765	68028	174997	5.58E-08

GO enrichment was performed using RIPSeeker built-in function `annotateRIP`. Only Biorep1 is displayed. Consistent with the biology, the top 10 Molecular Function (MF) GO terms enriched for genes overlapping or adjacent to the RIPSeeker predictions on both PRC2 datasets are mostly related to protein binding (adjusted p-value $<$ 1.17E-17) and nucleotide binding (adjusted p-value $<$ 1.29E-10). These genes involved in these processes thus represent a putative list of RNA transcripts bound by PRC2.

Table S3. Top 10 Biological Process (BP) GO terms enriched for genes associated with RIPSeeker predictions on the + strand of the CCNT1 second screen dataset.

go.id	go.term	Definition	Ontology	pvalue	count In Dataset	count In Genome	total term In Dataset	total term In Genome	BH adjusted pvalue
GO:0(	viral genome expression	The achievement of highly specific, quantitative, temporal and spatial control of virus gene expression within the limited genetic resources of the viral genome.	BP	9.85E-07	9	151	5378	805639	0.0005904
GO:0(	viral transcription	The mechanisms involved in viral gene transcription, especially referring to those with temporal properties unique to viral transcription.	BP	9.85E-07	9	151	5378	805639	0.0005904
GO:0(	nuclear-transcribed mRNA catabolic process, nonsense-mediated decay	The nonsense-mediated decay pathway for nuclear-transcribed mRNAs degrades mRNAs in which an amino-acid codon has changed to a nonsense codon; this prevents the translation of such mRNAs into truncated, and potentially harmful, proteins.	BP	1.60E-06	8	119	5378	805639	0.0005904
GO:0(	viral reproductive process	A reproductive process involved in viral reproduction. Usually, this is by infection of a host cell, replication of the viral genome, and assembly of progeny virus particles. In some cases the viral genetic material may integrate into the host genome and only subsequently, under particular circumstances, 'complete' its life cycle.	BP	1.63E-06	11	256	5378	805639	0.0005904
GO:0(	translational termination	The process resulting in the release of a polypeptide chain from the ribosome, usually in response to a termination codon (UAA, UAG, or UGA in the universal genetic code).	BP	3.38E-06	7	93	5378	805639	0.0009801
GO:0(	viral infectious cycle	A set of processes which all viruses follow to ensure survival; includes attachment and entry of the virus particle, decoding of genome information, translation of viral mRNA by host ribosomes, genome replication, and assembly and release of viral particles containing the genome.	BP	4.16E-06	10	229	5378	805639	0.0010051
GO:0(	translational elongation	The successive addition of amino acid residues to a nascent polypeptide chain during protein biosynthesis.	BP	7.57E-06	7	105	5378	805639	0.0010955
GO:0(	SRP-dependent cotranslational protein targeting to membrane	The targeting of proteins to a membrane that occurs during translation and is dependent upon two key components, the signal-recognition particle (SRP) and the SRP receptor. SRP is a cytosolic particle that transiently binds to the endoplasmic reticulum (ER) signal sequence in a nascent protein, to the large ribosomal unit, and to the SRP receptor in the ER membrane.	BP	7.57E-06	7	105	5378	805639	0.0010955
GO:0(	cotranslational protein targeting to membrane	The targeting of proteins to a membrane that occurs during translation. The transport of most secretory proteins, particularly those with more than 100 amino acids, into the endoplasmic reticulum lumen occurs in this manner, as does the import of some proteins into mitochondria.	BP	8.57E-06	7	107	5378	805639	0.0010955
GO:0(	protein targeting to ER	The process of directing proteins towards the endoplasmic reticulum (ER) using signals contained within the protein. One common mechanism uses a 16- to 30-residue signal sequence, typically located at the N-terminus of the protein and containing positively charged amino acids followed by a continuous stretch of hydrophobic residues, which directs the ribosome to the ER membrane and initiates transport of the growing polypeptide across the ER membrane.	BP	8.57E-06	7	107	5378	805639	0.0010955

Intriguingly, the most enriched Biological Process GO term on the second screen of CCNT1 datasets is viral genome expression followed by viral transcription, viral reproductive process, and virial infectious cycle. Indeed, CCNT1 is known to interact with nascent TAR HIV RNA, which competes with the endogenous lncRNA *RN7SK* for the control over RNAPII in viral specific transcription elongation (see **Introduction** in the main text). However, we cannot exclude the possibility that such enrichment of viral functions were caused by the usage of viral vector in delivering the tag to the CCNT1 gene.

SUPPLEMENTARY FIGURES

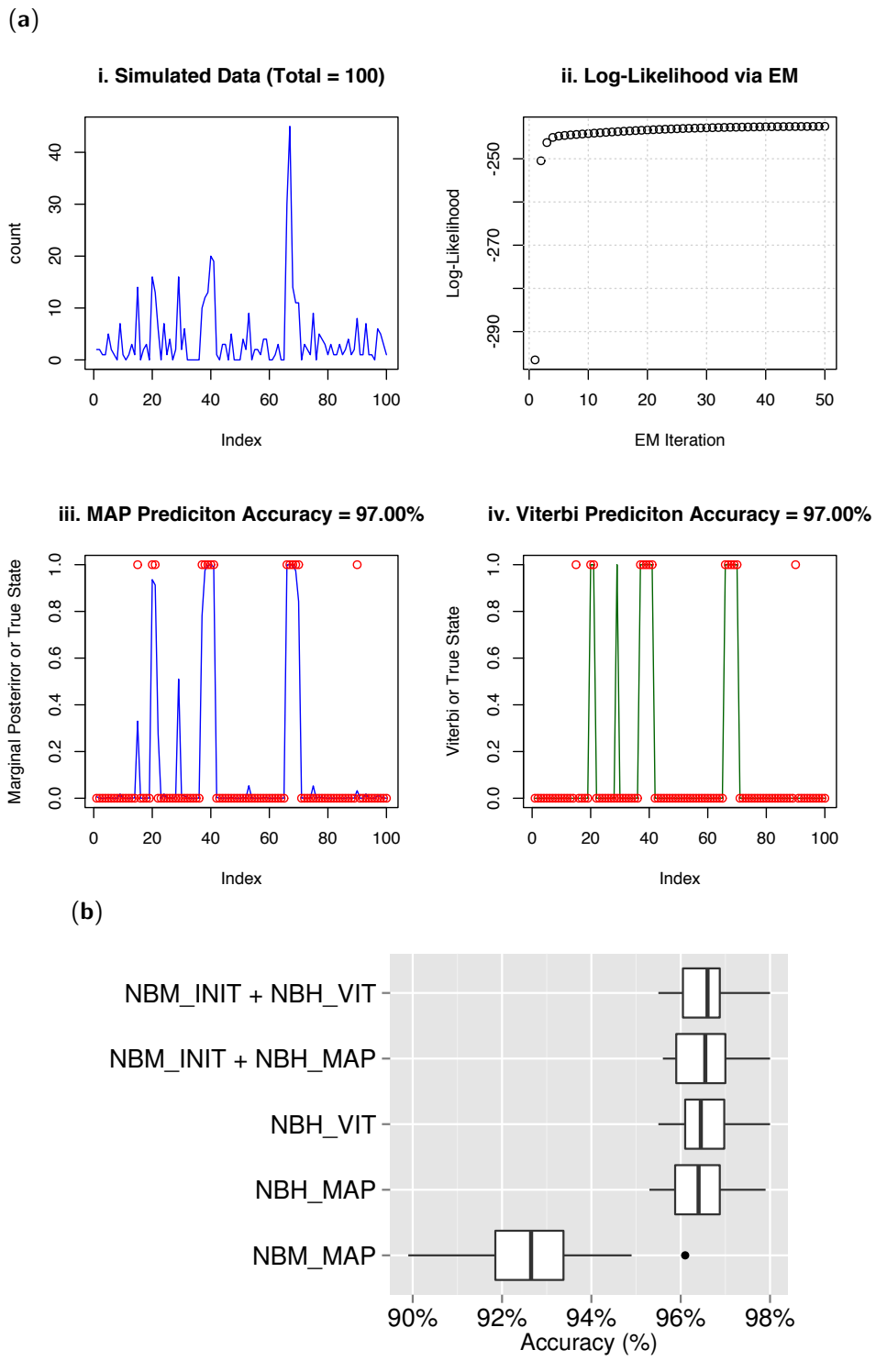


Figure S1. (**a**) Simulation test on RIPSeeker predictions. HMM predicts well on the (**i**) 100 simulated data points that are sampled from negative binomial mixture distribution of two components by following a arbitrary set of HMM parameters ($\alpha=\{2,4\},\beta=\{1,0.25\},A=\{0,9,0.1;0.3,0.7\}$); (**ii**) EM converges steadily and efficiently; (**iii** and (**iv**)) Both maximum a posterior (MAP) and Viterbi (i.e. ML) produces over 95% accuracy and around 85% for large number of data points (>5000) for both training and testing. (**b**) Comparison in terms prediction accuracy on simulated data among negative binomial mixture model (NBM) with *maximum a posterior* (MAP), negative binomial hidden Markov model (NBH) with MAP (NBH_MAP) and Viterbi (NBH_VIT), and NBM initialization (NBM_INIT) followed by NBH_VIT (NBM_INIT + NBH_VIT) or NBH_MAP (NBM_INIT + NBH_MAP). One thousand data points of discrete counts were simulated using the same model described in S1. The test was repeated 10 times. As shown, NBM_INIT + NBH_VIT achieves the best performance among all.

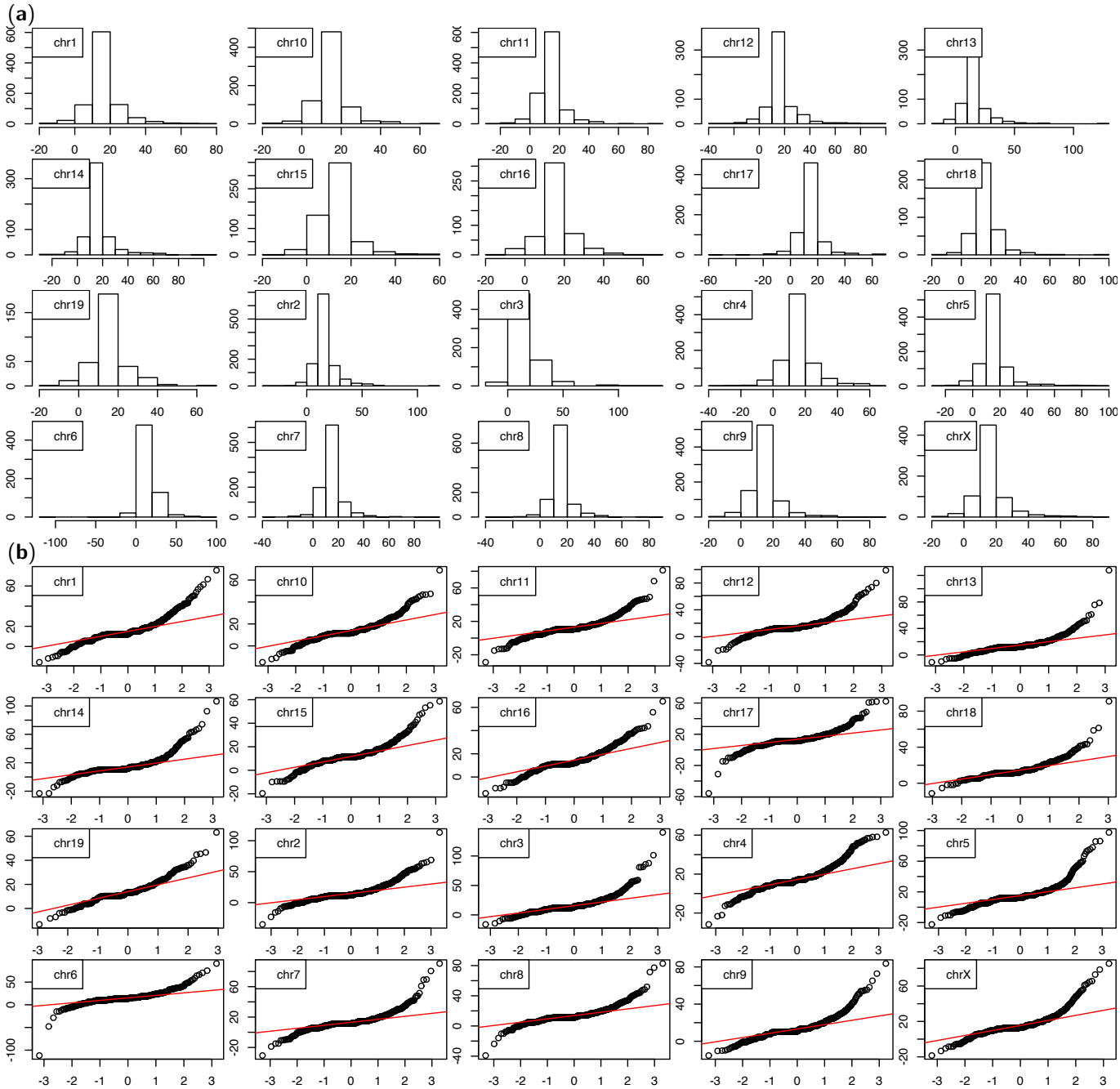
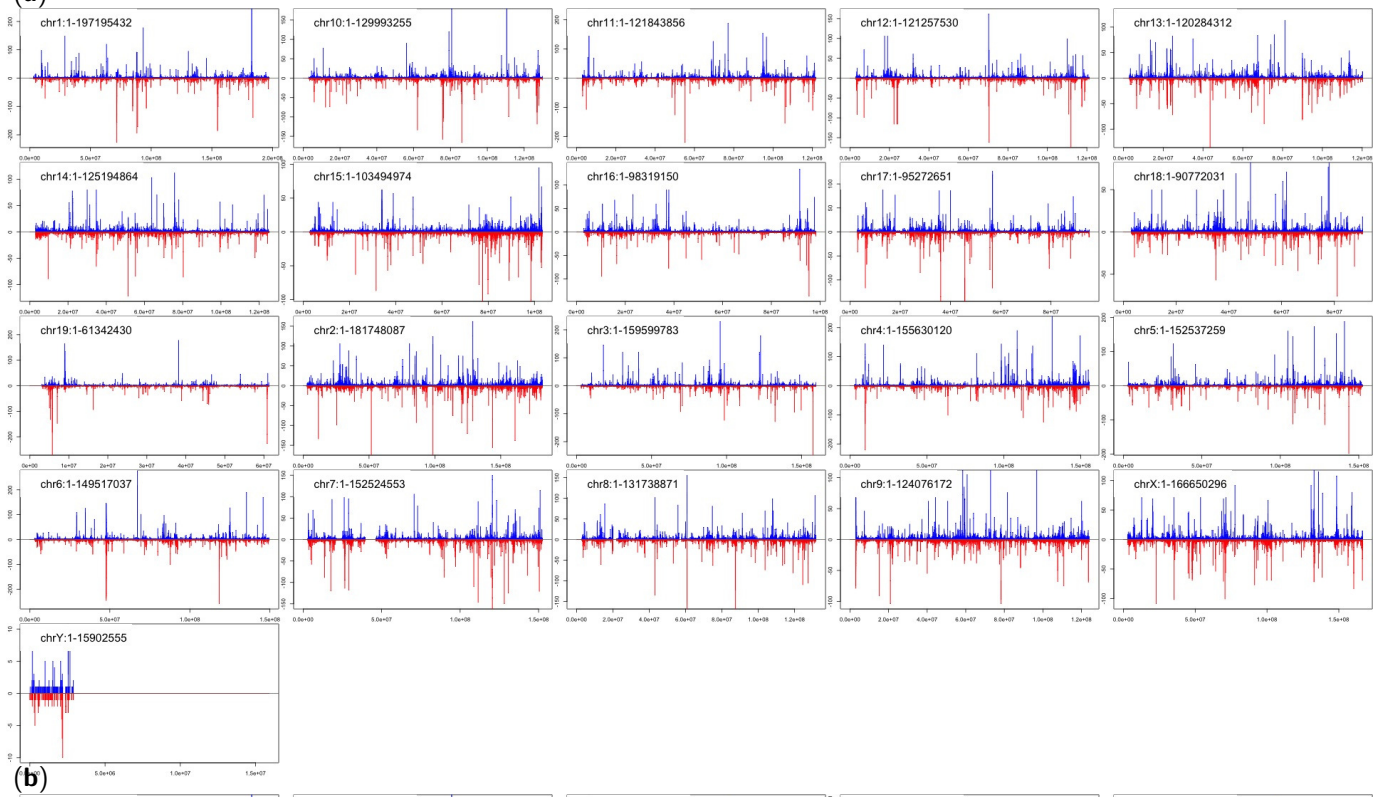
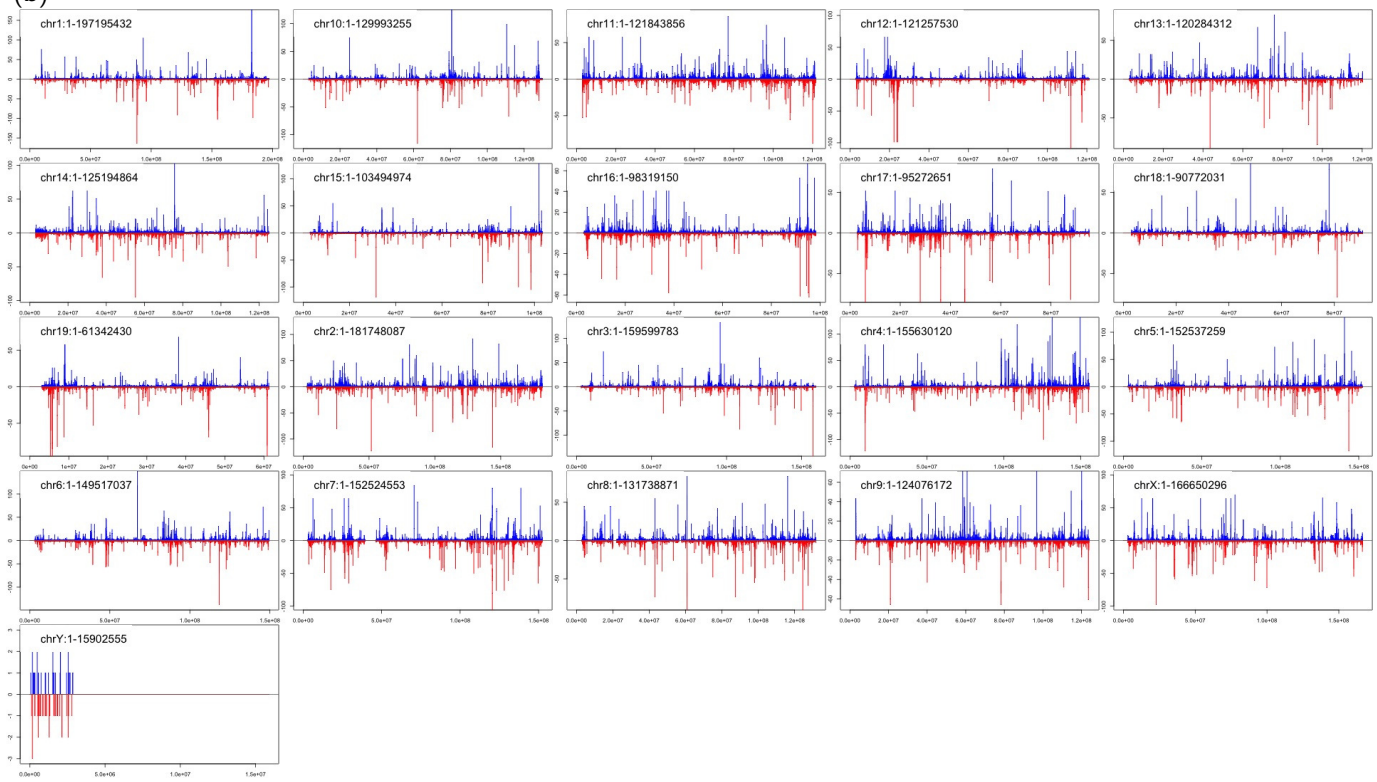


Figure S2. RIPScore (Equation **15** in the main text) approximately follows normal distribution based on (**a**) histogram and (**b**) quantile-quantile (QQ) plot. RIPScores per chromosome for PRC2 RIP-seq dataset are shown to reflect the peak calling process in RIPSeeker on per-chromosome basis. Distribution of RIPScores for other RIP-seq data are similar (not shown).

(**a**)


chr1:1-197195432
chr10:1-129993255
chr11:1-121843856
chr12:1-121257530
chr13:1-120284312
chr14:1-125194864
chr15:1-103494974
chr16:1-98319150
chr17:1-95272651
chr18:1-90772031
chr19:1-61342430
chr2:1-181748087
chr3:1-159599783
chr4:1-155630120
chr5:1-152537259
chr6:1-149517037
chr7:1-152524553
chr8:1-131738871
chr9:1-124076172
chrX:1-166650296
chrY:1-15902555

(**b**)


chr1:1-197195432
chr10:1-129993255
chr11:1-121843856
chr12:1-121257530
chr13:1-120284312
chr14:1-125194864
chr15:1-103494974
chr16:1-98319150
chr17:1-95272651
chr18:1-90772031
chr19:1-61342430
chr2:1-181748087
chr3:1-159599783
chr4:1-155630120
chr5:1-152537259
chr6:1-149517037
chr7:1-152524553
chr8:1-131738871
chr9:1-124076172
chrX:1-166650296
chrY:1-15902555

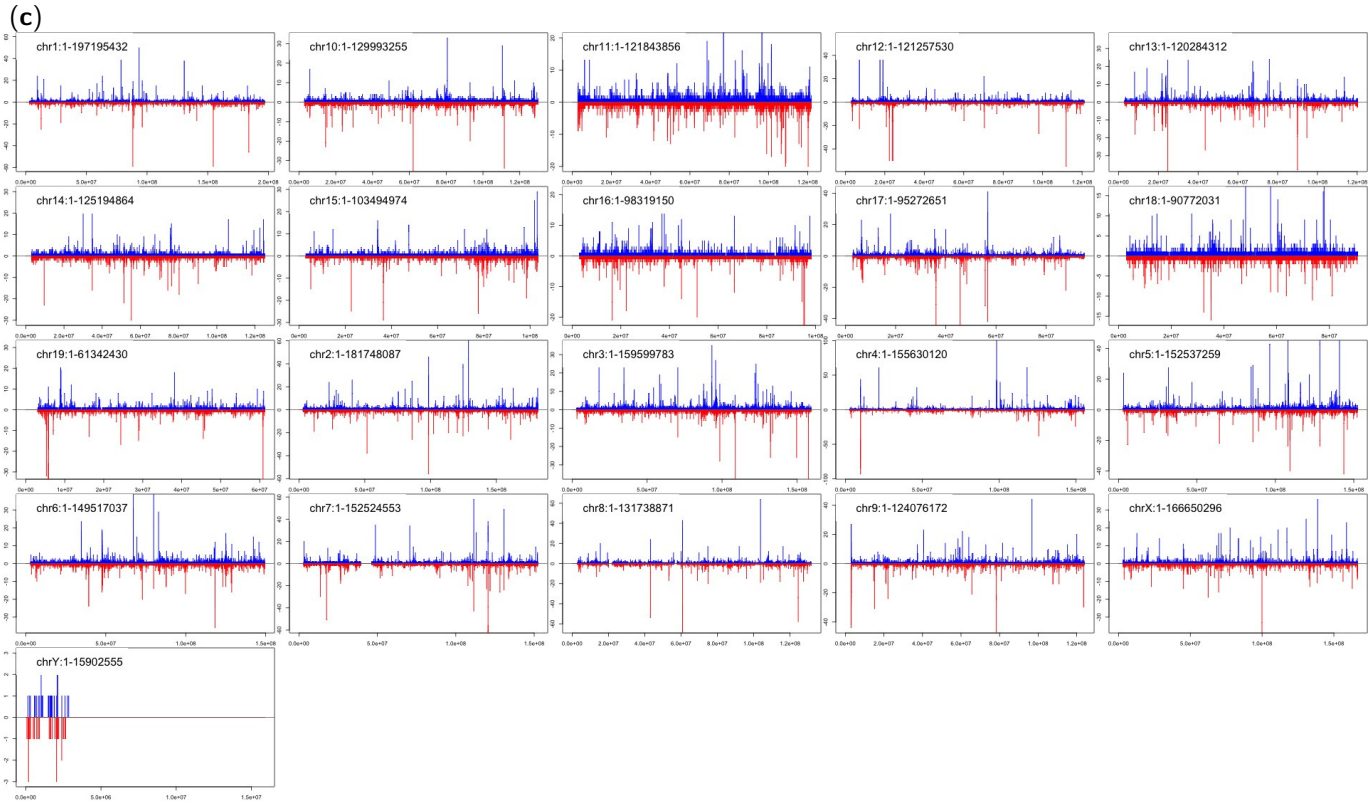
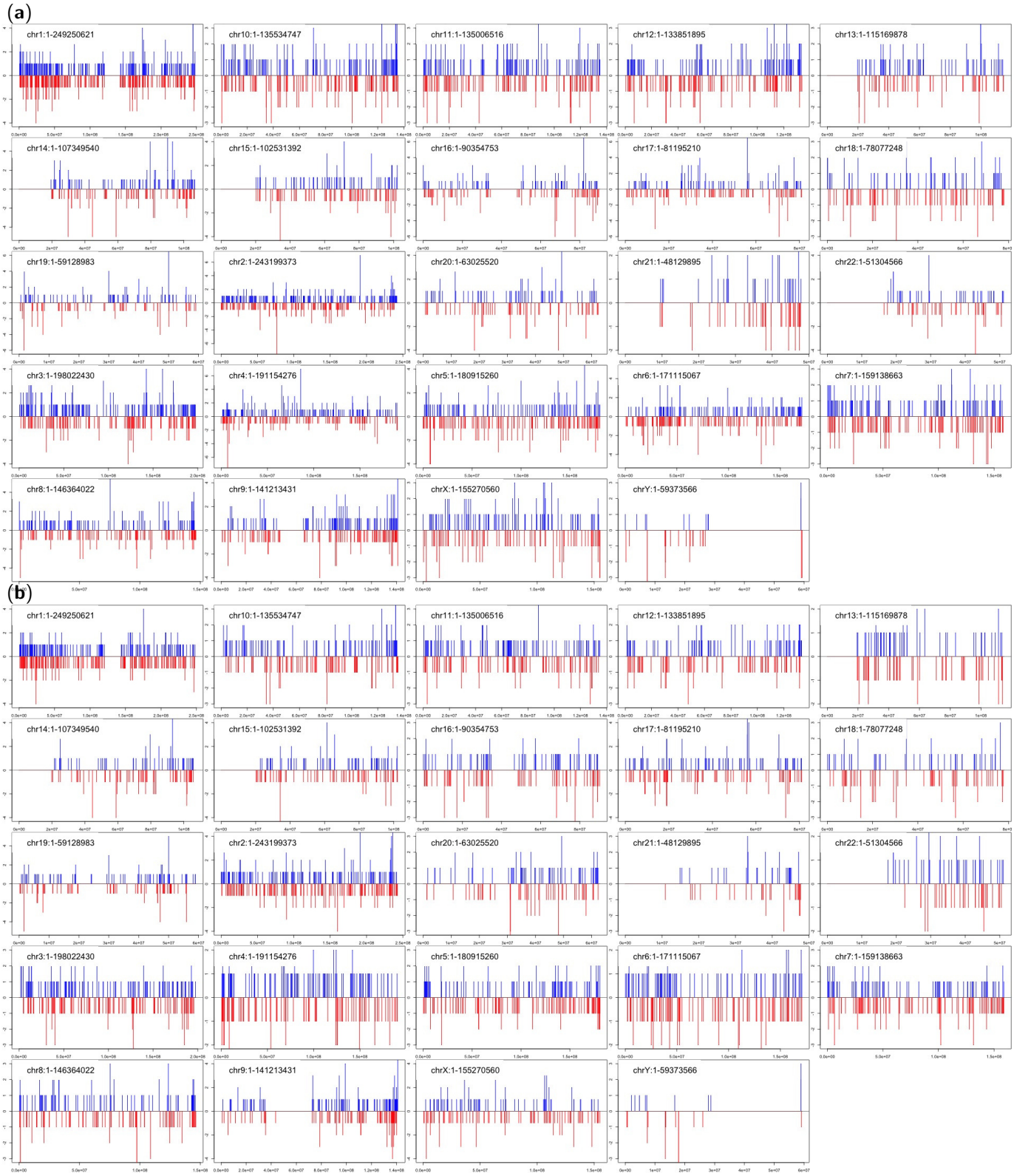


Figure S3. Read coverage of distinct alignments (after filtering duplicate alignments) across each chromosome for (**a**) PRC2 wild type biological replicate (Biorep) 1, (**b**) wild type Biorep 2, and (**c**) Ezh2 -/- mutant control. Read count on + and - strand are displayed as blue and red bars on the positive and negative y-axis, respectively. Read counts are drawn to scale within a chromosome. As expected, no symmetry is observed for the peak for the strand-specific sequencing data. However, considerable noise is observed within the Ezh2 -/- mutant library, which ideally should not have any aligned reads. This may also imply considerable noise within the wild type library and thus a high false discovery rate if the RIP regions were simply determined based on read counts. Biorep 1 is the pooled alignments of the two technical replicates. The plot was generated by `plotStrandedCoverage` from RIPSeeker with bin size fixed to 1 kb for all of the chromosomes. Chromosome with no alignment is omitted (e.g., chrM).


(a)
chr1:1-249250621
chr10:1-135534747
chr11:1-135006516
chr12:1-133851895
chr13:1-115169878
chr14:1-107349540
chr15:1-102531392
chr16:1-90354753
chr17:1-81195210
chr18:1-78077248
chr19:1-59128983
chr2:1-243199373
chr20:1-63025520
chr21:1-48129895
chr22:1-51304566
chr3:1-198022430
chr4:1-191154276
chr5:1-180915260
chr6:1-171115067
chr7:1-159138663
chr8:1-146364022
chr9:1-141213431
chrX:1-155270560
chrY:1-59373566
(b)
chr1:1-249250621
chr10:1-135534747
chr11:1-135006516
chr12:1-133851895
chr13:1-115169878
chr14:1-107349540
chr15:1-102531392
chr16:1-90354753
chr17:1-81195210
chr18:1-78077248
chr19:1-59128983
chr2:1-243199373
chr20:1-63025520
chr21:1-48129895
chr22:1-51304566
chr3:1-198022430
chr4:1-191154276
chr5:1-180915260
chr6:1-171115067
chr7:1-159138663
chr8:1-146364022
chr9:1-141213431
chrX:1-155270560
chrY:1-59373566

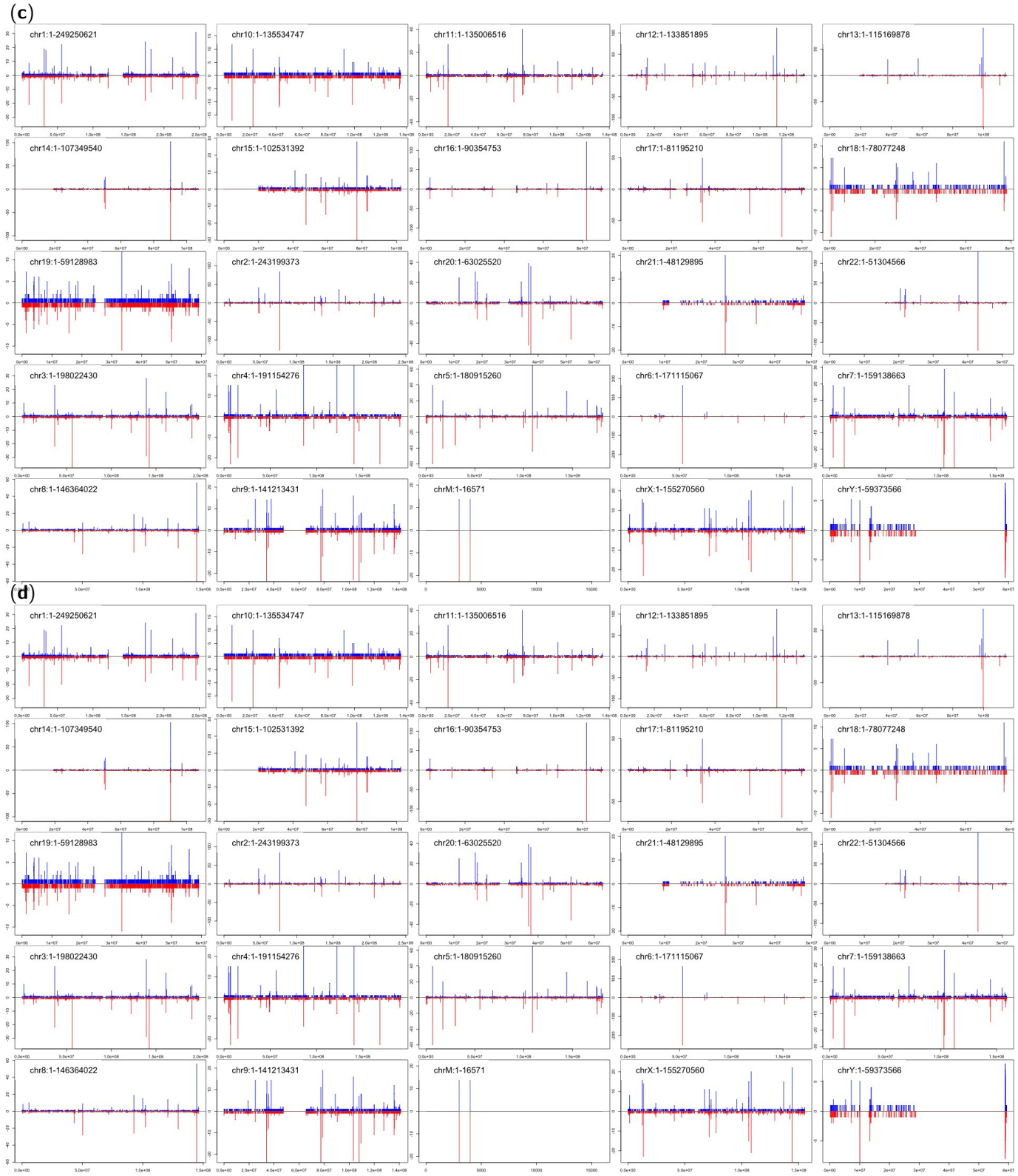


Figure S4. Read coverage of distinct alignments (after filtering duplicate alignments) across each chromosome for the strand-specific sequencing data for (**a**) CCNT1 and (**b**) GFP as well as non-strand-specific sequencing data of (**c**) CCNT1 and (**d**) GFP. All datasets were generated in-house. Much stronger symmetry of peaks is observed in the non-strand-specific comparing to strand-specific data. Also, noise is considerably high as implicated by both GFP datasets. The plot was generated by `plotStrandedCoverage` from RIPSeeker with bin size fixed to 1 kb for all of the chromosomes. Chromosomes with no alignment are omitted.

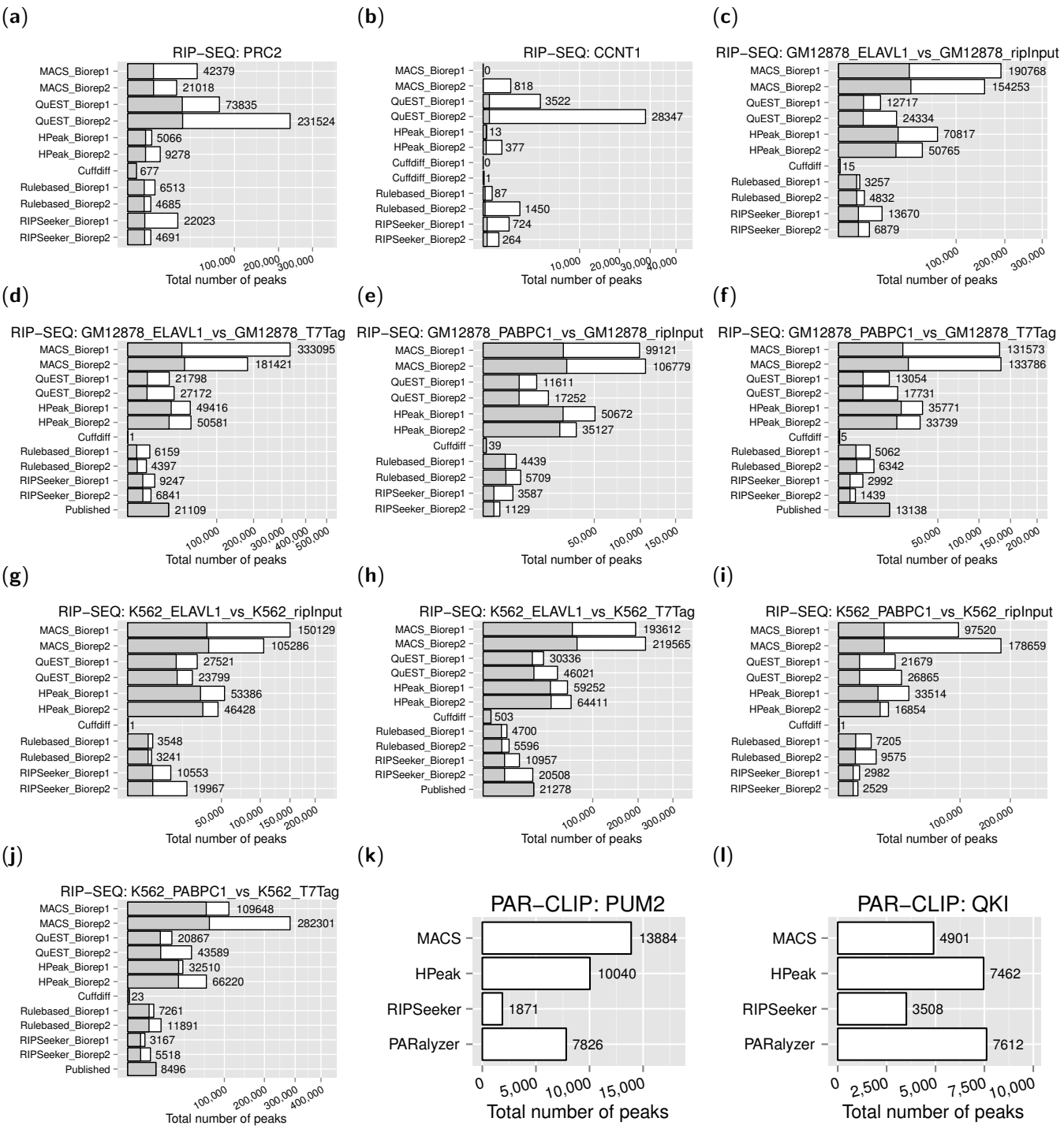


Figure S5. Total peaks predicted by each comparison method on RIP-Seq and PAR-CLIP data. (**a**) PRC2 in mES using mutant Ezh-/- as control (13); (**b**) CCNT1 in HEK293 using GFP as control (in-house); (**c**) ELAVL1 in GM12878 using RIP input RNA as control; (**d**) ELAVL1 in GM12878 using T7Tag as control; (**e**) PABPC1 in GM12878 using RIP input RNA as control; (**f**) PABPC1 in GM12878 using T7Tag as control; (**g**-**j**) correspond to the same proteins and control as in (**c**-**f**) respectively except performed in K562 cell line; (**k**, **l**) PAR-CLIP data for PUM2 and QKI in HEK293, respectively (14). (**c**) to (**j**) correspond to the RIP-seq data generated by Dr. Scott Tenenbaum laboratory at the ENCODE Consortium (15). For technical replicates, reads were pooled and subject to peak calling by each method. For biological replicate, all of the methods except for "Cuffdiff" and "Published" are applied to each biological replicate separately. The grey colour on the white bar indicate the peaks shared in common by both Biorep1 and 2. "Cuffdiff" and "Published" employ statistical hypothesis testing on both biological replicates to estimate the sample variance for each potential peak. "Published" represents the peaks analyzed by Dr. Tenenbaum group and were downloaded from UCSC Genome Browser (`http://genome.ucsc.edu/cgi-bin/hgTrackUi?db=hg19&g=wgEncodeSunyRipSeq`). Please refer to Table S1 for basic statistics about the data. Since CCNT1 (**b**) is only known to interact with *RN7SK*, the low concordance between biological replicates for each method perhaps implies that most peaks identified outside of the positive target loci are due to background noise.

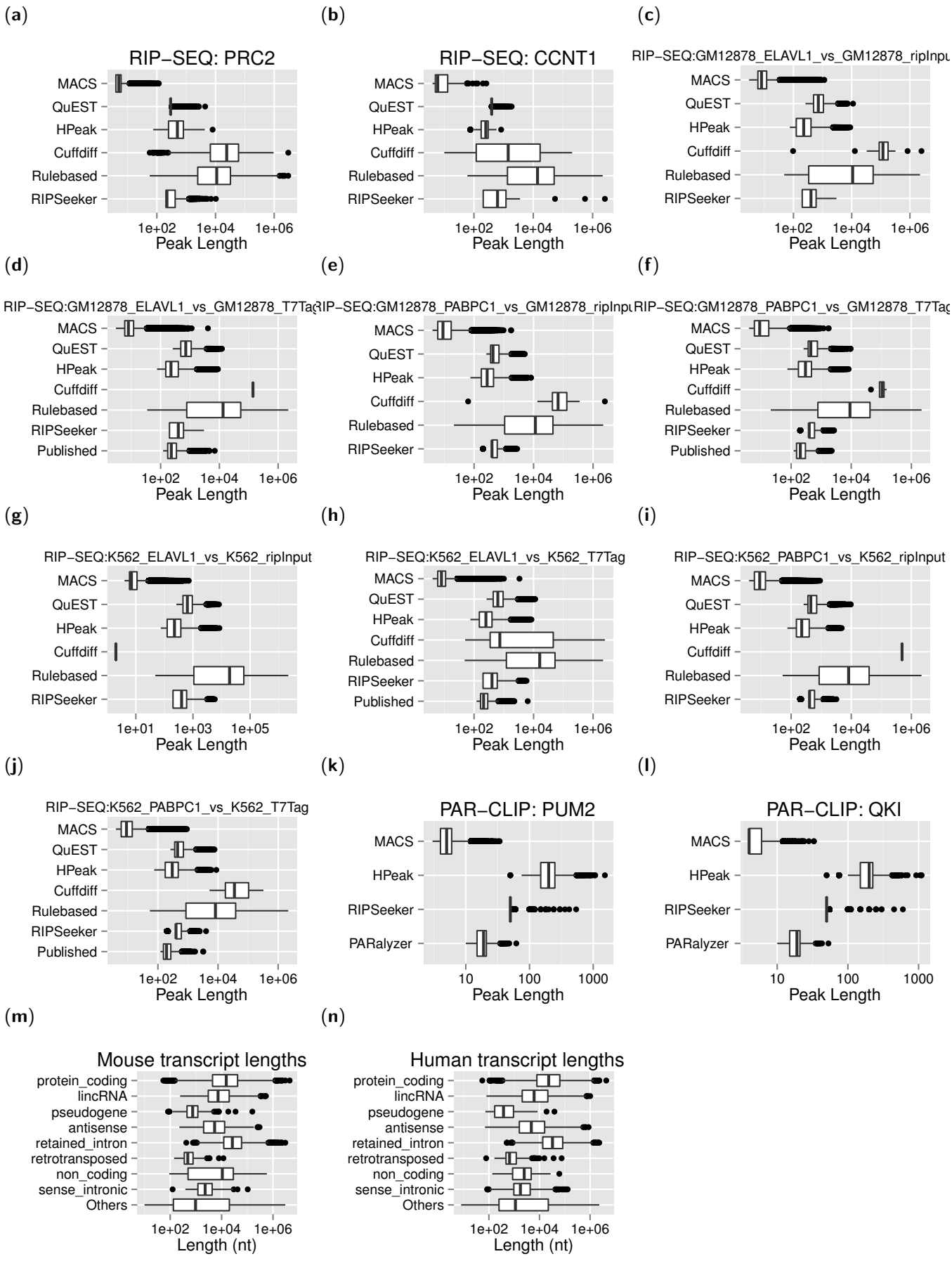

(a)
RIP−SEQ: PRC2
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
1e+02 1e+04 1e+06
Peak Length
(b)
RIP−SEQ: CCNT1
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
1e+02 1e+04 1e+06
Peak Length
(c)
RIP−SEQ:GM12878_ELAVL1_vs_GM12878_ripInpu
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
1e+02 1e+04 1e+06
Peak Length
(d)
RIP−SEQ:GM12878_ELAVL1_vs_GM12878_T7Tag
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
Published
1e+02 1e+04 1e+06
Peak Length
(e)
RIP−SEQ:GM12878_PABPC1_vs_GM12878_ripInpu
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
1e+02 1e+04 1e+06
Peak Length
(f)
RIP−SEQ:GM12878_PABPC1_vs_GM12878_T7Tag
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
Published
1e+02 1e+04 1e+06
Peak Length
(g)
RIP−SEQ:K562_ELAVL1_vs_K562_ripInput
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
1e+01 1e+03 1e+05
Peak Length
(h)
RIP−SEQ:K562_ELAVL1_vs_K562_T7Tag
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
Published
1e+02 1e+04 1e+06
Peak Length
(i)
RIP−SEQ:K562_PABPC1_vs_K562_ripInput
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
1e+02 1e+04 1e+06
Peak Length
(j)
RIP−SEQ:K562_PABPC1_vs_K562_T7Tag
MACS
QuEST
HPeak
Cuffdiff
Rulebased
RIPSeeker
Published
1e+02 1e+04 1e+06
Peak Length
(k)
PAR−CLIP: PUM2
MACS
HPeak
RIPSeeker
PARalyzer
10 100 1000
Peak Length
(l)
PAR−CLIP: QKI
MACS
HPeak
RIPSeeker
PARalyzer
10 100 1000
Peak Length
(m)
Mouse transcript lengths
protein_coding
lincRNA
pseudogene
antisense
retained_intron
retrotransposed
non_coding
sense_intronic
Others
1e+02 1e+04 1e+06
Length (nt)
(n)
Human transcript lengths
protein_coding
lincRNA
pseudogene
antisense
retained_intron
retrotransposed
non_coding
sense_intronic
Others
1e+02 1e+04 1e+06
Length (nt)

Figure S6. Lengths of peaks predicted by each comparison method on RIP-Seq and PAR-CLIP data. Peaks predicted from biological replicates are pooled. For (**a**)-(**j**), the peak lengths are draw at the log10 scale. Please refer to Figure S5 for details on each subfigure. As reference for comparison, transcript lengths from (**m**) mouse and (**n**) human based on Ensembl 65 and 69, respectively, are included.

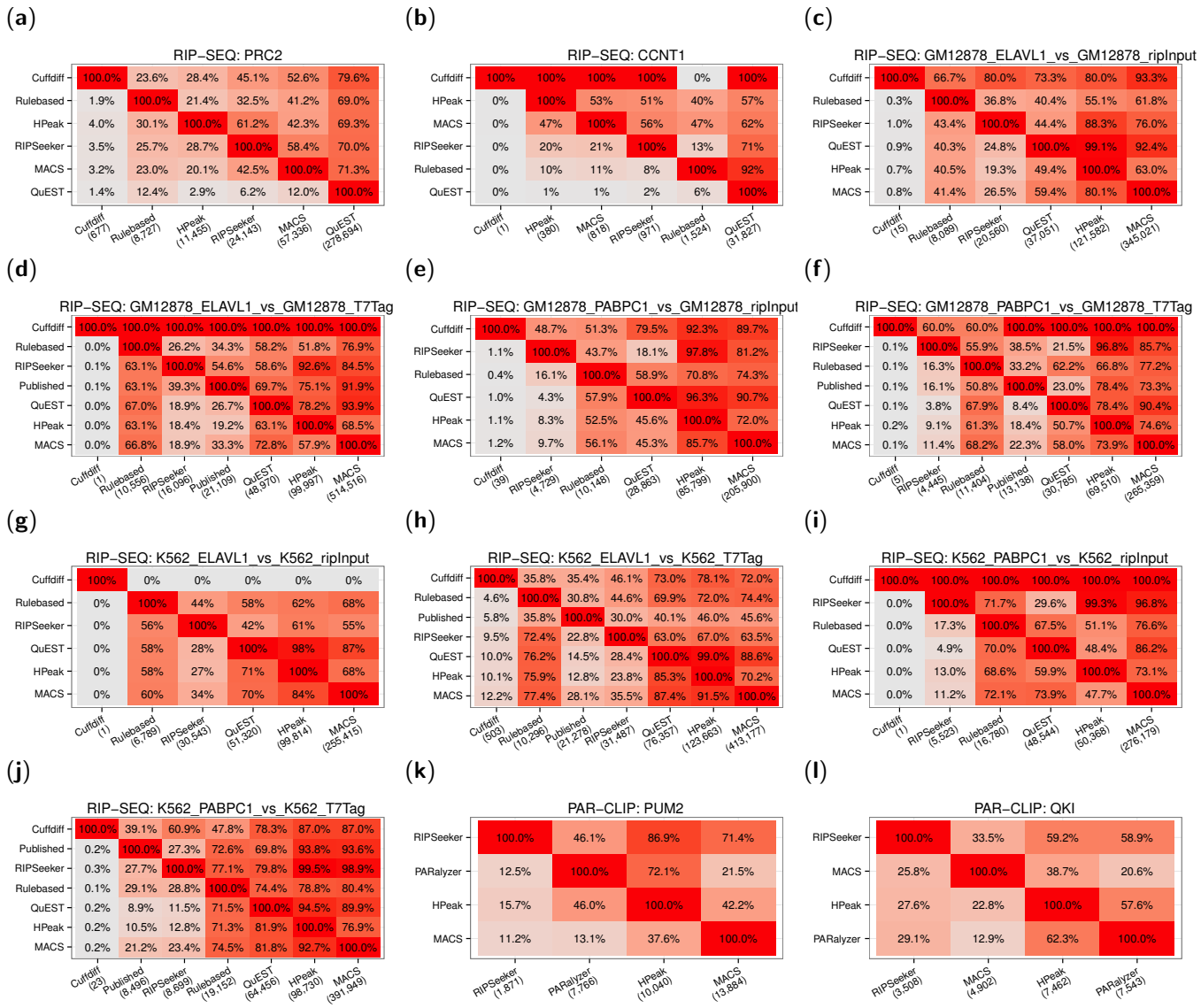
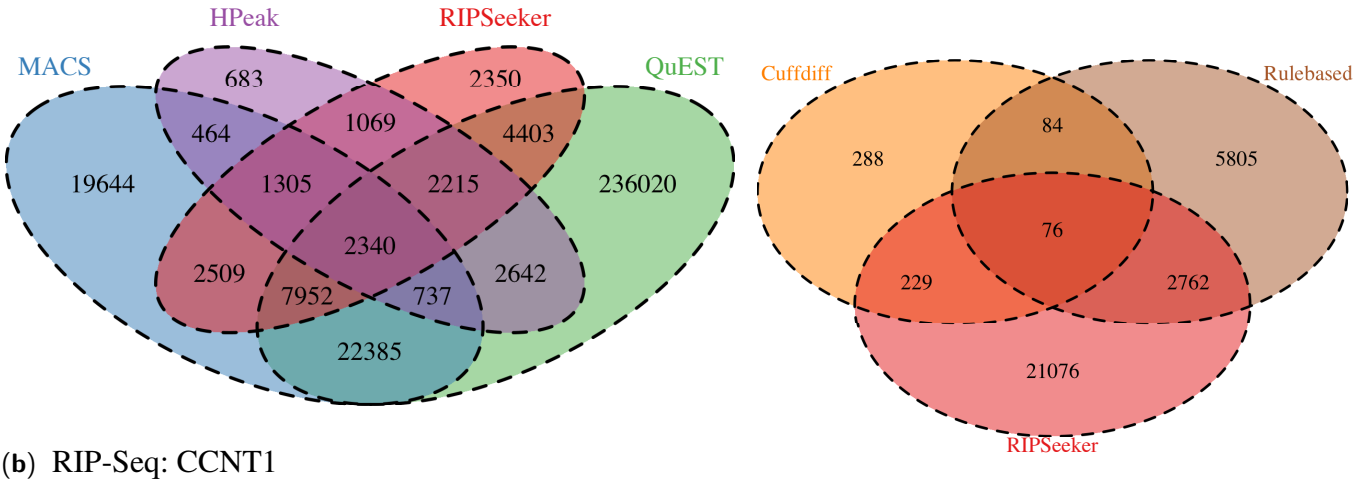


Figure S7. Pair-wise comparison of shared peaks between methods on all RIP-Seq and PAR-CLIP test data. Each panel shows the percentage of total peaks from methods on the y-axis that overlap with the peaks from the methods on the x-axis. Notably, the percentage matrices are not symmetrical because a number of peaks in method A may overlap with a different number of peaks in method B and vice versa. For instance, in panel (**a**), 70.0% of the RIPSeeker peaks (row 4, column 6) overlap with 6.2% of the QuEST peaks (row 6, column 4). Programs in both axes are sorted by increasing number of peaks and entries are shaded by color gradients such that red represents the highest shared proportion and grey, the lowest. Please refer to Figure S5 legend for details on the test data used in each of the 12 subfigures.

(**a**) RIP-Seq: PRC2



(**b**) RIP-Seq: CCNT1

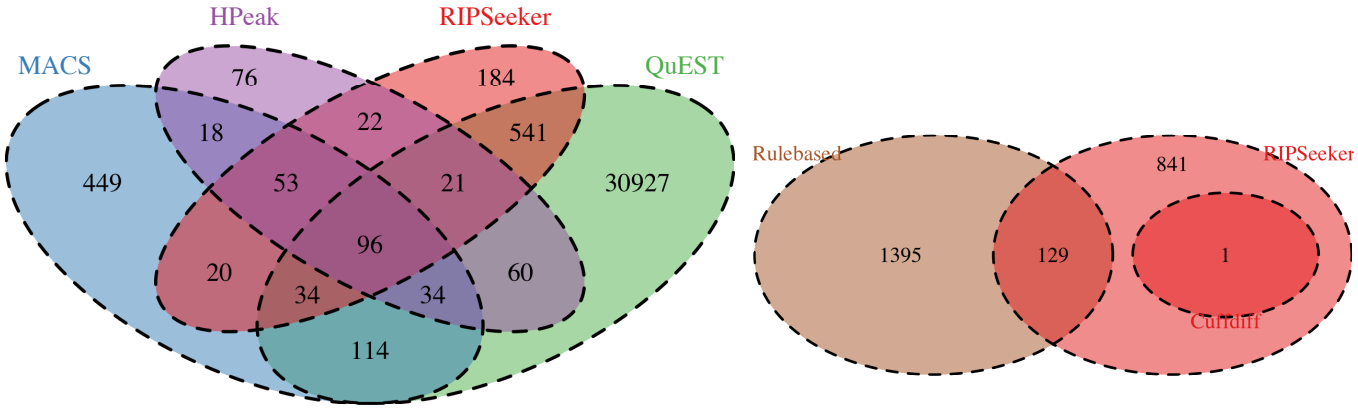
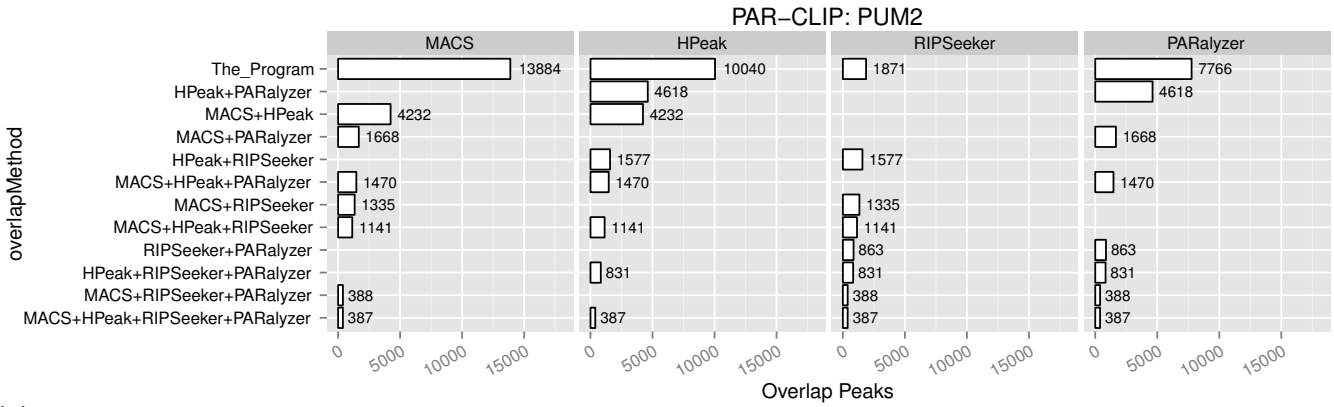


Figure S8. Venn diagram of RIPSeeker predictions for (**a**) PRC2 and (**b**) CCNT1 comparing with predictions from the three peak callers MACS, HPeak, and QuEST and the two transcript-based methods Cuffdiff and Rulebased. N peaks from one program may overlap with M peaks from another, where N is not necessarily equal to M. To resolve the ambiguity, the number of overlapping peaks is defined as the lower number (i.e. $\min(N,M)$). Similarly, for 3-way and 4-way overlap, the number of overlapping peaks among the 3 and 4 programs respectively is the lowest number of overlapping peaks from one of the programs. Such assignment will prevent some but not all of the cases where the same peaks from one program overlap with multiple peaks from one or more other programs, resulting in negative number of remaining peaks after subtracting the overlapping peaks from the total peaks. A more robust representation is via barplot as illustrated in Figure S9 for the PAR-CLIP data and Figure S10 for the ENCODE data (for examining robustness of each method), for which the Venn diagram cannot be solved without introducing negative values for the remaining non-overlapping peaks.

(**a**)



(**b**)

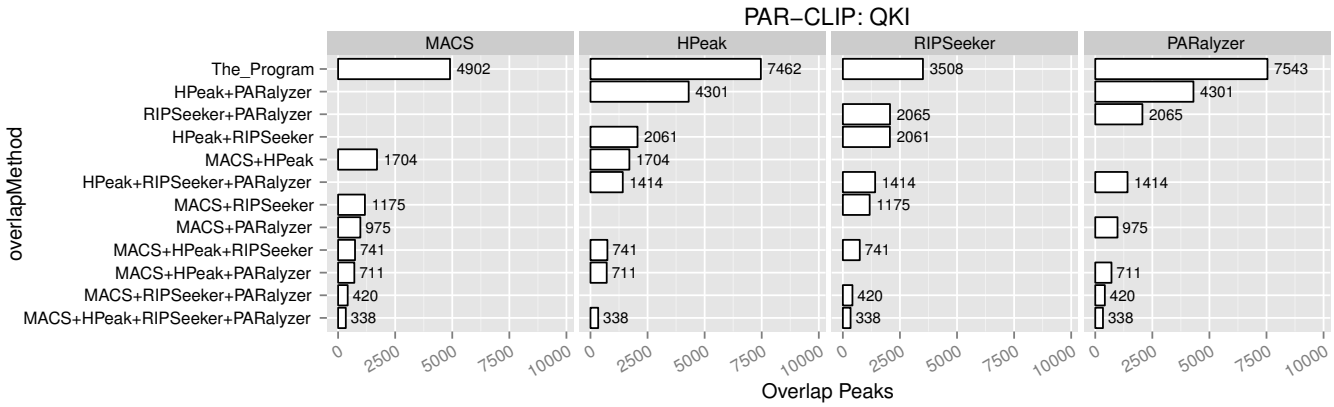
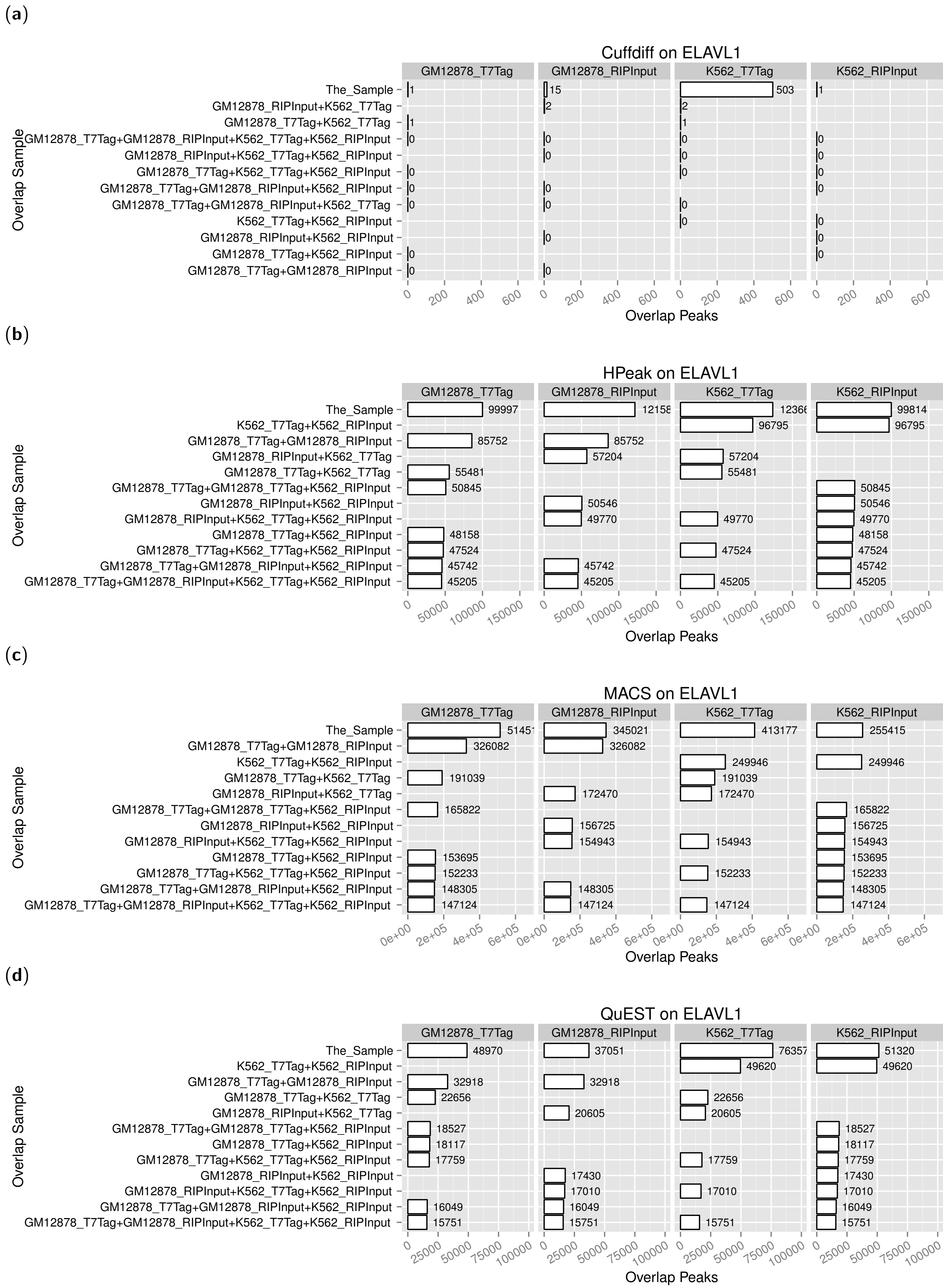
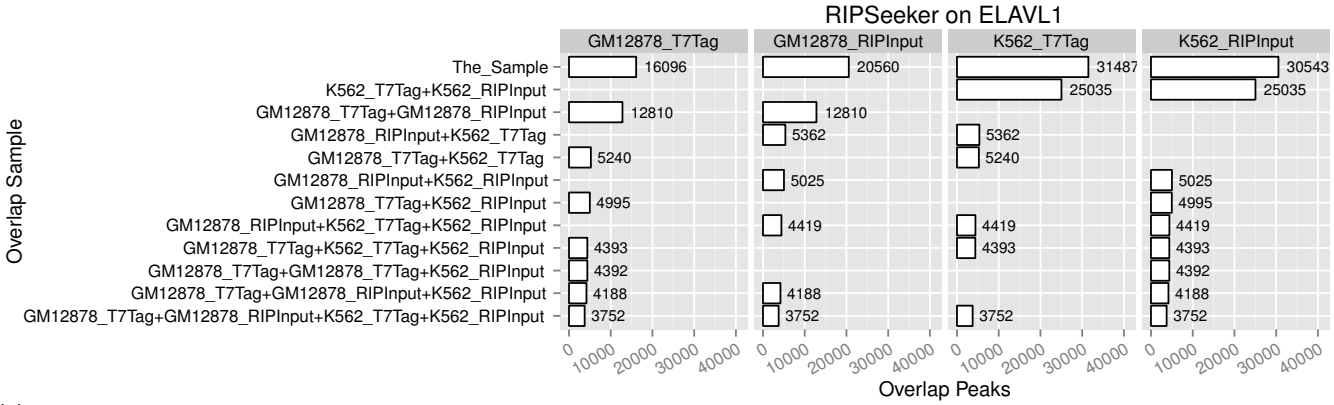


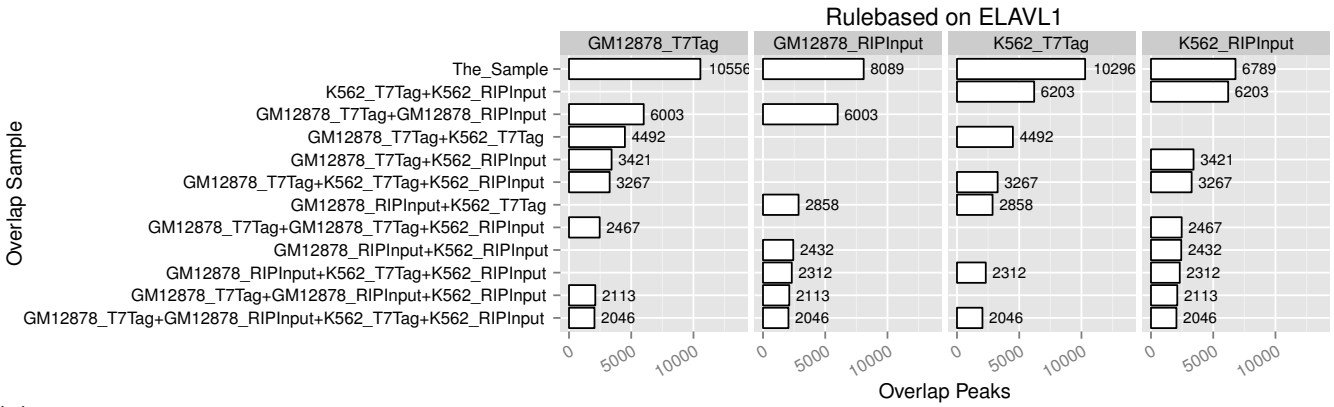
Figure S9. Four-way overlap for peaks identified from the four comparison methods on the two PAR-CLIP datasets (**a**) PUM2 and (**b**) QKI. For pairwise overlap, peaks identified from one program are used to overlap with peaks identified from each of the other three programs. N peaks from one program may overlap with M peaks from another, where N is not necessarily equal to M. To resolve the ambiguity, the number of overlapping peaks is defined as the lower number (i.e. $\min(N, M)$). Similarly, for 3-way and 4-way overlap, the number of overlapping peaks among the 3 programs is the lowest number of overlapping peaks from one of the programs. In each subfigure, there are four panels corresponding to the four program. Each panel shows the total number of peaks for the program ("The_Program") and its overlap with other programs. For instance, MACS+HPeak represent the number of overlapping peaks between MACS and HPeak (4232 for PUM2 and 1704 for QKI). The overlap representation convey more information than the conventional Venn diagram as the number of overlapping peaks relative to the total number of peaks can be directly visualized based on the heights of the bars. As clearly illustrated, for instance, relative to its total number of peaks RIPSeeker has very high overlap with both HPeak and MACS in the PUM2 dataset (**a**) and with HPeak and PARalyzer in the QKI dataset.


(a)
Cuffdiff on ELAVL1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
Overlap Peaks
(b)
HPeak on ELAVL1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
Overlap Peaks
(c)
MACS on ELAVL1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
Overlap Peaks
(d)
QuEST on ELAVL1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
Overlap Peaks

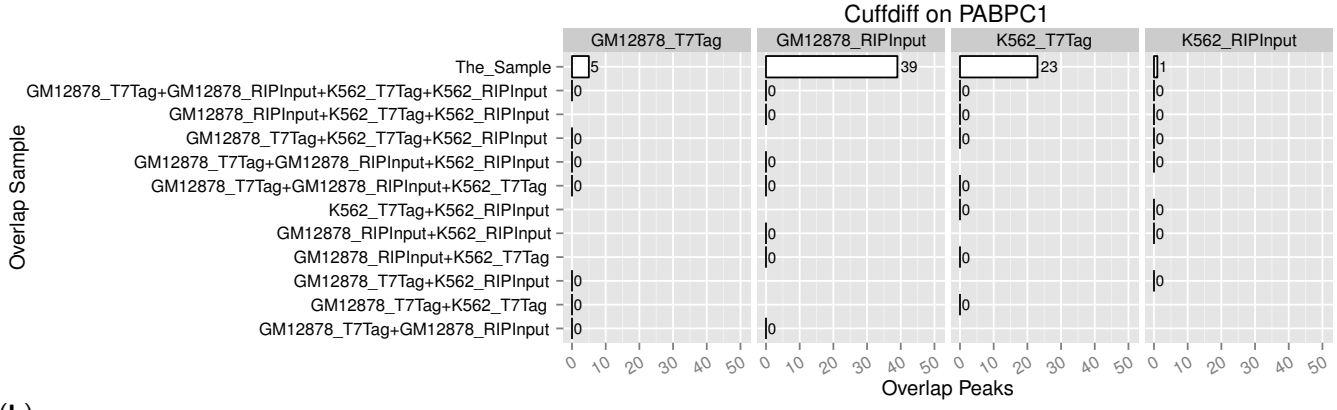
(**e**)


RIPSeeker on ELAVL1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput
GM12878_RIPInput+K562_T7Tag
GM12878_T7Tag+K562_T7Tag
GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
16096
20560
31487
30543
25035
25035
12810
12810
5362
5362
5240
5240
5025
5025
4995
4995
4419
4419
4419
4393
4393
4393
4392
4392
4188
4188
4188
3752
3752
3752
3752
0
10000
20000
30000
40000
Overlap Peaks

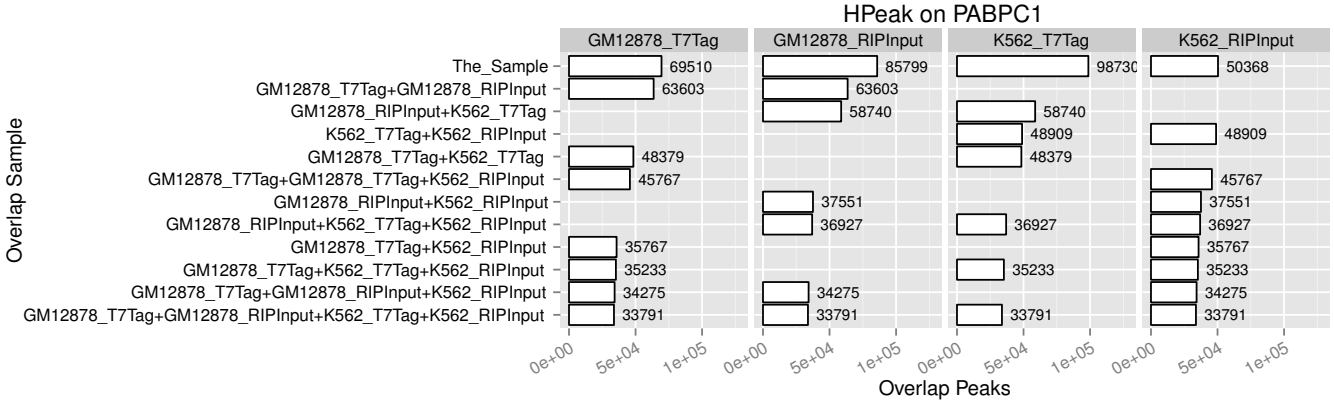
(**f**)


Rulebased on ELAVL1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput
GM12878_T7Tag+K562_T7Tag
GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_T7Tag
GM12878_T7Tag+GM12878_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
10556
8089
10296
6789
6203
6203
6003
6003
4492
4492
3421
3421
3267
3267
3267
2858
2858
2467
2467
2432
2432
2312
2312
2312
2113
2113
2113
2046
2046
2046
2046
0
5000
10000
Overlap Peaks

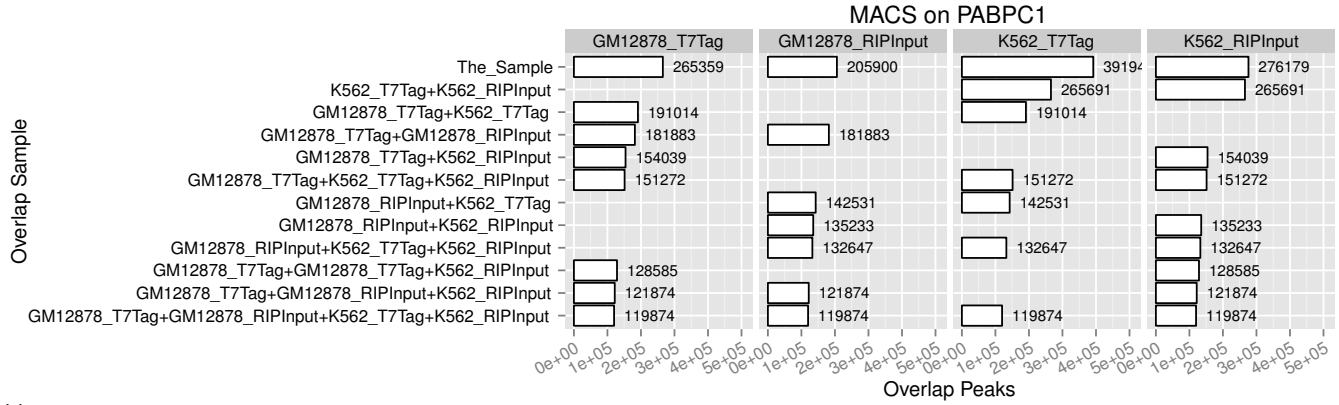
(**g**)


Cuffdiff on PABPC1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag
K562_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_RIPInput
GM12878_RIPInput+K562_T7Tag
GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag
GM12878_T7Tag+GM12878_RIPInput
5
39
23
1
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
10
20
30
40
50
Overlap Peaks

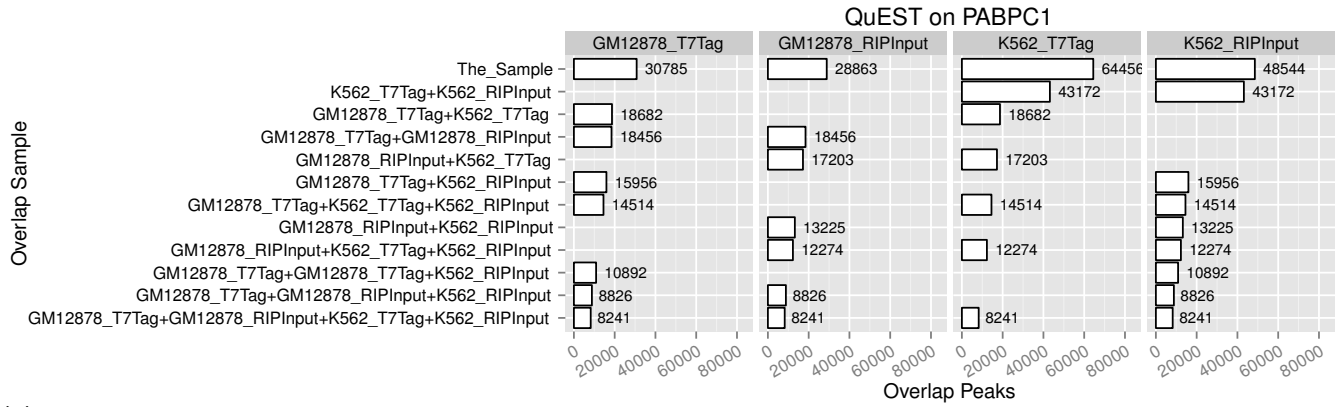
(**h**)


HPeak on PABPC1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
GM12878_T7Tag+GM12878_RIPInput
GM12878_RIPInput+K562_T7Tag
K562_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag
GM12878_T7Tag+GM12878_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
69510
85799
98730
50368
63603
63603
58740
58740
48909
48909
48379
48379
45767
45767
37551
37551
36927
36927
36927
35767
35767
35233
35233
35233
34275
34275
34275
33791
33791
33791
33791
0e+00
5e+04
1e+05
Overlap Peaks

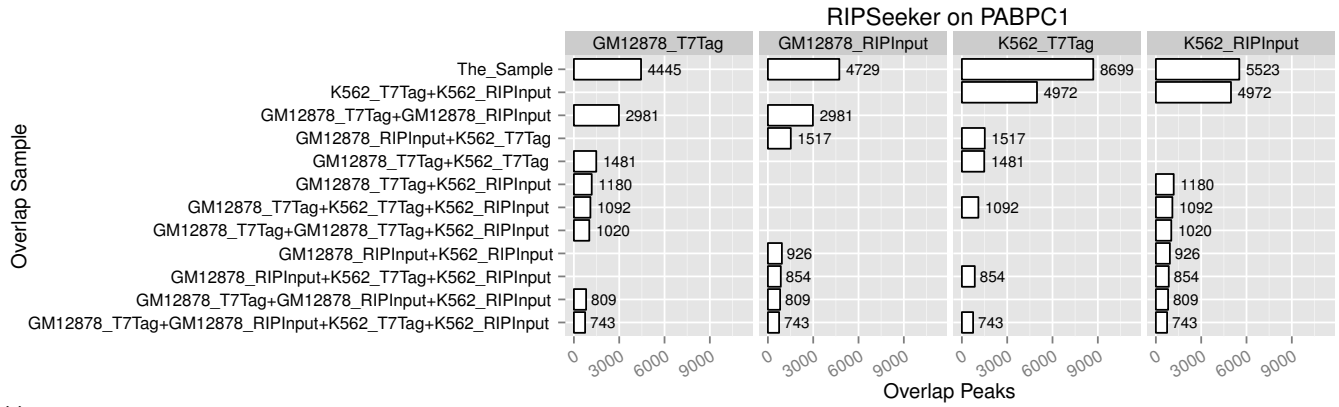
(i)


MACS on PABPC1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
K562_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag
GM12878_T7Tag+GM12878_RIPInput
GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_T7Tag
GM12878_RIPInput+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
265359
205900
265691
276179
191014
181883
154039
151272
142531
135233
132647
128585
121874
119874
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Overlap Peaks

(j)


QuEST on PABPC1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
K562_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag
GM12878_T7Tag+GM12878_RIPInput
GM12878_RIPInput+K562_T7Tag
GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
30785
28863
43172
48544
18682
18456
17203
15956
14514
13225
12274
10892
8826
8241
0
20000
40000
60000
80000
Overlap Peaks

(k)


RIPSeeker on PABPC1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput
GM12878_RIPInput+K562_T7Tag
GM12878_T7Tag+K562_T7Tag
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
4445
4729
8699
5523
4972
2981
1517
1481
1180
1092
1020
926
854
809
743
0
3000
6000
9000
Overlap Peaks

(l)

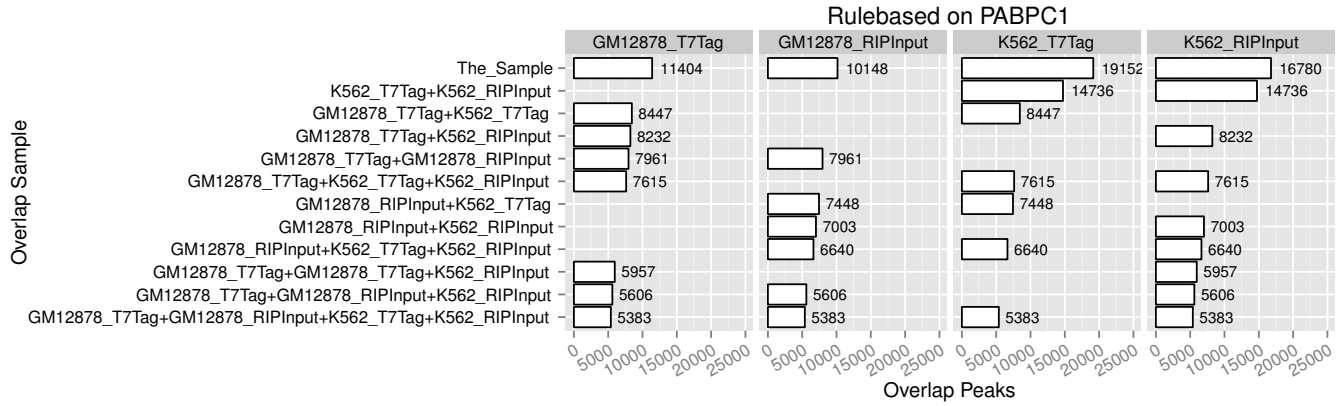
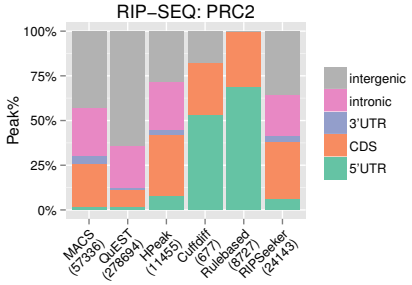

Rulebased on PABPC1
GM12878_T7Tag
GM12878_RIPInput
K562_T7Tag
K562_RIPInput
Overlap Sample
The_Sample
K562_T7Tag+K562_RIPInput
GM12878_T7Tag+K562_T7Tag
GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput
GM12878_T7Tag+K562_T7Tag+K562_RIPInput
GM12878_RIPInput+K562_T7Tag
GM12878_RIPInput+K562_RIPInput
GM12878_RIPInput+K562_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_T7Tag+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_RIPInput
GM12878_T7Tag+GM12878_RIPInput+K562_T7Tag+K562_RIPInput
11404
10148
19152
16780
14736
8447
8232
7961
7615
7448
7003
6640
5957
5606
5383
0
5000
10000
15000
20000
25000
Overlap Peaks

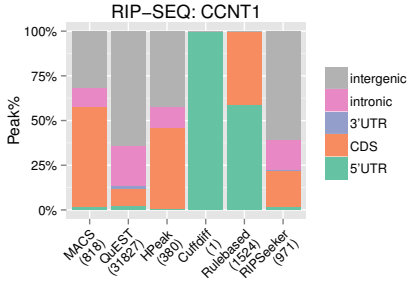
Figure S10. Overlap of peaks identified by each of the six comparison method for the same protein from two different cell lines and controls. The primary goal of this analysis is to examine the robustness of each method on data generated for the same protein using two different cell line and controls. Specifically, protein-RNA interaction sites were predicted as peaks by each method in cell lines GM12878 and K562 by comparing the RIP signal to the background generated from either T7-tag or RIP RNA input as two different types of negative control library for the specific interactions. The peaks identified from these four samples, namely GM12878_T7Tag, GM12878_RIPInput, K562_T7Tag, K562_RIPInput, are subject to 4-way overlap to compare the number of peaks identified from each sample comparison with the number of overlapping peaks between two sample comparisons, among three and among all of the four sample comparisons. For a robust method, a high proportion of overlap is expected within the cell line for the same protein despite using different controls (i.e. GM12878_T7Tag+GM12878_RIPInput; K562_T7Tag+K562_RIPInput) whereas lower number of overlaps is expected between the cell-lines (i.e. all of the remaining overlaps). (**a-f**) Six methods (presented in alphabetic order) Cuffdiff, HPeak, MACS, QuEST, RIPSeeker, and Rulebased on protein ELAVL1 and (**k-l**) on protein PABPC1.

(a)



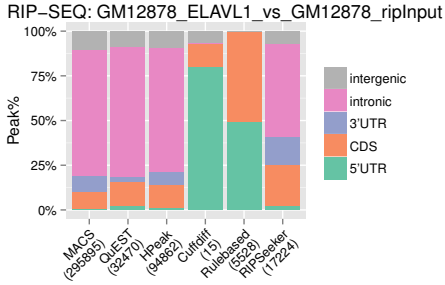
	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker
5'UTR	1e+00	1e+00	1e+00	5e-02	1e+00	1e+00
CDS	1e+00	1e+00	1e+00	8e-01	5e-02	1e+00
intronic	0e+00	0e+00	0e+00	2e-01	3e-03	0e+00
3'UTR	2e-70	1e+00	1e+00	1e+00	1e+00	2e-05

(b)



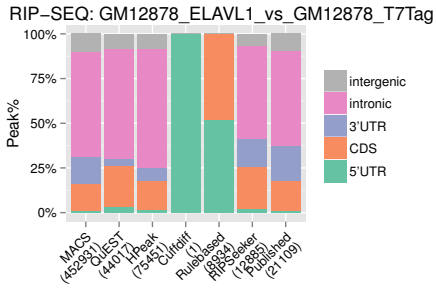
	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker
5'UTR	1e+00	1e+00	1e+00	5e-01	9e-01	1e+00
CDS	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00
intronic	8e-25	0e+00	5e-29	7e-01	1e-06	1e-94
3'UTR	1e+00	1e+00	1e+00	9e-08	1e-14	1e+00

(c)



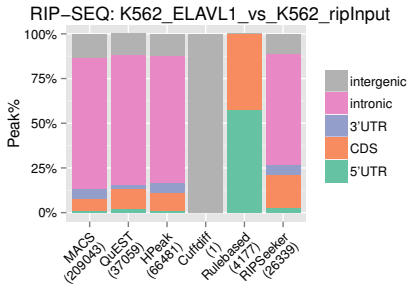
	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker
5'UTR	1e+00	1e+00	1e+00	8e-01	3e-02	1e+00
CDS	1e+00	1e+00	1e+00	6e-01	1e+00	1e+00
intronic	0e+00	0e+00	0e+00	1e-01	2e-33	0e+00
3'UTR	2e-114	1e+00	0e+00	8e-01	2e-16	0e+00

(d)



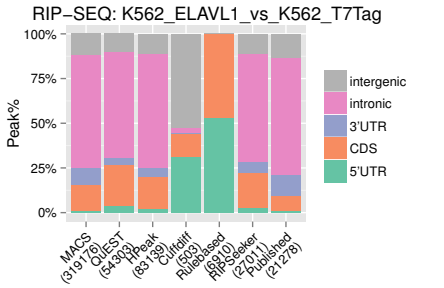
	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker	Published
5'UTR	1e+00	1e+00	1e+00	3e-01	3e-06	1e+00	1e+00
CDS	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00
intronic	0e+00	0e+00	0e+00	3e-01	3e-27	0e+00	0e+00
3'UTR	0e+00	1e-214	0e+00	2e-08	4e-38	0e+00	0e+00

(e)

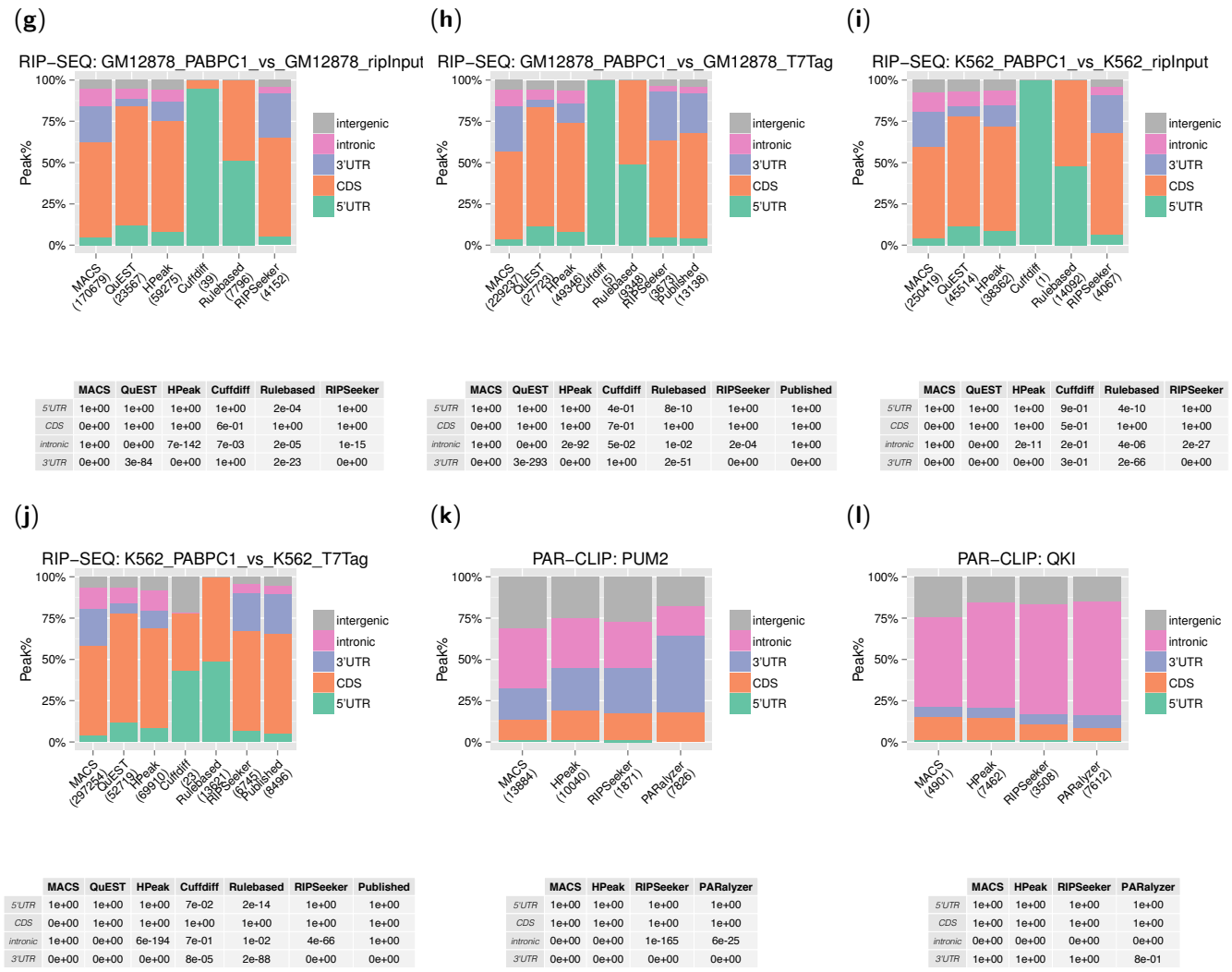


	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker
5'UTR	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00
CDS	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00
intronic	0e+00	0e+00	0e+00	1e+00	3e-25	0e+00
3'UTR	5e-15	1e+00	1e-34	1e+00	2e-10	8e-55

(f)



	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker	Published
5'UTR	1e+00	1e+00	1e+00	8e-01	1e+00	1e+00	1e+00
CDS	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00
intronic	0e+00	0e+00	0e+00	8e-02	9e-15	0e+00	0e+00
3'UTR	3e-186	2e-59	1e-186	1e-04	4e-21	2e-77	2e-199



	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker
5'UTR	1e+00	1e+00	1e+00	1e+00	2e-04	1e+00
CDS	0e+00	1e+00	1e+00	6e-01	1e+00	1e+00
intronic	1e+00	0e+00	7e-142	7e-03	2e-05	1e-15
3'UTR	0e+00	3e-84	0e+00	1e+00	2e-23	0e+00

	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker	Published
5'UTR	1e+00	1e+00	1e+00	4e-01	8e-10	1e+00	1e+00
CDS	0e+00	1e+00	1e+00	7e-01	1e+00	1e+00	1e+00
intronic	1e+00	0e+00	2e-92	5e-02	1e-02	2e-04	1e+00
3'UTR	0e+00	3e-293	0e+00	1e+00	2e-51	0e+00	0e+00

	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker
5'UTR	1e+00	1e+00	1e+00	9e-01	4e-10	1e+00
CDS	0e+00	1e+00	1e+00	5e-01	1e+00	1e+00
intronic	1e+00	0e+00	2e-11	2e-01	4e-06	2e-27
3'UTR	0e+00	0e+00	0e+00	3e-01	2e-66	0e+00

	MACS	QuEST	HPeak	Cuffdiff	Rulebased	RIPSeeker	Published
5'UTR	1e+00	1e+00	1e+00	7e-02	2e-14	1e+00	1e+00
CDS	0e+00	1e+00	1e+00	1e+00	1e+00	1e+00	1e+00
intronic	1e+00	0e+00	6e-194	7e-01	1e-02	4e-66	1e+00
3'UTR	0e+00	0e+00	0e+00	8e-05	2e-88	0e+00	0e+00

	MACS	HPeak	RIPSeeker	PARalyzer
5'UTR	1e+00	1e+00	1e+00	1e+00
CDS	1e+00	1e+00	1e+00	1e+00
intronic	0e+00	0e+00	1e-165	6e-25
3'UTR	0e+00	0e+00	0e+00	0e+00

	MACS	HPeak	RIPSeeker	PARalyzer
5'UTR	1e+00	1e+00	1e+00	1e+00
CDS	1e+00	1e+00	1e+00	1e+00
intronic	0e+00	0e+00	0e+00	0e+00
3'UTR	1e+00	1e+00	1e+00	8e-01

Figure S11. Proportion of peaks overlapping with basic genomic elements. Each peak is assigned with exactly one genomic feature according to the following order of preference: 5′ UTR, coding sequence (CDS), 3′ UTR, intronic, and intergenic based on Ensembl 65 for mouse and Ensembl 69 for human. Peaks predicted from biological replicates are pooled. Hypergeometric tests on enrichment of the four basic genomic elements were performed and the significance tables are presented below each corresponding barplot. Briefly, for each of the four types of genomic elements namely 5'UTR, CDS, intronic, or 3'UTR, we counted how many of them overlap with peaks from each comparison method and denoted this number as x. The hypergeometric test is conducted based on four quantities: x, m, n, k, where x is defined above, m is the total number of the target elements known to Ensembl database, n is the number of non-target elements, and k is the total number of target and non-target elements overlapped by the peaks. The probability that the event generating the four values at random is the computed using R built-in function `phyper(x,m,n,k)`. For instance, we have $x = 11146$ intronic elements and $k = 12580$ elements found in total by the RIPSeeker peaks on QKI PAR-CLIP dataset, and there are $m = 1{,}068{,}140$ total intronic elements and $n = 1{,}435{,}152$ non-intronic elements including 156,609 5'UTR, 1,141,234 CDS, and 137,309 3'UTR in the human genome (Ensembl 69). The probability that the intron is enriched by random is then `phyper(x,m,n,k)` = 0 (or $<$ 1e-308 the smallest decimal number in R) (i.e. almost impossible). Please refer to Figure S5 for more details on each subfigure.

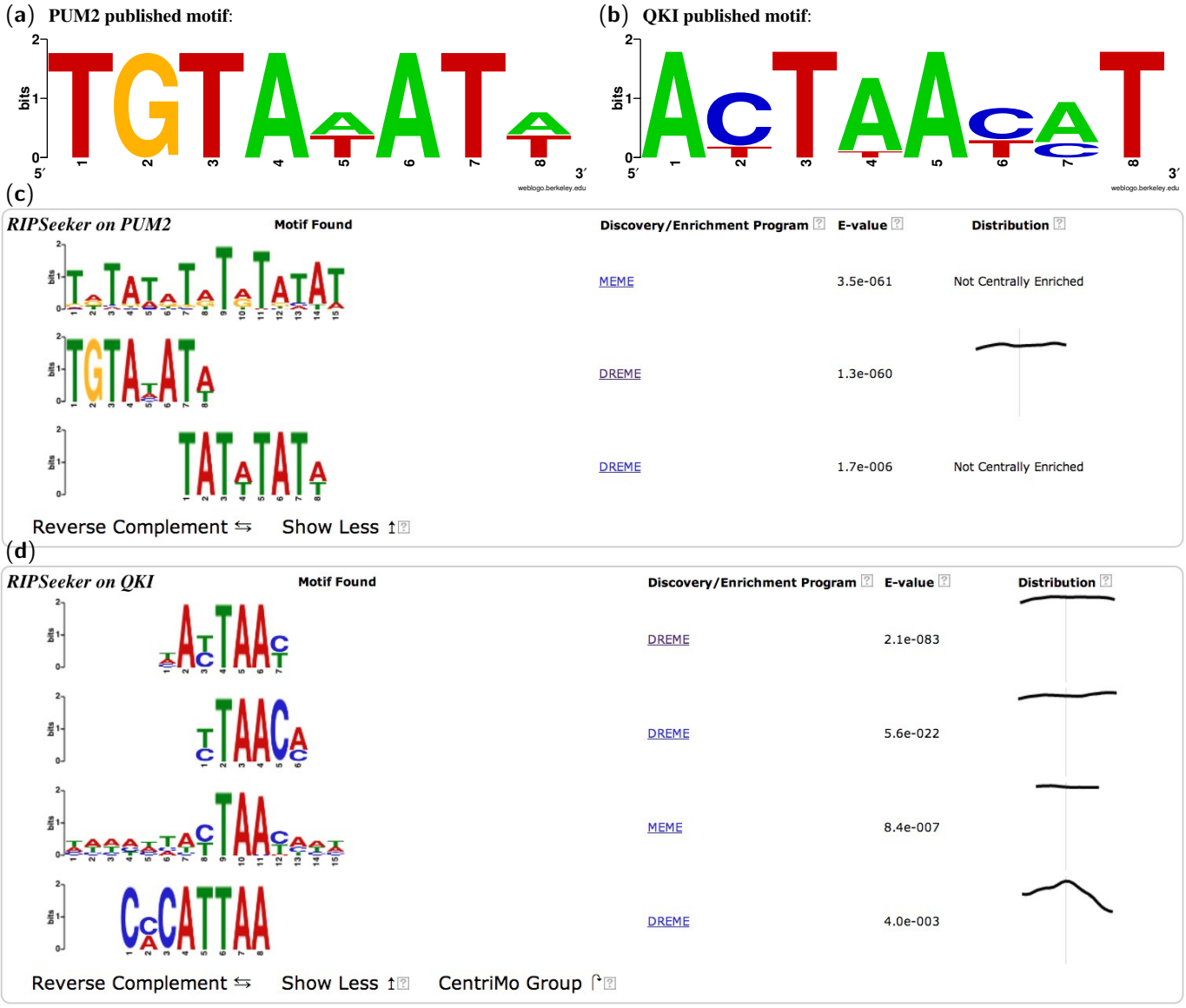


Figure S12. Motif enriched among the top 5000 peaks with the highest RIPScores from RIPSeeker on PUM2 and QKI PAR-CLIP data. (**a**) the published PUM2 motif (5); (**b**) the published QKI motifs (5); (**c**) PUM2 motif enriched among the top 5000 RIPSeeker peaks identified from the PUM2 PAR-CLIP data; (**d**) QKI motif enriched among the top 5000 RIPSeeker peaks identified the from QKI PAR-CLIP data. The motif enrichment was performed on the FASTA sequences of the top 5000 peaks using MEME-ChIP(16), which runs both MEME and DREME as complementary algorithms to predict long and short motif, respectively. As illustrated in the second row of (**c**), the top 5000 peaks from RIPSeeker are enriched for exactly the published motif for PUM2. Similarly, the first and second row of (**d**) bare striking similarity with the published motif for QKI.

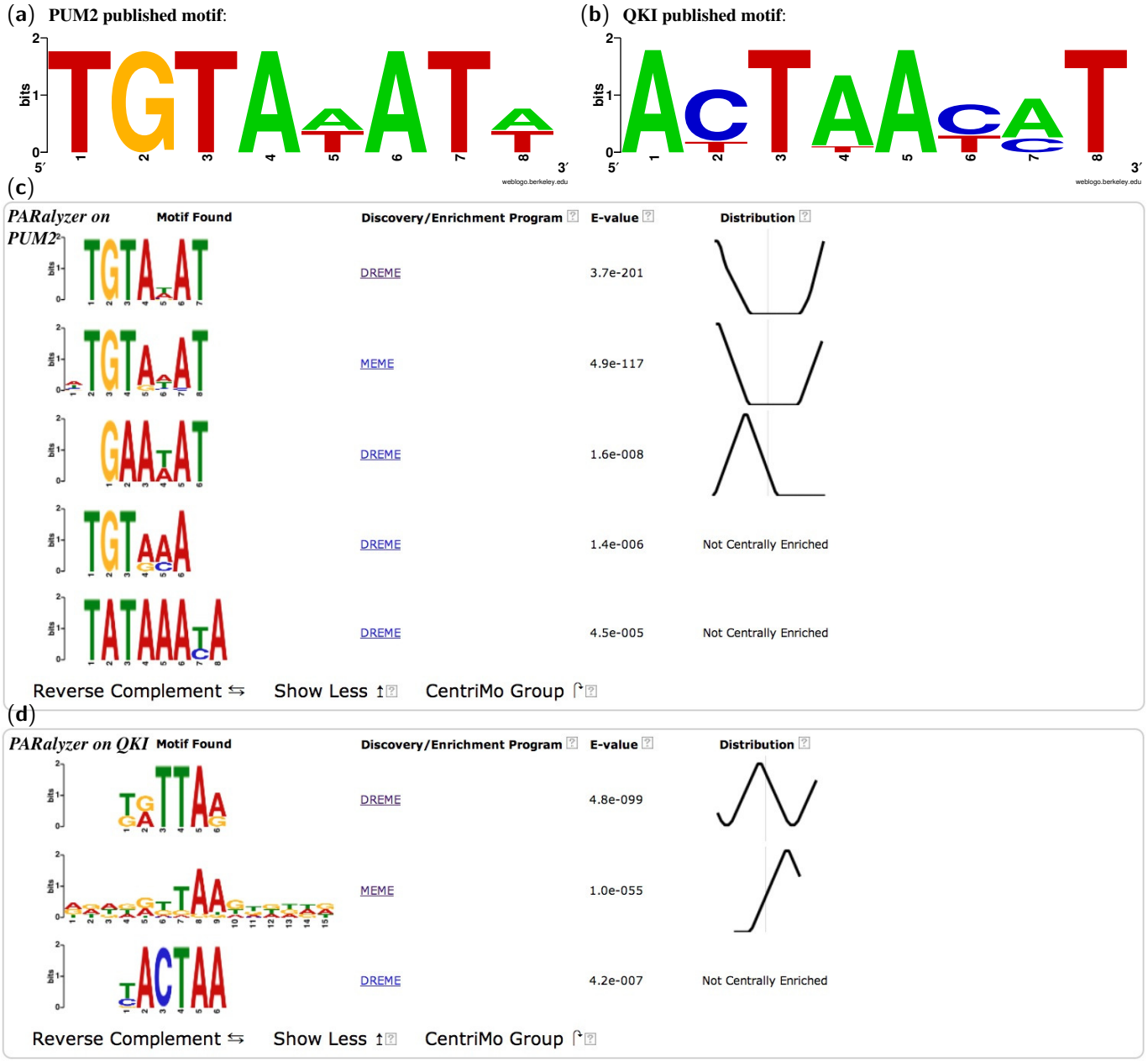


Figure S13. Motif enriched among the top 5000 peaks predicted by PARalyzer on PUM2 and QKI PAR-CLIP data. (**a**) the published PUM2 motif (5); (**b**) the published QKI motifs (5); (**c**) PUM2 motif enriched among the top 5000 PARalyzer peaks identified from the PUM2 PAR-CLIP data; (**d**) QKI motif enriched among the top 5000 PARalyzer peaks identified the from QKI PAR-CLIP data. The top peaks are chosen as the ones having the highest ModeScore (score of the highest signal / (signal + background) value). The motif enrichment was performed on the FASTA sequences of the top 5000 peaks using MEME-ChIP (16), which runs both MEME and DREME as complementary algorithms to predict long and short motif, respectively.

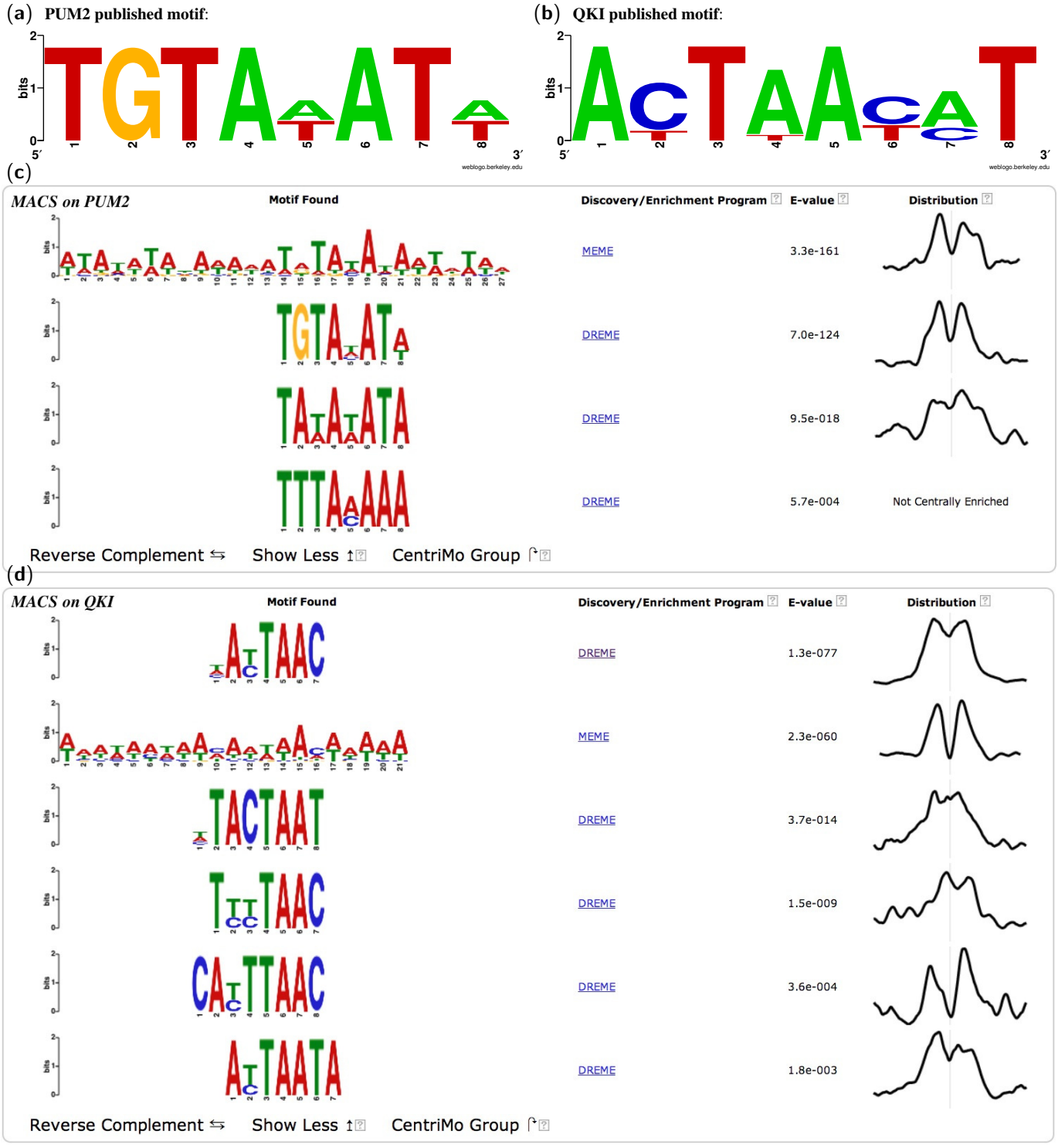


Figure S14. Motif enriched among the top 5000 peaks predicted by MACS on PUM2 and QKI PAR-CLIP data. (**a**) the published PUM2 motif (5); (**b**) the published QKI motifs (5); (**c**) PUM2 motif enriched among the top 5000 MACS peaks identified from the PUM2 PAR-CLIP data; (**d**) QKI motif enriched among the top 5000 MACS peaks identified the from QKI PAR-CLIP data. The top peaks are chosen as the ones having the highest MACS score (i.e. -10log10(p-$value$) in the 5th column of the output BED file). The motif enrichment was performed on the FASTA sequences of the top 5000 peaks using MEME-ChIP (16), which runs both MEME and DREME as complementary algorithms to predict long and short motif, respectively.

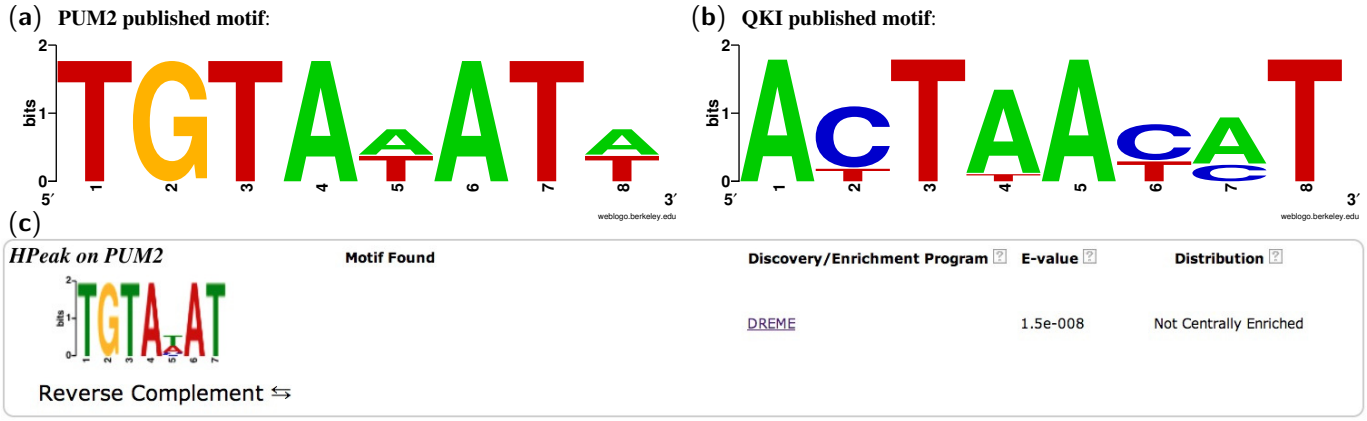


Figure S15. Motif enriched among the top 5000 peaks predicted by HPeak on the PUM2 and QKI PAR-CLIP data. (**a**) the published PUM2 motif (5); (**b**) the published QKI motifs (5); (**c**) PUM2 motif among the top 5000 HPeak peaks; QKI motif is not present among the corresponding top 5000 HPeak peaks. The top peaks are chosen as the ones having the highest absolute normalized cumulative log transformed posterior probability (i.e. the last column of the all.regions.txt output). The motif enrichment was performed on the FASTA sequences of the top 5000 peaks using meme-chip (16), which runs both MEME and DREME as complementary algorithms to predict long and short motif, respectively. The published PUM2 motif is not the #1 rank motif among the 5000 HPeak peaks and have much less significant E-value than the those for the same motif from RIPSeeker, PARalyzer, and MACS (Figure S12, S13, S14).

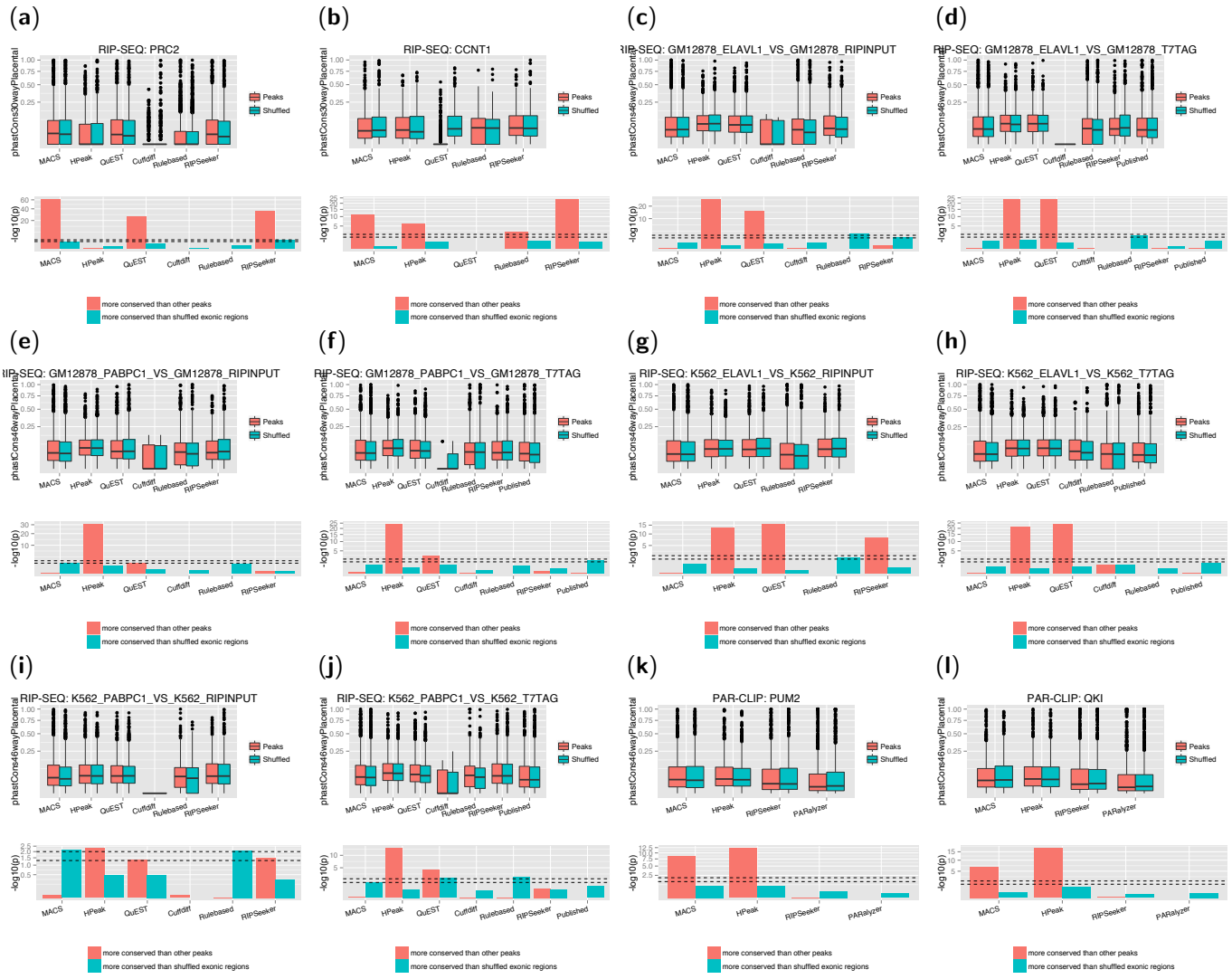


Figure S16. Comparison of conservation for the top 1000 peaks from each program on the RIP-seq and PAR-CLIP datasets. For each program, the top 1000 peaks are the ones having the highest scores based on the specific scoring scheme of that program: for MACS, the highest -10log10(p-$value$) (i.e. 5th column of the BED output); for HPeak, the highest absolute normalized cumulative log transformed posterior probability (i.e. the last column of the all.regions.txt output); for QuEST, highest normalized enrichment fold at the maximum position within the region (i.e. 5th column in ChIP_calls.filtered.bed); for Cuffdiff, highest -log10(p-$value$); for Rulebased, the highest fold-change of RPKM in RIP over control; for RIPSeeker, the highest RIPScores. The conservation score of each top peak is computed as the averaged per-base phastCons46waysPlacental scores for human hg19 reference genome or the averaged per-base phastCons30ways scores for mouse mm9 reference genome (for PRC2 dataset only) downloaded from the UCSC genome browser (`http://hgdownload.cse.ucsc.edu/goldenPath/hg19/phastCons46way/placentalMammals/`; `http://hgdownload.soe.ucsc.edu/goldenPath/mm9/phastCons30way/placental/`). As background control, peaks are randomly shuffled within the same chromosome restricted to the exonic regions based on the Ensembl 65 annotations. The restrictions control for chromosome-dependent and exon-dependent effects. The alternative hypotheses in the one-sided Wilcoxon tests are (1) the peaks from one method are more conserved than the peaks from other methods (i.e. in R, `wilcox.test(peak1, allotherpeaks, alternative=''greater'')`) AND (2) the peaks are more conserved than the randomly shuffled exonic regions (i.e. in R, `wilcox.test(peak, random, alternative=''greater'')`). In each subfigure, the top panel display the boxplots of the averaged conservation scores of the peaks and randomly shuffled exonic regions for each method, and the bottom panel displays the -log10(p-$value$) from the Wilcoxon tests for (1) (indicated as red bar) and (2) (indicated as cyan bar) hypothesis described above. Only when both tests are significant (i.e. both red and cyan bars are above the 0.05 or 0.01 cut-off showns as the dash line), can one ascertain that the corresponding peaks identified by that program are more conserved than others on the same dataset. As shown above, the results are inconclusive. Please refer to Figure S5 legend for details on the test data used in each subfigure.

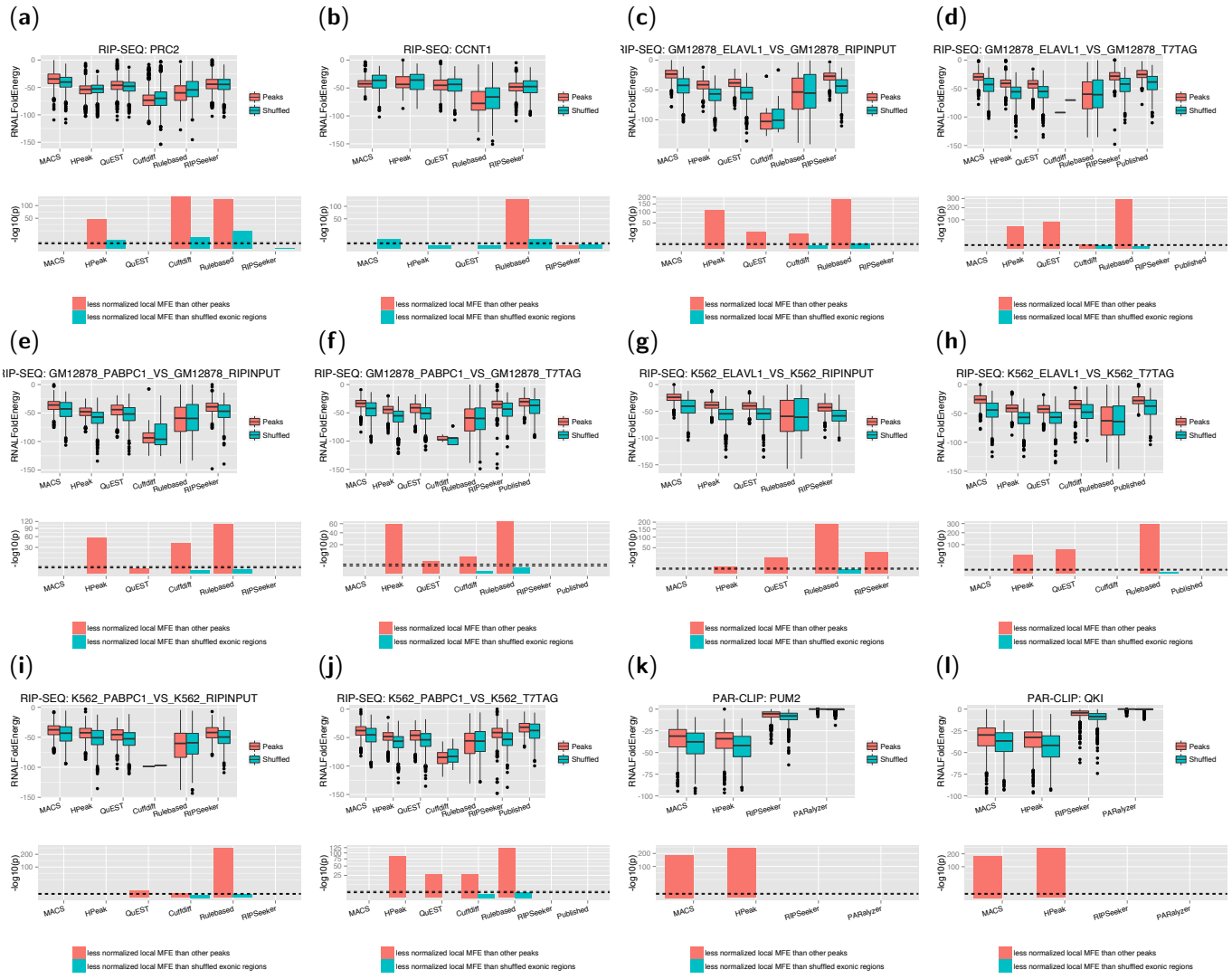


Figure S17. Comparison of folding energy from the predicted local RNA secondary structure for the top 1000 peaks from each program on the RIP-seq and PAR-CLIP datasets. The top 1000 peaks are selected the same way as described in the Figure S16 legend. Local RNA folding energy is computed using `RNALfold` from Vienna RNA Package 2.0 (17). The more negative the minimum free energy (MFE), the more stable the local secondary structure of the RNA. Thus, the alternative hypotheses in the one-sided Wilcoxon tests are (1) the peaks from one method are more stable (having lower engery) than the peaks from other methods (i.e. in R, `wilcox.test(peak1, allotherpeaks, alternative=''less'')`) AND (2) the peaks are more stable than the randomly shuffled exonic regions (i.e. in R, `wilcox.test(peak, random, alternative=''less'')`). In each subfigure, the top panel display the boxplots of the MFE of the peaks and randomly shuffled exonic regions for each method, and the bottom panel displays the -$\log 10(p\text{-}value)$ from the Wilcoxon tests for (1) (indicated as red bar) and (2) (indicated as cyan bar) hypothesis described above. Only when both tests are significant (i.e. both red and cyan bars are above the 0.05 or 0.01 cut-off shown as the dash line), can one ascertain that the corresponding peaks identified by that program are more likely to have stable second structure than others on the same dataset. As shown above, the results are inconclusive. Please refer to Figure S5 legend for details on the test data used in each subfigure.

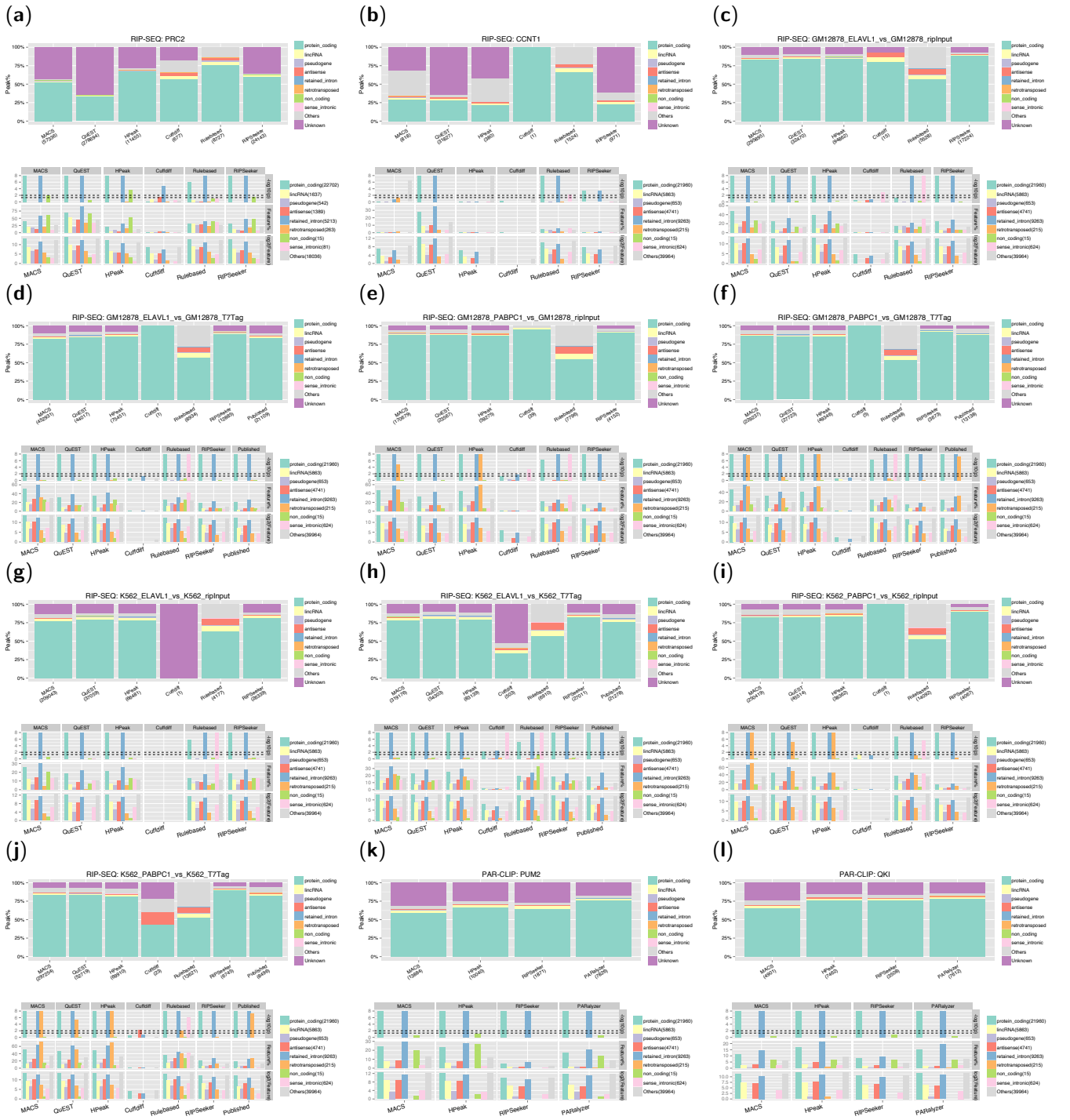
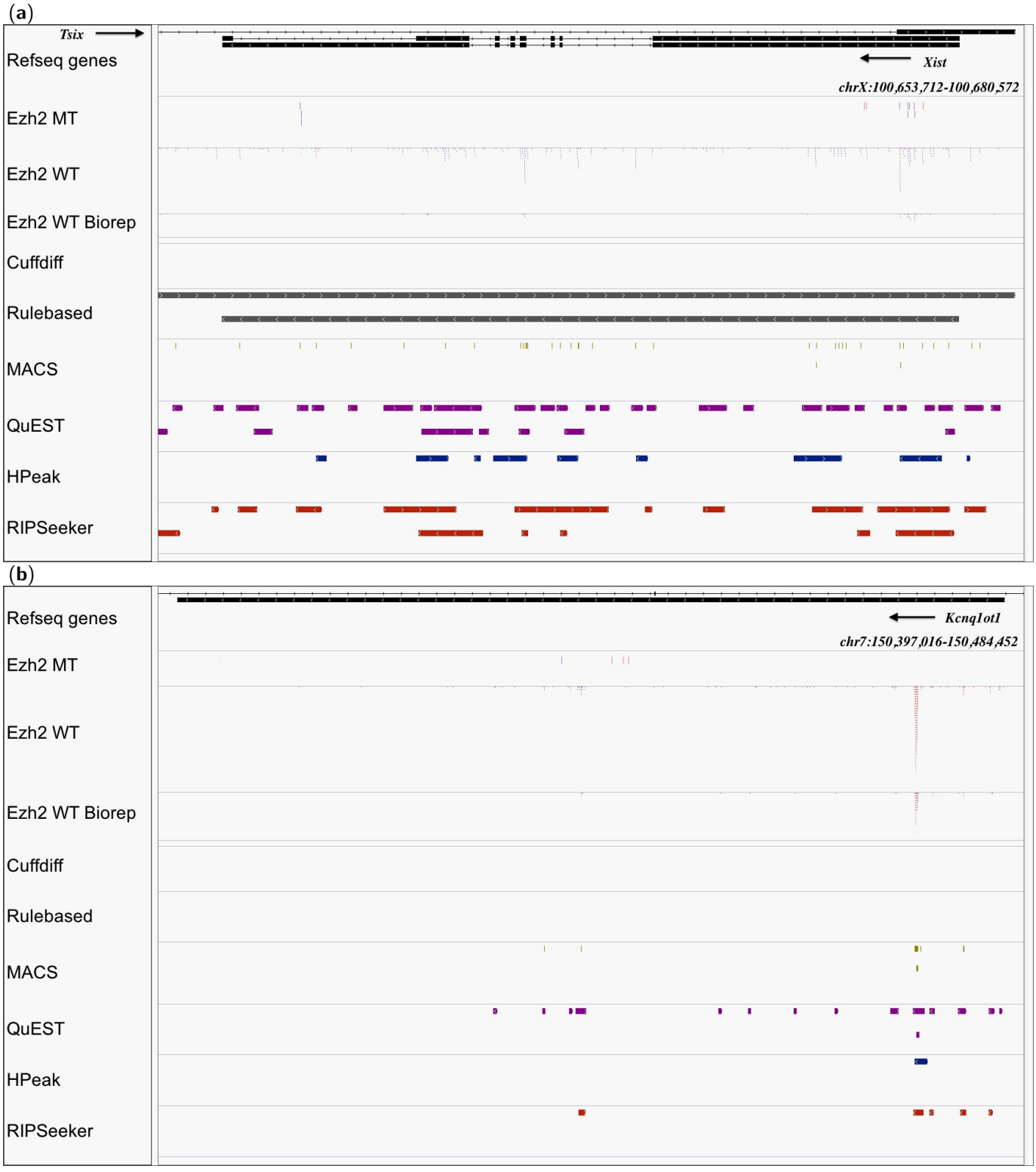


Figure S18. Biotypes category of known genes associated with the peaks from each comparison method. The selected biotypes of interest from "protein coding" to "sense_intronic" correspond to the `transcripts_biotypes` downloaded from Ensembl biomart using the *biomaRt* package. "Others" and "Unknown" correspond to other biotypes such as rRNA, mtRNA etc and peaks that do not overlap with any of the categories, respectively. In each subfigure, the top panel represent the proportion of peaks each overlapping with exactly one biotype category. Each peak is assigned by only one biotype category according to preferential order as listed in each panel legend (i.e. "protein coding" the most preferred to "Others" the least preferred) based on Ensembl 65 for mouse (for PRC2 dataset only) and Ensembl 69 for human; the bottom panel display three layers of information described in the bottom-up order: "log2(Feature)" is the number of transcripts at log2 scale that are overlapped by the peaks; "Feature%" is the percentage of genes in the biotype category that are overlapped by the peaks; "-log10(p)" is the -log10(p-value) from the hypergeometric test for the enrichment of the corresponding biotype category using the same approach as described in Figure S11 legend replacing the four genomic elements with the nine biotype categories. Peaks predicted from biological replicates are pooled, and overlapping peaks are merged before overlapping with the biotype categories. Please refer to Figure S5 for more details on each corresponding subfigure. We are unable to draw meaningful pattern from the results above.


(a)
Tsix
Refseq genes
Xist
chrX:100,653,712-100,680,572
Ezh2 MT
Ezh2 WT
Ezh2 WT Biorep
Cuffdiff
Rulebased
MACS
QuEST
HPeak
RIPSeeker
(b)
Refseq genes
Kcnq1ot1
chr7:150,397,016-150,484,452
Ezh2 MT
Ezh2 WT
Ezh2 WT Biorep
Cuffdiff
Rulebased
MACS
QuEST
HPeak
RIPSeeker

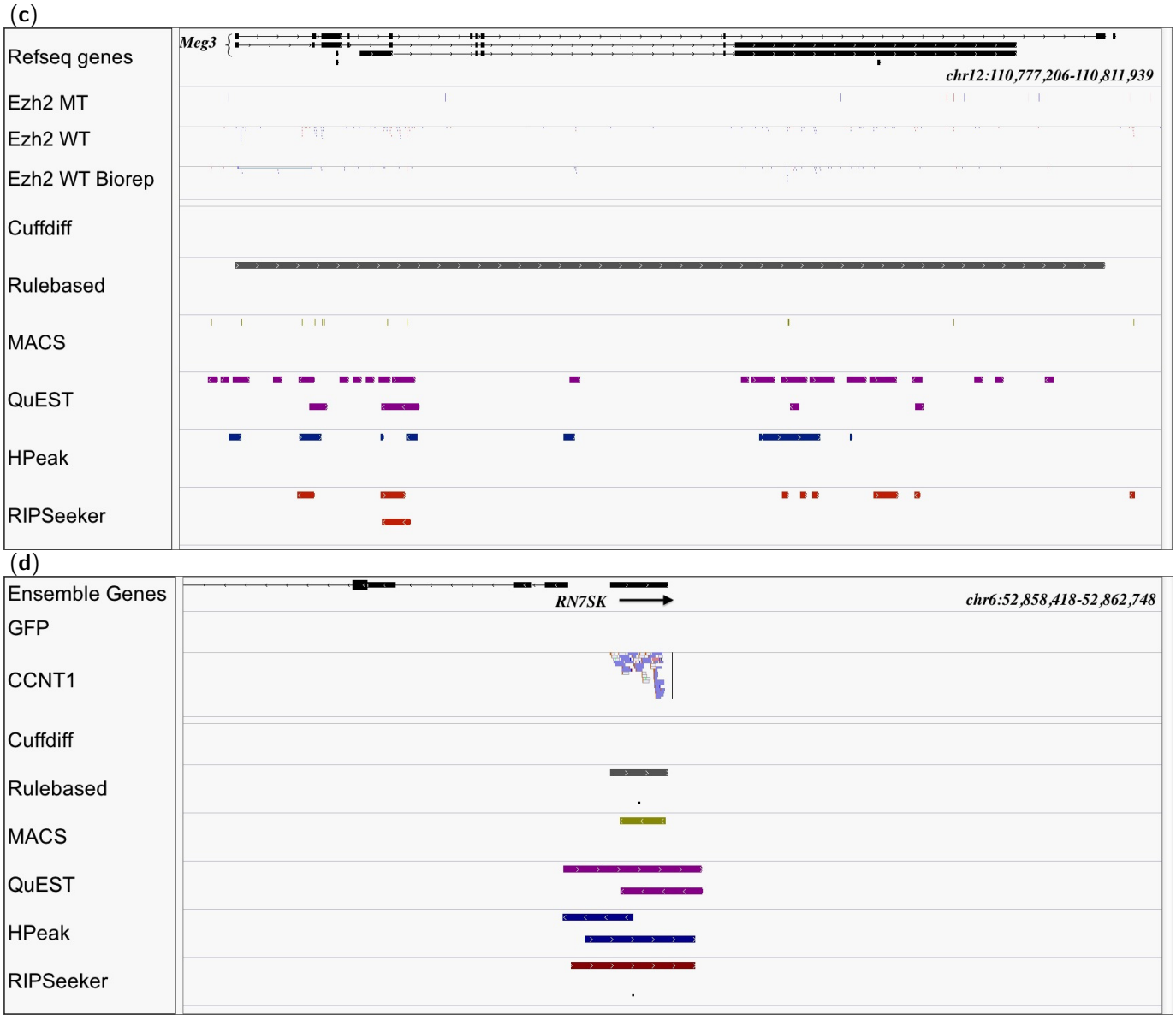
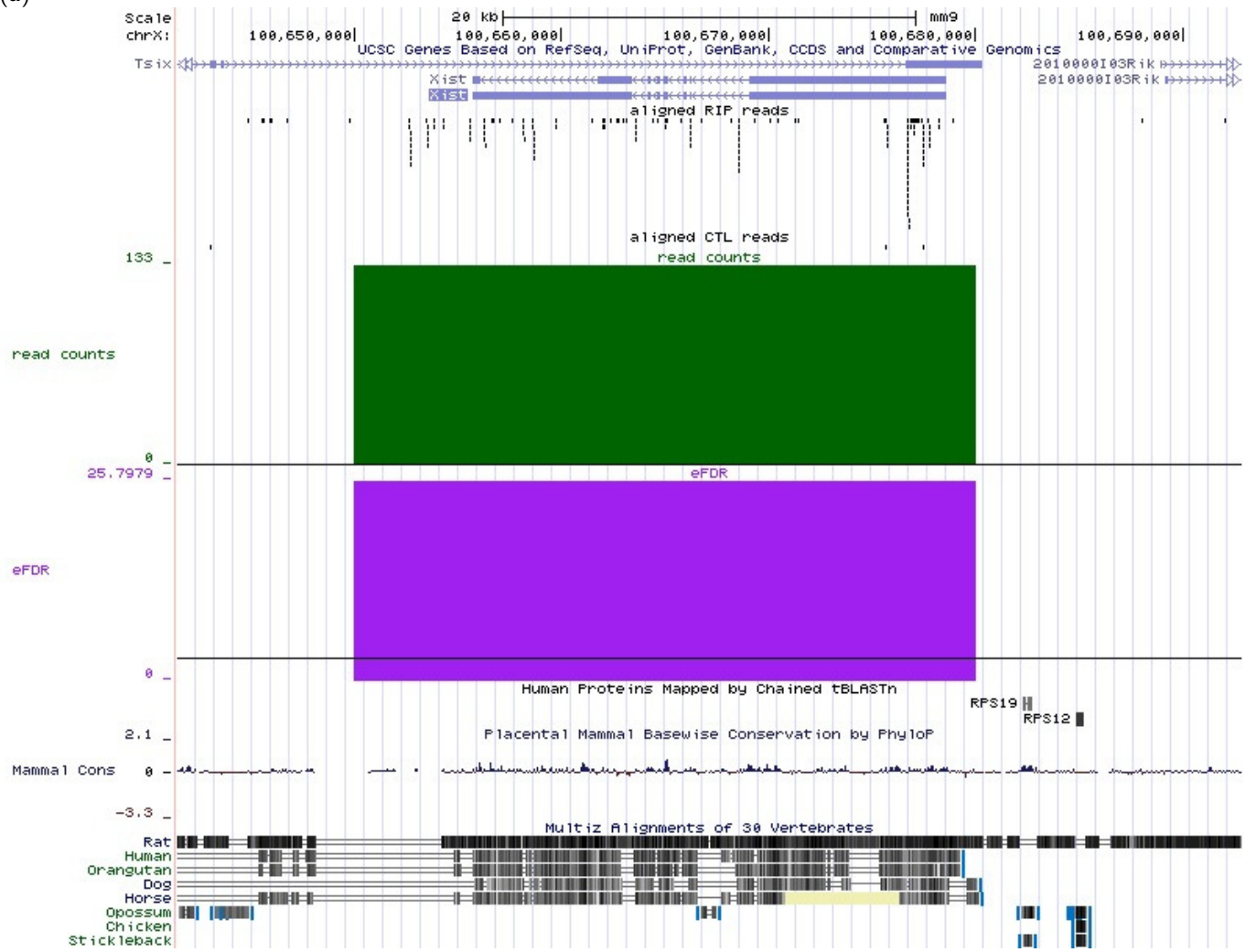
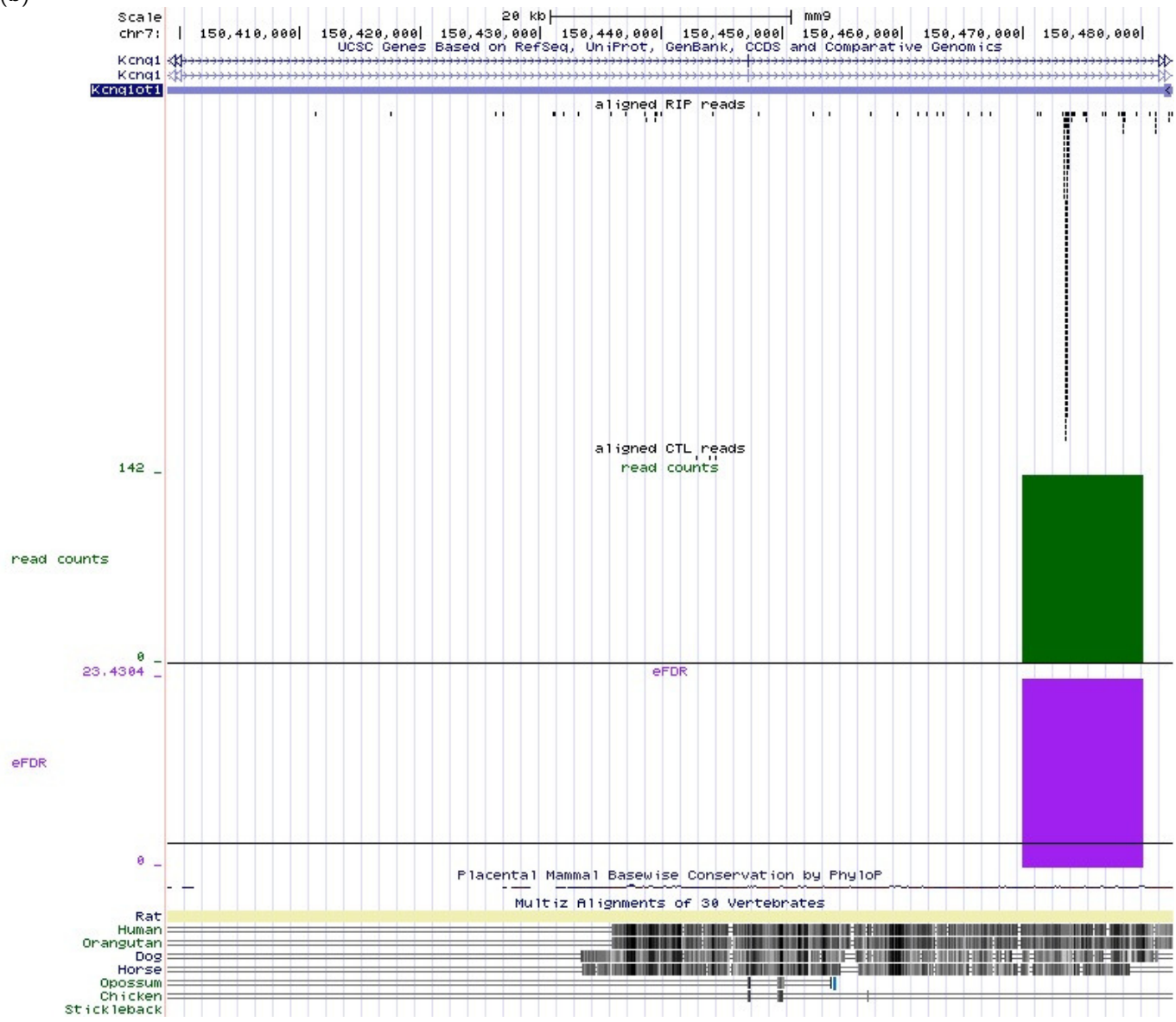


Figure S19. Visualization of predictions by each algorithm on known PRC2-ncRNA (**a**) *Xist*, (**b**) *Kcnq1ot1* and (**c**) *Meg3* (with three alternative transcripts) and (**d**) known CCNT1-ncRNA *RN7SK*. In panel **a**, **b**, and **c**, the top track indicates the reference gene from RefSeq; the track 2, 3, and 4 represent the alignments in Ezh2 mutant (MT) and the two biological replicates for Ezh2 wild type (WT), respectively; the remaining track from top to the bottom represent the peaks predicted by Cuffdiff, Rulebased, MACS, QuEST, HPeak, and RIPSeeker, respectively. In panel **d**, top track indicates the reference gene from Ensembl; the second and third track represent read alignment in GFP control and CCNT1 RIP library, respectively; the remaining tracks are the predictions from the six method as in the previous 3 tracks.

(**a**)


Scale
20 kb
mm9
chrX:
100,650,000
100,660,000
100,670,000
100,680,000
100,690,000
UCSC Genes Based on RefSeq, UniProt, GenBank, CCDS and Comparative Genomics
Tsix
Xist
Xist
2010000I03Rik
2010000I03Rik
aligned RIP reads
aligned CTL reads
read counts
133
read counts
0
25.7979
eFDR
eFDR
0
Human Proteins Mapped by Chained tBLASTn
RPS19
RPS12
2.1
Placental Mammal Basewise Conservation by PhyloP
Mammal Cons
0
-3.3
Multiz Alignments of 30 Vertebrates
Rat
Human
Orangutan
Dog
Horse
Opossum
Chicken
Stickleback

(**b**)


Scale
20 kb
mm9
chr7:
150,410,000
150,420,000
150,430,000
150,440,000
150,450,000
150,460,000
150,470,000
150,480,000
UCSC Genes Based on RefSeq, UniProt, GenBank, CCDS and Comparative Genomics
Kcnq1
Kcnq1
Kcnq1ot1
aligned RIP reads
aligned CTL reads
read counts
142
read counts
0
23.4304
eFDR
eFDR
0
Placental Mammal Basewise Conservation by PhyloP
Multiz Alignments of 30 Vertebrates
Rat
Human
Orangutan
Dog
Horse
Opossum
Chicken
Stickleback

(**c**)

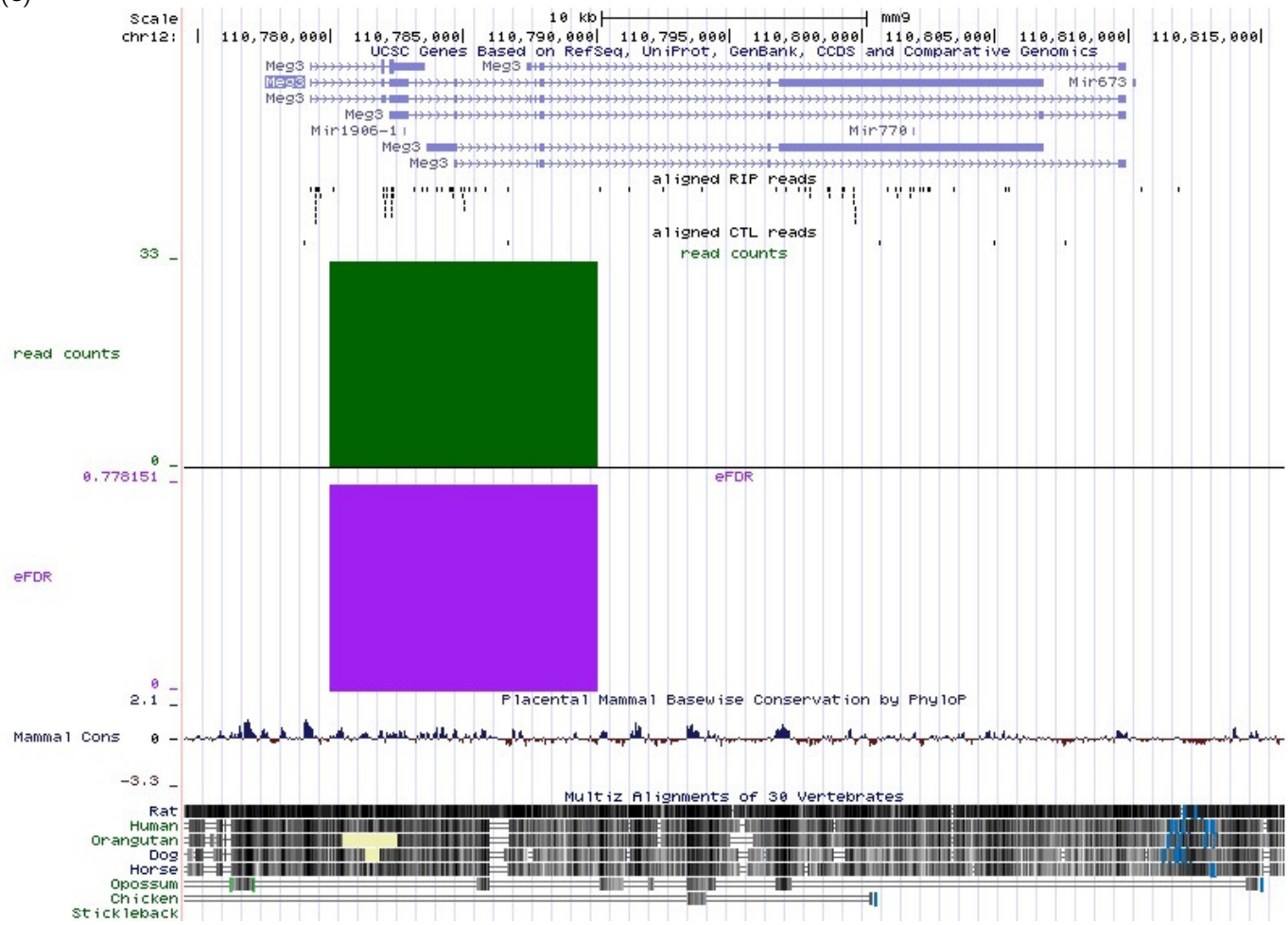


Figure S20. Visualization using UCSC browser on known PRC2-ncRNA (**a**) *Kcnq1ot1* and (**b**) *Xist* on the - strand and (**c**) *Meg3* on the + strand. The UCSC browser is launched by the RIPSeeker's built-in function `viewRIP` with the peak information automatically uploaded as separate tracks in the browser without needing to manually upload files. Users can relate the RIP-seq peaks with other information from the UCSC database by interacting with the browser as they normally. The selected track in this illustration are described as follows. "aligned RIP reads" and "aligned CTL reads": actual read alignments for RIP and control from the RIP-seq data, respectively; "read counts": read count in the RIP library; "eFDR": -log10(eFDR); "Mammal Cons": conservation score from UCSC. For demonstration purpose only, the range for bin size was set between 10 kb and 10.01 kb.

NEGATIVE BINOMIAL HIDDEN MARKOV MODEL

In this section, we present the detailed mathematical formulation of hidden Markov model (HMM) with negative binomial (NB) emission probability to facilitate in text reference of the equations used in the main manuscript. Readers knowledgeable in HMM may skip this section. Section **Definition of HMM** characterizes HMM in terms of its intuition, basic definition, model assumption, and probability properties. Section **Expectation-Maximization** outlines Expectation-Maximization procedures, which is a tractable and efficient way (but might not be optimal) to optimize HMM parameters. Section **Forward-Backward Algorithm** describes Forward-Backward algorithm, which is an efficient way to derive the intermediate quantities required in Section **Expectation-Maximization**. Section **Numerical Approximation of Negative Binomial Parameters** outlines the conditional maximization procedures needed to update NB parameters for HMM. Section **Viterbi Algorithm** describes Viterbi algorithm to derive the most probable sequence of hidden states.

Intuition

Hidden Markov model (HMM) has wide applications in various areas including speech recognition, predicting climate changes, financial forecasting, and gene predictions along the DNA sequence (11). Its wide popularity is due to its ability to efficiently model distribution of observations that are correlated in a sequence of events and thus are not independent identically distributed (*i.i.d.*). None-*i.i.d.* is common in biological data. Expression of gene at the upstream may affect the gene expression at the downstream. Binding of one RNA transcript to a protein may affect the protein recruitment by other RNA molecules. In reality, however, whether the gene is active/inactive or the RNA truly binds/unbinds to a protein is hidden from us. The read coverage within each defined genomic region from a sequencing instrument is merely a noisy reflection of the true hidden states. Using the probabilistic approach of HMM, we would like to (1) model the data distribution by maximizing the likelihood of the model parameters based on the observed data (covered by the following subsections up to Viterbi algorithm) (2) infer under uncertainty the most probable sequence of (correlated) hidden states from the learnt distribution (covered by Viterbi algorithm). The following detailed formulation of HMM is largely based on Chapter 13 from (11).

Definition of HMM

Let $\mathbf{Z}$ be the hidden variables and $\mathbf{X}$ be the observed variables, HMM is defined by the following five properties. Readers can refer to Figure S21 to visualize the model.

1. K, the number of hidden states for $\mathbf{Z}$. For instance, there are two states $k \in \{1,2\}$ in the HMM digram (Figure S21).
2. N, the length of the hidden variables or observations such that $\mathbf{Z} = \{z_1, z_2, \ldots, z_N\}$ and $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$.
3. $\mathbf{A}$, the transition matrix with each entry $a_{jk} \equiv p(z_n = k | z_{n-1} = j)$. In other words, a_{jk} is the probability at which the latent variable switches from state j at $(n-1)^{th}$ time point or position to k at n^{th} time point or position. Notably, a_{jk} is subject to probability constraints such that

   $\forall a_{jk} 0 \leq a_{jk} \leq 1$

   $\sum_{k=1}^{K} a_{jk} = 1$
4. ϕ, a set of parameters governing conditional distribution of each $p(x_n | z_{n,k} \phi_k)$ (Figure S21). For instance, $\mathbf{X}$ can be continuous variables and follows a Gaussian conditional distribution, $p(x_n | \phi_k) \sim \mathcal{N}(\mu_k, \Sigma_k)$; or discrete in a negative binomial distribution, $p(x_n | \phi_k) \sim NB(a_k, b_k)$, which is the interest in this study.

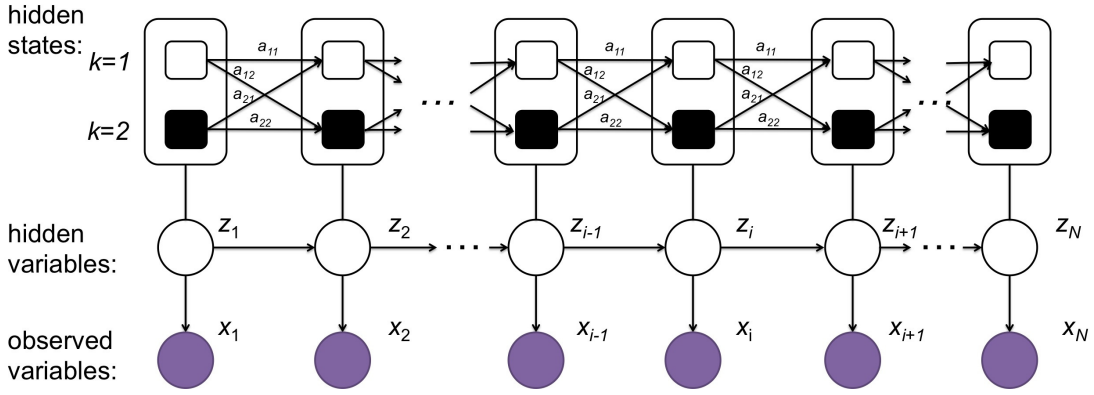


Figure S21. Schematic diagram of two-state Hidden Markov model.

5. $\boldsymbol{\pi}$, a set of initial distributions with each $\pi_k \equiv p(z_1 = k)$, where $k \in 1,\ldots,K$.

For clarity of notation, we use $\boldsymbol{\theta}$ to represent all of the model parameters:

$$\boldsymbol{\theta} = \{\mathbf{A}, \boldsymbol{\phi}, \boldsymbol{\pi}\}$$

Markovian Assumptions

Two important assumptions of HMM that makes maximization of the entire joint distribution of observed and latent variables tractable. Besides the formal definitions below, these assumptions are implicated in Figure S21.

1. **Markov Assumption**: Given the state of the current latent variable, the state of the next latent variable is conditionally independent of all the earlier preceding latent variables (Figure S21):

$$z_{n+1} \perp\!\!\!\perp z_{n-1} | z_n \tag{1}$$

   Due to the Markov property, the conditional (or transition) probability of the state for each latent variables can be induced as:

$$p(z_{n+1}|z_1, z_2, \ldots, z_n) = p(z_{n+1}|z_n) \tag{2}$$

   where each $p(z_{n+1}|z_n)$ has already been defined as a_{jk} in the transition matrix $\mathbf{A}$ from the definition above.

2. **Independent Assumption**: Given the state of its parent latent variable, the observed variable is independent from other variables:

$$x_n \perp\!\!\!\perp \{\mathbf{X}_{/x_n}, \mathbf{Z}_{/z_n}\} | z_n \tag{3}$$

   where $\mathbf{X}_{/x_n}$ and $\mathbf{Z}_{/z_n}$ denote the corresponding variables excluding x_n and z_n, respectively. As a result, the conditional probability of generating the entire observed data given the latent variables can be factorized as follows.

$$p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\phi}) = \prod_{n=1}^{N} p(x_n|z_n, \boldsymbol{\phi}) \tag{4}$$

Joint Probability

Based on (**2**) and (**4**), the entire joint probability of data and latent variables can be simplified as follows.

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) &= p(x_1, x_2, \ldots, x_N, z_1, z_2, \ldots, z_N|\boldsymbol{\theta}) \\ &= p(z_1|\boldsymbol{\pi}) \left[\prod_{n=2}^{N} p(z_n|z_{n-1}, \mathbf{A}) \right] \prod_{m=1}^{N} p(x_m|z_m, \boldsymbol{\phi}) \end{aligned} \tag{5}$$

where we have taken advantage of the conditional independence of latent and observed variables. The objective is then to maximize the likelihood $L(\boldsymbol{\theta}|\mathbf{X}, \mathbf{Z}) \equiv p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ by fitting the model parameters $\boldsymbol{\theta}$. Because the latent variables are unobserved, we need to marginalize over all of their hidden states and instead maximize the expectation of the logarithmic likelihood of the joint probability:

$$Q(\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \tag{6}$$

In short, (**6**) is the *objective function* we need to optimize by tuning parameters $\boldsymbol{\theta}$. However, direct maximization of (**6**) by partial differentiation leads to complex expression with no closed-form solution. We therefore turn to an iterative optimization approach described in the following section.

Expectation-Maximization

The Expectation-Maximization (EM) algorithm starts with an initial setting $\boldsymbol{\theta}^{old}$ for the parameters in order to computes the expectation in (**6**) (E-step). It then maximizes (**6**) by setting the model parameters to their maximum likelihood solution (M-step). The process alternates between E and M-step until n iterations or no further improvement observed in the objective function (**6**). Specifically, if we substitute (**5**) into (**6**) and perform some simple manipulation of the formula, we obtain:

$$Q(\boldsymbol{\theta}^{old}) = \sum_{k=1}^{K} \gamma(z_{1,k}) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{n,k}) \ln A_{jk} + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{n,k}) \ln p(x_n|\phi_k) \tag{7}$$

where

$$\gamma(z_{n,k}) = p(z_{n,k}|\mathbf{X}, \boldsymbol{\theta}^{old}) \tag{8}$$

$$\xi(z_{n-1,j}, z_{n,k}) = p(z_{n-1,j}, z_{n,k}|\mathbf{X}, \boldsymbol{\theta}^{old}) \tag{9}$$

To simplify notation, $z_{n,k}$ denotes z_n having state k (i.e., $z_n = k$).

Therefore, the goal of E-step is to evaluate (**8**) and (**9**), which can be done efficiently by an algorithm described in the following section.

The M-step requires solving the partial derivative equation $\frac{\partial Q(\boldsymbol{\theta}^{old})}{\partial \boldsymbol{\theta}} = 0$ for $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$. The solutions for each parameter are listed as follows (11).

$$\pi_k = \frac{\gamma(z_{1,k})}{\sum_{j=1}^{K} \gamma(z_{1,j})} \tag{10}$$

$$A_{j,k} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,k})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,l})} \tag{11}$$

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{n,k}) x_n}{\sum_{n=1}^{N} \gamma(z_{n,k})} \tag{12}$$

where $\mu_k \in \boldsymbol{\mu} \equiv \boldsymbol{\phi}$ is the mean of the discrete one-dimensional observed variables for latent state k.

Therefore, we need $\gamma(z_{n,j})$ and $\xi(z_{n-1,j}, z_{n,k})$ in order to evaluate (**10**), (**11**), and (**12**). The procedure to obtain both quantities is described in the following section.

Forward-Backward Algorithm

The Forward-Backward algorithm is formulated naturally through the following mathematical derivation for $\gamma(z_n)$.

$$\begin{aligned}
\gamma(z_n) &= p(z_n|\mathbf{X}) \\
&= \frac{p(\mathbf{X}|z_n)p(z_n)}{p(\mathbf{X})} && (Bayes'\ rule) \\
&= \frac{p(x_1, x_2, \ldots, x_n|z_n)p(x_{n+1}, x_{n+2}, \ldots, x_N|z_n)p(z_n)}{p(\mathbf{X})} && (by\ (\mathbf{4})) \\
&= \frac{p(x_1, x_2, \ldots, x_n, z_n)p(x_{n+1}, x_{n+2}, \ldots, x_N|z_n)}{p(\mathbf{X})} \\
&= \frac{\alpha(z_n)\beta(z_n)}{p(\mathbf{X})}
\end{aligned} \tag{13}$$

where $\alpha(z_n)$ and $\beta(z_n)$ represent the quantities computed in the forward and backward phase, respectively. The mathematical derivations of both quantities naturally falls into recurrent relations as shown below.

$$\begin{aligned}
\alpha(z_n) &= p(x_1,x_2,\ldots,x_n,z_n) \\
&= p(x_1,\ldots,x_n|z_n)p(z_n) \\
&= p(x_n|z_n)p(x_1,\ldots,x_{n-1}|z_n)p(z_n) && (by\ \mathbf{(4)}) \\
&= p(x_n|z_n)p(x_1,\ldots,x_{n-1},z_n) \\
&= p(x_n|z_n)\sum_{z_{n-1}} p(x_1,\ldots,x_{n-1},z_{n-1},z_n) \\
&= p(x_n|z_n)\sum_{z_{n-1}} p(x_1,\ldots,x_{n-1},z_n|z_{n-1})p(z_{n-1}) \\
&= p(x_n|z_n)\sum_{z_{n-1}} p(x_1,\ldots,x_{n-1}|z_{n-1})p(z_n|z_{n-1})p(z_{n-1}) && (by\ \mathbf{(2)}) \\
&= p(x_n|z_n)\sum_{z_{n-1}} p(x_1,\ldots,x_{n-1},z_{n-1})p(z_n|z_{n-1}) \\
&= p(x_n|z_n)\sum_{z_{n-1}} \alpha(z_{n-1})p(z_n|z_{n-1})
\end{aligned}$$

Therefore, we obtain the recursion for $\alpha(z_n)$:

$$\alpha(z_n) = p(x_n|z_n)\sum_{z_{n-1}} \alpha(z_{n-1})p(z_n|z_{n-1}) \qquad \mathbf{(14)}$$

Evaluation of $\alpha(z_n)$ requires the value of $\alpha(z_{n-1})$. To start the recursion (i.e., the base case), we must have the value for $\alpha(z_1)$ for each state k:

$$\alpha(z_{1,k}) = p(x_1,z_{1,k}) = p(z_{1,k})p(x_1|z_{1,k}) = \pi_k p(x_1|\phi_k) \qquad \mathbf{(15)}$$

Since we know both π_k and $p(x_1|\phi_k)$ (from the initial parameters $\boldsymbol{\theta}^{old}$), we can start the process by passing the "message" of $\alpha(z_1)$ forward along the chain to obtain $\alpha(z_n)$. If we set $n=N$, we can compute all of the $\alpha(z_1),\ldots,\alpha(z_N)$ in a single forward pass.

The backward pass for $\beta(z_n)$ involves a similar derivation:

$$\begin{aligned}
\beta(z_n) &= p(x_{n+1},x_{n+2},\ldots,x_N|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1},x_{n+2},\ldots,x_N,z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1},x_{n+2},\ldots,x_N|z_{n+1},z_n)p(z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1},x_{n+2},\ldots,x_N|z_{n+1})p(z_{n+1}|z_n) && (\text{by } \mathbf{(4)}) \\
&= \sum_{z_{n+1}} p(x_{n+2},\ldots,x_N|z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n) && (\text{by } \mathbf{(4)}) \\
&= \sum_{z_{n+1}} \beta(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n)
\end{aligned}$$

Thus, we obtain the solution for $\beta(z_n)$, which is another recurrent relation:

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n) \qquad \mathbf{(16)}$$

To initiate the backward message passing, we set the end of the Markov chain $\beta(z_N)=1$ for all $k\in\{1,\ldots,K\}$ states. Now, if we set $n=1$, we can obtain all of the $\beta(z_1),\ldots,\beta(z_N)$ in a single backward pass.

To compute $\gamma(z_n)$ (**13**), we also need $p(\mathbf{X})$, which is easy to obtain by picking any n^{th} observation and performing the following.

$$\begin{aligned} p(\mathbf{X}) &= \sum_{z_n} p(x_1,x_2,\ldots,x_n,z_n)p(x_{n+1},x_{n+2},\ldots,x_N|z_n) \\ &= \sum_{z_n} \alpha(z_n)\beta(z_n) \end{aligned}$$

Together, we have all the components, $\alpha(z_n)$, $\beta(z_n)$, and $p(\mathbf{X})$ required to compute all of the $\gamma(z_1)\ldots\gamma(z_N)$ specified in (**13**). The evaluation of $\xi(z_{n-1},z_n)$ also depends on $\alpha(z_n)$ and $\beta(z_n)$:

$$\begin{aligned} \xi(z_{n-1},z_n) &= p(z_{n-1},z_n|\mathbf{X}) \\ &= \frac{p(\mathbf{X}|z_{n-1},z_n)p(z_{n-1},z_n)}{p(\mathbf{X})} && \text{(Bayes' rule)} \\ &= \frac{p(x_1,\ldots,x_{n-1}|z_{n-1})p(x_n|z_n)p(x_{n+1},\ldots,x_N|z_n)p(z_n|z_{n-1})p(z_{n-1})}{p(\mathbf{X})} && \text{(by (4))} \\ &= \frac{p(x_1,\ldots,x_{n-1},z_{n-1})p(x_n|z_n)p(z_n|z_{n-1})p(x_{n+1},\ldots,x_N|z_n)}{p(\mathbf{X})} \\ &= \frac{\alpha(z_{n-1})p(x_n|z_n)p(z_n|z_{n-1})\beta(z_n)}{p(\mathbf{X})} && \text{(by definition)} \end{aligned}$$

Therefore, we now have both $\gamma(z_n)$ and $\xi(z_{n-1},z_n)$ and can evaluate (**10**), (**11**), and (**12**) for $\boldsymbol{\theta}=\{\boldsymbol{\pi},\mathbf{A},\boldsymbol{\phi}\}$ promised in the M-step of the EM algorithm in the previous section. However, there is a further complication in optimizing the parameters of negative binomial conditional distribution, which is addressed in the following section.

Negative Binomial distribution

For the specific emission probability distribution, RIPSeeker uses negative binomial (NB) distribution to model the read count distributions in order to infer background and RIP regions. NB has been shown by (10) to be a more realistic parametric model than Gaussian and Poisson models. Gaussian distribution (popular in microarray analysis) might provide a good approximation to large read counts but becomes inadequate to model the majority of the low count values due to their discreteness and skewness. Poisson distribution is intuitive in that it expresses the probability of a given number of events occurring in a fixed interval given the average rate λ, which can be interpreted in our context as the probability of number of reads sampled/sequenced from the fixed bin given the average read counts across all of the bins. The Poisson model offers the convenience that the single model parameter λ represents both the mean and the variance of the discrete distribution but becomes inadequate to model data with larger variance than the mean, which is known as over-dispersion problem (10). On the other hand, NB is suitable to model discrete events with separate mean and variance via two parameters a and b. Formally, let x_n be the read count in n^{th} bin sampled from NB(a,b), then the conditional probability of x_n can be expressed as:

$$p(x_n|z_{n,k}) = \binom{x_n+a_k-1}{a_k-1}\left(\frac{b_k}{1+b_k}\right)^{a_k}\left(\frac{1}{1+b_k}\right)^{x_n} \quad \textbf{(17)}$$

where a_k and b_k are the parameters for the function given the latent variable with state k. Importantly, the mean μ_k and variance σ_k^2 of the data sampled from NB$_k$ can also be expressed in terms of a_k and b_k:

$$\mu_k = \frac{a_k}{b_k} \quad \textbf{(18)}$$

$$\sigma_k^2 = \frac{a_k(1+b_k)}{b_k^2} \quad \textbf{(19)}$$

Notably, NB can also be viewed as a Poisson(λ) distribution, where λ is itself a random variable, distributed according to Gamma(a, b) with a and b as shape and scale parameter, respectively (10). Thus, NB is a both intuitive and flexible choice for modelling read count distribution.

Numerical Approximation of Negative Binomial Parameters

Now, in the M-step of EM algorithm, we obtain the maximum likelihood function for μ_k (**12**). However, we cannot evaluate a_k and b_k simultaneously. Thus, we turn to a variation of the M-step called *conditional maximization* (12), where we fix variable a_k to evaluate b_k using (**12**) and (**18**), and then use Newton's method to update a_k. Specifically, we solve for b_k as follows.

$$\frac{a_k^{old}}{b_k^{new}} = \frac{\sum_{n=1}^{N}\gamma(z_{n,k})x_n}{\sum_{n=1}^{N}\gamma(z_{n,k})} \qquad \text{(by (12) and (18))}$$

$$b_k^{new} = \frac{a_k^{old}\sum_{n=1}^{N}\gamma(z_{n,k})}{\sum_{n=1}^{N}\gamma(z_{n,k})x_n}$$

$$= \frac{a_k^{old}\sum_{n=1}^{N}\alpha(z_{n,k})\beta(z_{n,k})}{\sum_{n=1}^{N}\alpha(z_{n,k})\beta(z_{n,k})x_n} \qquad \text{(by (13))} \qquad \textbf{(20)}$$

$$a_k^{new} = a_k^{old} - \frac{f'(a_k^{old})}{f''(a_k^{old})} \qquad \textbf{(21)}$$

where $f(a_k^{old})$ is the logarithmic posterior probability of the data $p(\mathbf{X}|\mathbf{Z})$, which is the product of the conditional probability by the assumption (**4**); f' and f'' are the first and second derivative of $f\ w.r.t\ a_k$, respectively:

$$f(a_k^{old}) = \ln\prod_{n=1}^{N} p(x_n|z_{n,k}) \qquad \textbf{(22)}$$

$$f'(a_k^{old}) = N\left[\ln\left(\frac{b_k^{new}}{1+b_k^{new}}\right) - \psi(a_k^{old})\right] + \sum_{n=1}^{N}\left[\beta(z_n)\psi(x_n + a_k^{old})\right] \qquad \textbf{(23)}$$

$$f''(a_k^{old}) = -N\psi_1(a_k^{old}) + \sum_{n=1}^{N}\left[\beta(z_n)\psi_1(x_n + a_k^{old})\right] \qquad \textbf{(24)}$$

where $\psi(a_k^{old})$ and $\psi_1(a_k^{old})$ are the di and trigamma function, which are the first and second derivative of the logarithmic gamma function $\Gamma(x) = (x-1)!$, respectively:

$$\psi(a_k^{old}) = \frac{d}{d\mathbf{X}}\ln\Gamma(\mathbf{X}) \qquad \textbf{(25)}$$

$$\psi_1(a_k^{old}) = \frac{d^2}{d^2\mathbf{X}}\ln\Gamma(\mathbf{X}) \qquad \textbf{(26)}$$

For detailed numerical approximation of (**25**) and (**26**), refer to the Matlab code in the software package.

Viterbi Algorithm

After learning the model parameters θ, we can obtain the sequence of hidden states for $1\ldots N$ latent variables that maximizes the joint log-likelihood $p(x_1,\ldots,x_N,z_1,\ldots,z_N)=p(\mathbf{X},\mathbf{Z})$ using Viterbi algorithm.

$$
\begin{aligned}
p(\mathbf{X},\mathbf{Z}) &= p(z_1)\left[\prod_{n=2}^{N} p(z_n|z_{n-1})\right]\prod_{m=1}^{N} p(x_m|z_m)\\
\ln p(\mathbf{X},\mathbf{Z}) &= \ln p(z_1)+\sum_{n=2}^{N}\ln p(z_n|z_{n-1})+\sum_{m=1}^{N}\ln p(x_m|z_m)\\
&= \ln p(z_1)+\ln p(x_1|z_1)+\sum_{n=2}^{N}\Big[\ln p(z_n|z_{n-1})+\ln p(x_n|z_n)\Big]\\
\max_{z_1,\ldots,z_N}\ln p(\mathbf{X},\mathbf{Z}) &= \max_{z_1,\ldots,z_N}\left\{\ln p(z_1)+\ln p(x_1|z_1)+\sum_{n=2}^{N}\Big[\ln p(z_n|z_{n-1})+\ln p(x_n|z_n)\Big]\right\}\\
&= \max_{z_2,\ldots,z_N}\left\{\max_{z_1}\Big[\ln p(z_1)+\ln p(x_1|z_1)+\ln p(z_2|z_1)\Big]+\ln p(x_2|z_2)\right.\\
&\qquad \left.+\sum_{n=3}^{N}\Big[\ln p(z_n|z_{n-1})+\ln p(x_n|z_n)\Big]\right\}\\
&= \max_{z_2,\ldots,z_N}\left\{\max_{z_1}\Big[\ln p(x_1,x_2,z_1,z_2)\Big]+\sum_{n=3}^{N}\Big[\ln p(z_n|z_{n-1})+\ln p(x_n|z_n)\Big]\right\}\\
&= \max_{z_3,\ldots,z_N}\left\{\max_{z_2}\Big[\max_{z_1}\Big[\ln p(x_1,x_2,z_1,z_2)\Big]+\ln p(z_3|z_2)\Big]+\ln p(x_3|z_3)\right.\\
&\qquad \left.+\sum_{n=4}^{N}\Big[\ln p(z_n|z_{n-1})+\ln p(x_n|z_n)\Big]\right\}\\
&= \max_{z_3,\ldots,z_N}\left\{\max_{z_1,z_2}\Big[\ln p(x_1,x_2,x_3,z_1,z_2,z_3)\Big]+\sum_{n=4}^{N}\Big[\ln p(z_n|z_{n-1})+\ln p(x_n|z_n)\Big]\right\}\\
&\vdots\\
&= \max_{z_N}\left\{\max_{z_{N-1}}\left[\max_{z_1,\ldots,z_{N-2}}\big[\ln p(x_1,\ldots,x_{N-2},z_1,\ldots,z_{N-2})\big]\right.\right.\\
&\qquad \left.\left.+\ln p(z_N|z_{N-1})\right]+\ln p(x_N|z_N)\right\}\\
&= \max_{z_N}\left\{\max_{z_1,\ldots,z_{N-1}}\Big[\ln p(x_1,\ldots,x_{N-1},z_1,\ldots,z_{N-1})+\ln p(z_N|z_{N-1})\Big]+\ln p(x_N|z_N)\right\}
\end{aligned}
$$

where $p(\mathbf{X},\mathbf{Z}|\theta)\equiv p(\mathbf{X},\mathbf{Z})$ for clarity; a uniform prior of $p(z_1)$ is used; and $\max_{z_1}\ln p(x_1,x_2,z_1,z_2)$ indicates that for each state of z_2, find z_1 that maximizes the $\ln p$ of reaching to that state. Similarly, $\max_{z_1,z_2}\ln p(x_1,x_2,x_3,z_1,z_2,z_3)$ denotes finding z_1 and z_2 that maximizes the $\ln p$ of reaching to each state of z_3. Because z_1 is obtained at the previous iteration, we only needs to find z_2 by following the "path" from z_1. More generally, we have derived a recurrent relation:

$$
\begin{aligned}
&\max_{z_1,\ldots,z_N}\ln p(x_1,\ldots,x_N,z_1,\ldots,z_N)\\
&= \max_{z_N}\left\{\max_{z_1,\ldots,z_{N-1}}\Big[\ln p(x_1,\ldots,x_{N-1},z_1,\ldots,z_{N-1})+\ln p(z_N|z_{N-1})\Big]+\ln p(x_N|z_N)\right\}
\end{aligned}
\tag{27}
$$

The derivation of (**27**) itself is in fact the algorithm used to find z_n that maximizes $\ln p(x_1,\ldots,x_{n+1},z_1,\ldots,z_{n+1})$ for each state of z_{n+1}. The choice of z_n is saved in order to *backtrack* to find the entire sequence of latent states $z_1,\ldots,z_{N-1}$. More specifically, at the end of iteration, we first choose the state for z_N that maximizes the entire joint probability and *backtrack* to the choice of z_{N-1} that reaches to the optimal state of z_N, then to z_{N-2} that reaches to z_{n-1}, and so forth.

If there are K states, then K^2 computation is required at each iteration to find the optimal state of z_{n-1} for each state of z_n. For N data points corresponding to N latent variables, the Viterbi algorithm takes O(K^2N). Thus, the algorithm is linear to the growth of data points and takes only $O(4N)$ for a two-state HMM.

REFERENCES

1. Martin, M. (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet journal,*.
2. Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology,* **10**(3), R25.
3. Trapnell, C., Pachter, L., and Salzberg, S. L. (May, 2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics (Oxford, England),* **25**(9), 1105–1111.
4. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and 1000 Genome Project Data Processing Subgroup (August, 2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England),* **25**(16), 2078–2079.
5. Corcoran, D. L., Georgiev, S., Mukherjee, N., Gottwein, E., Skalsky, R. L., Keene, J. D., and Ohler, U. (2011) PARalyzer: definition of RNA binding sites from PAR-CLIP short-read sequence data.. *Genome Biology,* **12**(8), R79.
6. Quinlan, A. R. and Hall, I. M. (March, 2010) BEDTools: a flexible suite of utilities for comparing genomic features.. *Bioinformatics (Oxford, England),* **26**(6), 841–842.
7. Valouev, A., Johnson, D. S., Sundquist, A., Medina, C., Anton, E., Batzoglou, S., Myers, R. M., and Sidow, A. (August, 2008) Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data. *Nature Methods,* **5**(9), 829–834.
8. Zhang, Y., Liu, T., Meyer, C., and Eeckhoute, J. (2008) Model-based analysis of ChIP-Seq (MACS). *Genome . . . ,*.
9. Roberts, A., Pimentel, H., Trapnell, C., and Pachter, L. (June, 2011) Identification of novel transcripts in annotated genomes using RNA-Seq.. *Bioinformatics (Oxford, England),*.
10. Anders, S. and Huber, W. (October, 2010) Differential expression analysis for sequence count data. *Genome Biology,* **11**(10), R106.
11. Bishop, C. (2006) Pattern recognition and machine learning, Number 605–631 in Information Science and StatisitcsSpringer Science, .
12. Meng, X. (1993) Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika,*.
13. Zhao, J., Ohsumi, T. K., Kung, J. T., Ogawa, Y., Grau, D. J., Sarma, K., Song, J. J., Kingston, R. E., Borowsky, M., and Lee, J. T. (December, 2010) Genome-wide Identification of Polycomb-Associated RNAs by RIP-seq. *Molecular Cell,* **40**(6), 939–953.
14. Hafner, M., Landthaler, M., Burger, L., Khorshid, M., Hausser, J., Berninger, P., Rothballer, A., Ascano, M., Jungkamp, A.-C., Munschauer, M., Ulrich, A., Wardle, G. S., Dewell, S., Zavolan, M., and Tuschl, T. (April, 2010) Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell,* **141**(1), 129–141.
15. ENCODE Project Consortium (June, 2007) Identification and analysis of functional elements in 1by the ENCODE pilot project.. *Nature,* **447**(7146), 799–816.
16. Machanick, P. and Bailey, T. L. (June, 2011) MEME-ChIP: motif analysis of large DNA datasets.. *Bioinformatics (Oxford, England),* **27**(12), 1696–1697.
17. Lorenz, R., Bernhart, S. H., Höner Zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011) ViennaRNA Package 2.0.. *Algorithms for molecular biology : AMB,* **6**, 26.