nature structural & molecular biology

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41594-019-0313-z

In the format provided by the authors and unedited.

PRMT5 methylome profiling uncovers a direct link to splicing regulation in acute myeloid leukemia

Aliaksandra Radzisheuskaya[1,2,3], Pavel V. Shliaha[4,5], Vasily Grinev[6], Eugenia Lorenzini[1,2], Sergey Kovalchuk[4,7], Daria Shlyueva[1,2], Vladimir Gorshkov[4], Ronald C. Hendrickson[5], Ole N. Jensen[4] and Kristian Helin[1,2,3]*

[1]Biotech Research and Innovation Centre, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen, Denmark. [2]The Danish Stem Cell Center, University of Copenhagen, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen, Denmark. [3]Cell Biology Program and Center for Epigenetics, Memorial Sloan Kettering Cancer Center, New York, NY, USA. [4]Department of Biochemistry and Molecular Biology, VILLUM Center for Bioanalytical Sciences, University of Southern Denmark, Odense, Denmark. [5]Microchemistry and Proteomics Core Facility, Memorial Sloan Kettering Cancer Center, New York, NY, USA. [6]Department of Genetics, Faculty of Biology, Belarusian State University, Minsk, Belarus. [7]Laboratory of Bioinformatic Methods for Combinatorial Chemistry and Biology, Shemyakin-Ovchinnikov Institute of Bioorganic Chemistry, Moscow, Russia.
*e-mail: helink@mskcc.org

Supplementary methods

TMT-labelled sample analysis by LC-MS/MS

50 µg of pooled 10plex TMT-labelled samples were analyzed by 2D-LC-MS/MS, using high pH RP fractionation as a first dimension and low pH RP chromatography as a second dimension. For high pH RP fractionation peptides were separated on an Ultimate3000 HPLC system (ThermoScientific) on ACQUITY CSH C18 1.7 µm column (300 µm X 100 mm) (Waters) with a linear 90 min gradient of acetonitrile in water at pH 9 at a flow-rate of 5 µL/min. Buffer A was H_2O, 20 mM ammonium formate, pH 9; buffer B was 80% ACN, 20% buffer A):

16.1	NC_Pump.%B =	16.5 [%]
19.1	NC_Pump.%B =	18.0 [%]
22.1	NC_Pump.%B =	19.3 [%]
25.1	NC_Pump.%B =	21.0 [%]
28.1	NC_Pump.%B =	22.7 [%]
31.1	NC_Pump.%B =	23.8 [%]
34.1	NC_Pump.%B =	25.0 [%]
37.1	NC_Pump.%B =	26.3 [%]
40.1	NC_Pump.%B =	27.4 [%]
43.1	NC_Pump.%B =	28.2 [%]
46.1	NC_Pump.%B =	29.0 [%]
49.1	NC_Pump.%B =	30.3 [%]
52.1	NC_Pump.%B =	31.5 [%]
55.1	NC_Pump.%B =	32.7 [%]
58.1	NC_Pump.%B =	34.0 [%]
61.1	NC_Pump.%B =	35.5 [%]
64.1	NC_Pump.%B =	37.5 [%]
67.1	NC_Pump.%B =	39.3 [%]
70.1	NC_Pump.%B =	42.0 [%]
73.1	NC_Pump.%B =	44.0 [%]
76.1	NC_Pump.%B =	46.0 [%]
79.1	NC_Pump.%B =	48.0 [%]
82.1	NC_Pump.%B =	51.0 [%]
85.1	NC_Pump.%B =	54.0 [%]
88.1	NC_Pump.%B =	58.5 [%]
91.1	NC_Pump.%B =	68.8 [%]
94.1	NC_Pump.%B =	95.0 [%]
100	NC_Pump.%B =	95.0 [%]
101	NC_Pump.%B =	2.0 [%]

Fraction collection timing was adjusted to collect different number of subfractions in different experiments using different second dimension LC systems and separation parameters. In all experiments 2 subfractions from different parts of the gradient were concatenated. 192

subfractions were collected for evoSep1 analysis and concatenated into 96 fractions: Subfraction 1 was combined with Subfraction 97, Subfraction 2 was combined with Subfraction 98, etc.) 60 subfractions (30 concatenated fractions) were collected for nanoAcquity UPLC analysis and 30 subfractions (15 fractions) - for Ultimate 3000 analysis. For nanoAcquity and Ultimate 3000 UPLC systems analysis the fractions were vacuum dried in a speedvac and resolubilized in 12 µL of 0.1% TFA, and 8 µL was used for the LC-MS/MS analysis. For evoSep 1 analysis the fractions were diluted with 130 µL of 0.1% TFA in water and directly loaded on the evoTips. Supplementary Table 1.3 details the methods used on the three LC systems.

All the data was acquired on the Orbitrap Fusion LUMOS platform using multi-notch selection method[1], as recommended by the manufacturer (using "TMT MS3" from the library of ready-to-use peptide quantitation methods). The parameters are shown below:

Global Settings

Use Ion Source Settings from Tune = False
Ion Source Type = NSI
Spray Voltage: Positive Ion (V) = 2900
Spray Voltage: Negative Ion (V) = 600
Infusion Mode (LC)= False
Sweep Gas (Arb) = 0
Ion Transfer Tube Temp (°C) = 275
APPI Lamp = Not in use
FAIMS Mode = Not Installed
Application Mode = Peptide
Default Charge State = 2
Advanced Peak Determination = False
Xcalibur AcquireX enabled for method modifications = False

Experiment 1

Cycle Time (sec) = 3

Scan MasterScan

MSn Level = 1
Use Wide Quad Isolation = True
Detector Type = Orbitrap
Orbitrap Resolution = 120K
Mass Range = Normal
Scan Range (m/z) = 375-1500
Maximum Injection Time (ms) = 50
AGC Target = 400000
Microscans = 1
RF Lens (%) = 30
Use ETD Internal Calibration = False
DataType = Profile
Polarity = Positive
Source Fragmentation = False
Scan Description =

Filter MIPS

MIPS Mode = Peptide

Filter ChargeState

Include charge state(s) = 2-7
Include undetermined charge states = False
Include charge states 25 and higher = False

Filter DynamicExclusion
Exclude after n times = 1
Exclusion duration (s) = 60
Mass Tolerance = ppm
Mass tolerance low = 10
Mass tolerance high = 10
Exclude isotopes = True
Perform dependent scan on single charge state per precursor only = False

Filter IntensityThreshold
Minimum Intensity = 5000
Relative Intensity Threshold = 0
Intensity Filter Type = IntensityThreshold
Maximum Intensity = 1E+20

Data Dependent Properties
Data Dependent Mode= Cycle Time

Scan Event 1

Scan ddMSnScan
MSn Level = 2
Isolation Mode = Quadrupole
Isolation Offset = Off
Isolation Window = 0.7
Reported Mass = Original Mass
Multi-notch Isolation = False
Scan Range Mode = Auto Normal
FirstMass = 120
Scan Priority= 1
ActivationType = CID
Collision Energy Mode = Fixed
Collision Energy (%) = 35
Activation Time (ms) = 10
Activation Q = 0.25
Multistage Activation = False
Detector Type = IonTrap
Ion Trap Scan Rate = Turbo
Maximum Injection Time (ms) = 50
AGC Target = 10000
Inject ions for all available parallelizable time = False
Microscans = 1
Use ETD Internal Calibration = False
DataType = Centroid
Polarity = Positive
Source Fragmentation = False
Scan Description =

Filter PrecursorSelectionRangeMSn
Mass Range = 400-1200
Range relative to parent mass (%) = 0-1000
Mass Tolerance Unit = mz

Filter PrecursorIonExclusion
Mass Tolerance is mz
Low 18
High 5

Filter IsobaricTagExclusion
Exclusion Mass Tolerance is Unknown
Low 0
High 0
Reagant Tag Type = TMT

Data Dependent Properties
Data Dependent Mode= Scans Per Outcome

Scan ddMSnScan
MSn Level = 3
Isolation Mode = Quadrupole
Isolation Window = 2
Isolation Offset = Off
Reported Mass = Original Mass
Multi-notch Isolation = True
MS2 Isolation Window (m/z) = 2
Number of Notches = 10
Scan Range Mode = Define m/z range
Scan Priority= 1
ActivationType = HCD
Collision Energy Mode = Fixed
Collision Energy (%) = 65
Detector Type = Orbitrap
Orbitrap Resolution = 50K
Scan Range (m/z) = 100-500
Maximum Injection Time (ms) = 105
AGC Target = 200000
Inject ions for all available parallelizable time = False
Microscans = 1
Use ETD Internal Calibration = False
DataType = Centroid
Polarity = Positive
Source Fragmentation = False
Scan Description =

The only difference in the MS parameters between the three hyperfractionation strategies (3 LC systems used for the analysis) was the dynamic exclusion time necessary to account for different peak width: 15 sec for evoSep, 40 sec for nanoAcquity and 60 sec for Ultimate 3000.

Proteomics data analysis

Data were analyzed in Proteome Discoverer 2.3 software. All the files from all acquisitions (hyperfractionations and enrichment) were searched together. A database search was performed with Mascot 2.3.2 using Homo Sapiens UniProt database containing only reviewed entries and canonical isoforms (retrieved on 06/11/2017). Oxidation (M), Deamidated (NQ), Dimethyl (R), Methyl (R) were set as variable modifications, while Carbamidomethyl (C), TMT6plex (K), and TMT6plex (N-term) were specified as fixed modifications. A maximum of four missed cleavages were permitted (such a high number was chosen due to lower trypsin efficiency at methylated Arg sites). The precursor and fragment mass tolerances were 10 ppm and 0.6 Da, respectively. Peptides were validated by Mascot percolator with a 0.01 posterior error probability (PEP) threshold. ptmRS algorithm was used to validate Arg methylation position using standard settings for CID data (PhosphoRS Mode was disabled, fragment mass tolerance was 0.5 Da with no mass accuracy correction, the diagnostic ions use was enabled and no neutral loss ions were considered). The quantification results of peptide spectrum matches were combined into peptide-level quantitation, which in turn was converted into protein quantitation using danteR package[2]. Quantitation profiles of the methylated peptides

were normalized on the profiles of their corresponding protein expression (example normalization is presented in Supplementary Table 1.5).

Peptide synthesis

Peptides were produced on Intavis ResPep (expected yield of 5 μmole) instrument. Solutions used: Fmoc-aa solution (650 mM solution in DMF (Fmoc-Pro-OH and Fmoc-Phe-OH were dissolved in NMP)), Activator (600 mM HBTU in DMF), Base (N-Methylmorpholine in DMF 0.81:1 v/v), Capping solution (5% acetic anhydrate in DMF), Deprotection solution (20% Piperidine in DMF), ethanol, DCM, Fmoc-Gly-Wang resin (Sigma 47659) 10 mg per tube. Program used:

START ROUTINE:

1) RinseNeedle (2500 / 2500 µl)
2) WashColumns (150 µl, Reservoir->Peptides, 5x)
3) MoveToZone (pierceSepta)
4) Deprotection (120 µl, Piperidine->Peptides)
5) Deprotection (120 µl, Piperidine->Peptides)
6) WashColumns (150 µl, Reservoir->Peptides, 8x)

SYNTHESIS CYCLES:

7) Coupling (25.6 (activator)+7 (base)+5 (DMF)+21.7 (derivative)->Peptides – 30 min)
8) Coupling (25.6 (activator)+7 (base)+5 (DMF)+21.7 (derivative)->Peptides – 30 min)
9) Coupling (25.6 (activator)+7 (base)+5 (DMF)+21.7 (derivative)->Peptides – 30 min)
10) Capping (100 µl, Capping->Peptides– 30 min)
11) WashColumns (130 µl, Reservoir->Peptides, 8x)
12) Deprotection (120 µl, Piperidine->Peptides)
13) Deprotection (120 µl, Piperidine->Peptides)
14) WashColumns (130 µl, Reservoir->Peptides, 8x)

FINAL WASHES

15) WashColumns (150 µl, Reservoir->Peptides)
16) WashColumns (150 µl, DCMwash->Peptides)
17) WashColumns (150 µl, Reservoir->Peptides)
18) Deprotection (120 µl, Piperidine->Peptides)
19) Deprotection (120 µl, Piperidine->Peptides)
20) Deprotection (120 µl, Piperidine->Peptides)

21) WashColumns (150 µl, Reservoir->Peptides, 3x)

22) WashColumns (150 µl, Ethanol->Peptides, 3x)

23) WashColumns (150 µl, DCMwash->Peptides, 3x)

24) RinseNeedle (2500 / 2500 µl)

The sequences of the synthesized peptides are listed in Supplementary Table 1.6.

Bioinformatics analysis of the RNA-Seq data

1. Quality assessment and pre-processing of the RNA-Seq raw data

A comprehensive RNA-Seq data quality assessment and pre-processing of the raw data was performed using the R/Bioconductor library ShortRead v.1.38.0[3] and the standard R infrastructure. The main in-laboratory developed high-level R functions are available at GitHub repository:

https://github.com/VGrinev/transcriptome-analysis/blob/master/TranscriptsFeatures.

2. Alignment of the short RNA-Seq reads against the reference genome

GRCh38/hg38 reference assembly of the human genome was downloaded as twoBit file from the FTP server of the UCSC Genome Browser. It was then converted to a standard FASTA format with *twoBitToFa* utility[4]. A hash table for the reference genome was built with the function *buildindex* from the R/Bioconductor library Rsubread v.1.22.3[5]. At this step, 16-mers subreads were extracted in every 3 bases from the reference genome and the threshold 24 was used to exclude highly repetitive subreads from the created hash table.

A global alignment of RNA-Seq reads against the reference genome was carried out with the function *subjunc* from the R/Bioconductor library Rsubread v.1.22.3[6] and the created hash table. This function implements a seed-and-vote mapping paradigm for a fast and accurate alignment[6]. At this step, we used the default settings of the function *subjunc* and collected only uniquely mapped reads with the maximum of 3 mismatched bases in the alignment. The resulting BAM files were sorted with the function *sortBam* and indexed with the function *indexBam* both from the R/Bioconductor library Rsamtools v.1.24.0[7].

3. Development a list of non-overlapping genomic bins (fragments)

First, the annotations of the human genes were downloaded from the Ensembl database. These annotations were converted into an object of a class TranscriptDb[8] and saved as a local SQLite database. All the subsequent manipulations with genomic intervals were performed using the genomic ranges infrastructure[8].

Next, the genomic coordinates of the annotated retained introns were calculated as follows. For each gene, the genomic coordinates of exons and introns were extracted from a

TranscriptDb object and intersected using the function *findOverlaps* from the R/Bioconductor library GenomicRanges v.1.32.3[9]. The intronic intervals that completely fell into the coordinates of exon(-s) of the same gene were selected. These intervals were subsequently intersected with the remaining exons of the gene and, if necessary, disjoined and dropped out due to overlapping. Additionally, the intervals shorter than 75 nucleotides or duplicated intervals were removed from the final list. We called these intervals annotated retained introns, since they are already present in the Ensembl annotation database. Moreover, we further sub-divided these introns into four sub-groups: i) annotated retained introns that completely fall into the alternative first exon (AnnoRI_FIRST), ii) annotated retained introns that are flanked by the internal exons (AnnoRI_INTERNAL), iii) annotated retained introns that completely fall into the alternative last exon (AnnoRI_LAST), and iv) introns that can be inferred as retained because of the overlap with one-exon transcripts (AnnoRI_OneExonTranscript).

Third, the function *intronicParts* of the R/Bioconductor library GenomicFeatures v.1.32.0[10] was used to extract non-overlapping intronic bins from a TranscriptDb object. Extracted intronic bins were disjoined and dropped out against exons of one- and multi-exons genes (including genes of rRNAs, tRNAs, miRNAs, miscRNAs, ribozymes, vaultRNAs, sRNAs, snRNAs, scaRNAs, scRNAs and snoRNAs). Additionally, any intronic bins shorter than 75 nucleotides were removed. We called these intronic bins canonical introns. The final list of such introns was extended with the annotated retained introns, sorted, indexed, assigned with genes information and converted into an object of a class GRanges.

Fourth, the function *exonicParts* of the R/Bioconductor library GenomicFeatures v.1.32.0[11] was used to extract non-overlapping exonic bins from a TranscriptDb object. Extracted genomic bins were disjoined and dropped out against genomic coordinates of the annotated retained introns, and genomic bins shorter than 10 nucleotides were removed. The final list of the genomic bins was sorted, indexed, annotated with genes information and converted into an object of a class GRanges.

Finally, all the above-mentioned GRanges objects were joined into GRangesList and used in the downstream analysis.

4. Development a list of exon clusters

For each gene, the genomic coordinates of exons were retrieved from Ensembl annotations. These coordinates were intersected and joined into overlapping groups called exon clusters. The exon clusters shorter than 10 nucleotides were removed, and the final list of genomic

intervals was sorted, assigned with genes information and converted into an object of a class GRanges.

5. Read summarisation

We used the function *featureCounts* from the R/Bioconductor library Rsubread v.1.22.3[5,12] to assign the mapped RNA-Seq reads to the genomic features in case of exonic and/or intronic bins or to the meta-features (genes) in case of exon clusters. Each read pair was counted in unstranded mode with the minimum 1 base overlapping an exonic bin or exon cluster and minimum 5 bases overlapping an intronic bin.

6. In silico identification of retained introns

First, the primary count matrix of intronic bins (see sections 3 and 5) was loaded into the R workspace and an effective length for each intron was calculated using the wgEncodeCrgMapabilityAlign75mer mapability table[13] from the UCSC Genome Browser[4]. During this step, positions of the non-unique 75-mer alignments and 5 nucleotides from each end of the intron were summed and then subtracted from the original intron's length to produce mapability-adjusted intron length.

Second, the primary count matrix of the intronic bins was filtered against the non-expressed genes, intron effective length less than 75 nucleotides and one-bin genes.

Third, a variance stabilizing transformation based on the square root of the intron effective length adjusted by the RNA-Seq read length was used to weight individual introns[14,15]. The sum of intronic reads per gene in each RNA-Seq sample was then partitioned and allocated to each intron proportional to its weight. This led to an *in silico* null model sample, one corresponding to each of the original RNA-Seq samples.

Fourth, differential analysis was carried out to determine introns enriched in the observed reads compared to the *in silico* expected reads (if all the introns within a gene are present at equal levels). We used standard DESeq2[16] and edgeR-limma[17,18] pipelines at this step. Herewith, we discretized the null distributions for the first approach, since DESeq2 uses negative binomial generalized linear modelling, and we loaded the null distributions as it they are for the edgeR-limma pipeline.

Finally, the results of the differential analysis were parsed and filtered. We selected only introns that passed a false discovery rate (FDR) adjusted p-value threshold of 0.01, fold change threshold of 2 and a required minimum of 20 reads per 100 nucleotides of the effective length of an intron averaged over all the original RNA-Seq samples. These introns were called

in silico detected retained introns, or simply retained introns. The primary count matrix of intronic bins was then reduced to a list of retained introns and it was added to the primary count matrix of the exonic bins.

7. Inferring of the differentially used exons (diffUEs)

7.1. Identification of diffUEs with DEXSeq

First, the primary count matrix of the exonic bins was loaded into the R workspace and filtered against non-expressed genes, one-bin genes and too low sequencing depth. The filtered count matrix was subsequently used to create a flattened GTF file and it was wrapped (together with a flattened GTF file, sample annotations and experimental design) into an object of a class DEXSeqDataSet[19].

Second, the size of each RNA-Seq library was normalized using the "median ratio method"[20] and dispersion estimates were obtained using the function *estimateDispersions* from the R/Bioconductor library DESeq2 v.1.20.0[16,21]. Third, the diffUEs were determined using the functions *testForDEU* and *estimateExonFoldChanges* from the R/Bioconductor library DESeq2 v.1.20.0[21] in the default mode. At last, the final results were summarized using the function *DEXSeqResults* from the R/Bioconductor library DESeq2 v.1.20.0[21].

7.2. Identification of diffUEs with function diffSplice

First, the primary count matrix of exonic bins was subjected to filtering against the non-expressed genes, one-bin genes and too low sequencing depth and it was wrapped (together with the sample information) into a DGEList object[22].

Second, to calculate effective sizes of RNA-Seq libraries, the scaling factors were estimated using the "trimmed mean of M-values" method.

Third, by applying the calculated scaling factors, the count data were converted into counts per million, or CPM, and logarithmically transformed, the mean-variance relationship was estimated, and the appropriate observational-level weights were calculated using the voom algorithm[23].

Fourth, the multiple simple linear models were fitted to the normalized count matrix by least squares method using the function *lmFit* from the R/Bioconductor library limma v.3.36.1.

Fifth, contrast coefficients (logarithms for base two of fold changes, or $\log_2$ FC, between the treatment conditions) were calculated and loaded into the function *diffSplice*[17,24]. This function calculates the difference between the $\log_2$ FC for a given exon versus the average $\log_2$ FC for all the other exons for the gene of interest. In other words, this function tests for differential usage of exons for each gene and for each treatment condition. Finally, from

moderated t-statistics, p-values were adjusted for multiple testing with the method of Benjamini Y. and Hochberg Y., which controls the expected FDR below the specified value.

8. Identification of exon-exon junctions (EEJs)

All possible variants of EEJs were identified according to Liao et al.[5]. The resulting BED files were parsed and converted into the primary count matrix of EEJs with an in-laboratory developed R code. This matrix included a full list of identified EEJs with the respective genomic coordinates and a number of reads supporting each exon-exon junction in every sample.

9. Identifying the differentially used exon-exon junctions (diffEEJs)

9.1. Identification of diffEEJs with function diffSplice

The primary count matrix of EEJs was subjected to filtering against too low sequencing depth and it was wrapped (together with the sample information) into a DGEList object. All the subsequent steps of the analysis were carried out in accordance with subsection 7.2, but at the level of EEJs.

9.2. Identification of diffEEJs using functionality of JunctionSeq library

First, the overall quality of the BAM files was assessed with Picard v.2.9.0 (http://broadinstitute.github.io/picard/) and low-quality reads were removed using in-laboratory developed R code. Second, the flattened GFF file was created using toolset QoRTs. This file was based on the Ensemble annotations of the human genome and included all the exons, annotated and novel EEJs. Third, reads counts were generated by QoRTs. At this step, we counted all the reads mapped to exons, annotated or novel EEJs with minimum mapping quality of 30. Fourth, the diffEEJs were identified by the sequential application of two functions *runJunctionSeqAnalyses* and *writeCompleteResults* in the default mode to reads counts. These functions are part of the R/Bioconductor library JunctionSeq v.1.10 and they use DEXSeq statistical infrastructure to detect diffEEJs. Finally, output results were parsed and adjusted to the format of *diffSplice* output by in-laboratory developed R code.

10. Classification of EEJs according to the modes of alternative splicing

Our classifier of EEJs is based on the idea of hypothetical "non-alternative" precursor of RNA, or hnapRNA. hnapRNA is an RNA molecule that would have turned out if the gene had only one transcription start site, if there were no alternative splice sites, if there was no alternative splicing and if there was only one transcription termination site. In other words, hnapRNA is a generalization of all the RNA isoforms produced by the gene.

For each gene, the structure of the hnapRNA was calculated using Ensembl models of human genes. We clustered exons of the gene of interest into overlapping groups with the exception

of retained introns, alternative 5’ and/or 3’ terminal exons. The outer boundaries of the resulting exon clusters were recorded as genomic coordinates of exons of the hnapRNA. The list of these coordinates was extended with coordinates of retained introns, alternative 5’ and/or 3’ terminal exons and it was converted into an object of a class GRanges.

Next, the genomic coordinates of EEJs were intersected with the coordinates of the features of the hnapRNA, and the mode of each EEJ was determined. According to our approach, all the EEJs were classified into eight modes of alternative splicing:

i) canonical event, if the coordinates of the empirical event exactly match the model event,

ii) alternative 5' splice site, if only the 3’ splice site of the empirical event exactly matches the respective model site,

iii) alternative 3' splice site, if only the 5’ splice site of the empirical event exactly matches respective model site,

iv) alternative both splice sites (intron isoform), if both splice sites of the empirical event do not match splice sites of respective model event,

v) skipped cassette exon(-s), if the empirical event includes skipping one or more exons of the model,

vi) alternative first exon, if the 5’ splice site of the empirical event exactly matches the 3’ end of alternative first exon in model,

vii) alternative last exon, if the 3’ splice site of the empirical event exactly matches the 5’ end of alternative last exon in model.

11. Reference-based transcriptome assembly

First, for each sample of RNA, we used Cufflinks[25] and the respective *subjunc*-generated BAM file to assemble the alignments into a parsimonious set of transcripts. Herewith, Cufflinks was supplied with i) Ensembl annotation of the human genome to guide RABT assembly, ii) a GTF file containing annotated human rRNA and mitochondrial genes to mask these genomic features during estimation of transcripts abundance, iii) complete sequence of the human genome in multiFASTA format to bias correction during the estimation of transcripts abundance, and iv) a minimal isoform fraction threshold assigned to 0.05.

Second, individual Cufflinks assembled transcriptomes were merged into one consolidated set of transcripts with Cuffmerge[26]. This set of transcripts was filtered against i) unstranded transcripts, ii) too short transcripts (<300 nucleotides), iii) transcripts with too short exon(-s) (<25 nucleotides), iv) transcripts with too short intron(-s) (<50 nucleotides), and vi) transcripts with low abundance (fragments per kilobase of transcript per million mapped reads, or FPKM, below 1). Filtration was controlled by in-laboratory developed R code.

Third, the consolidated and filtered set of transcripts was submitted to Cuffdiff for the simultaneous calculation of the transcript abundance and differential expression. Cuffdiff was provided with a GTF file containing annotated human rRNA and mitochondrial genes and a multiFASTA file with complete sequence of the human genome, and it was run in default mode except for the minimal isoform fraction threshold that was assigned to 0.05. Finally, for the fast retrieving of the data and easy subsequent manipulations, the main outcomes of Cuffdiff were parsed, converted into an object of a class TranscriptDb[8] and saved as a local SQLite database.

12. Analysis of differential gene expression

12.1. Identification of differentially expressed genes with Cuffdiff

We used Cuffdiff differential expression tests data (see section 11 above) to identify differential expression at transcript or gene levels between experimental conditions. Herewith, only transcripts or genes with at least 2-fold changes in expression and q-value below 0.05 were annotated as differentially expressed.

12.2. Identification of differentially expressed genes with DESeq2

First, the mapped RNA-Seq reads were assigned to the genomic meta-features (genes) as described in section 5. Second, the resulting count matrix was subjected to filtering against too low sequencing depth and it was wrapped (together with the sample information) into a DESeqDataSet object. Third, differentially expressed genes were identified using the functions *DESeq* and *results* from the R/Bioconductor library DESeq2 v.1.16.1. These functions were run in default mode and according to the standard DESeq2 pipeline. Finally, results were parsed and genes with at least 2-fold changes in expression and q-value below 0.05 were annotated as differentially expressed.

12.3. Identification of differentially expressed genes with edgeR-limma

First, the mapped RNA-Seq reads were assigned to the genomic meta-features as described in section 5. Second, the resulting count matrix was subjected to filtering against too low sequencing depth and it was wrapped (together with the sample information) into a DGEList object[22]. Third, to calculate an effective size of each RNA-Seq library, the scaling factors were estimated using the "trimmed mean of M-values" method. Fourth, by applying the calculated scaling factors, the count data were converted into CPM and logarithmically transformed, the mean-variance relationship was estimated, and the appropriate observational-level weights were calculated using the voom algorithm.

Fifth, the multiple simple linear models were fitted to the normalized count matrix by least squares method using the function *lmFit* from the R/Bioconductor library limma v.3.36.1[17,24].

Sixth, $\log_2$ FC coefficients and empirical Bayes statistics were calculated using respective functions from R/Bioconductor libraries edgeR v.3.22.3[18,27] and limma v.3.34.9[17,24]. Finally, results were parsed and genes with at least 2-fold changes in expression and the q-value below 0.05 were annotated as differentially expressed.

13. Enrichment test

We used up-to-date ODO and GAF files from the Gene Ontology Consortium[28,29] to develop a comprehensive list of the reference functional gene sets. From this list, we selected the gene sets containing ten or more members for downstream analysis. Next, two-sided Fisher's exact test was used to find out the under- and/or over-represented query gene set(-s) among the reference gene sets. Query results were parsed and under- or over-represented gene sets that passed FDR adjusted p-value threshold of 0.05 were collected. Finally, Cytoscape plug-in EnrichmentMap[30] was used to handle gene-set redundancy and hierarchical visualization of the enrichment results.

14. Motif enrichment analysis

We collected sequences of experimentally verified binding sites that were recognized by SRSF1[31-33], SRSF2[31-33] and SRSF3[34-36] splicing-related proteins. For each protein and the respective set of sequences, we performed a motif search by the discriminative motif discovery algorithm motifRG[37] against the background set of randomly extracted human intronic and exonic sequences. The primary motif was refined by the function *refinePWMMotif* from the R/Bioconductor library motifRG v.1.18.0[37] with the default settings and was converted into the $\log_2$ position weight matrix with the correction against the background nucleotides frequency. Occurrence of a motif in the sequence of interest was determined by the function *matchPWM* from the R/Bioconductor library Biostrings v.2.42.0[38] using either dynamic or fixed thresholding, as indicated in Figure legends. For dynamic thresholding, the threshold was adjusted to maximize the difference between the normalized motif frequencies in the compared data sets and the motif occurrence was normalized relative to the length of the analysed sequence. For fixed thresholding, we used the 99^{th} quantile of the motif weight distribution as a threshold in the identification of the true motif occurrence.

Additionally, we collected oligomeric sequences that were bound by splicing proteins SFPQ[39-42] and SRSF7[34,43,44]. We were not able to calculate position weight matric of the motifs for these proteins due to a limited number of sequences of the experimentally verified binding sites. For this reason, we used an alternative approach in the assessment of the strength of binding sites for the mentioned above splicing proteins, as proposed by Murray et al.[45]:

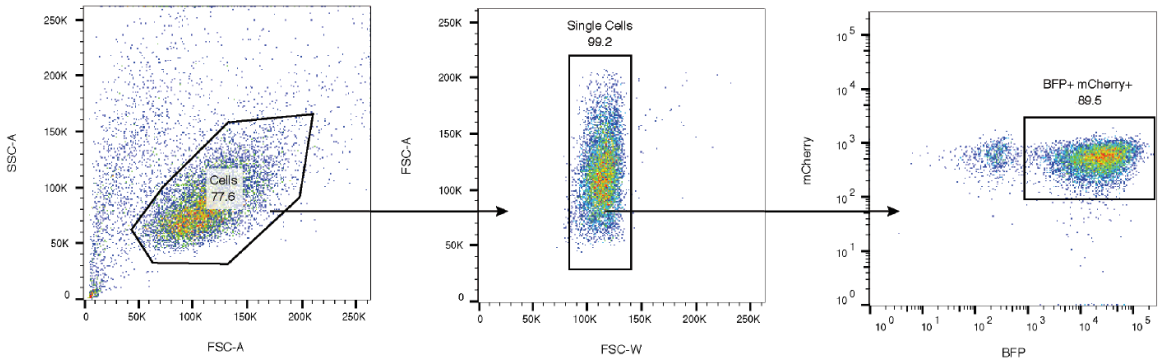
$$BSS = \frac{\sum_{i=0}^{i=L-k+1} ln\,(4^k f_{n_i})}{L-k+1},$$

where BSS is the strength of binding sites, L is the length of the sequence of interest, k is the length of an oligomer found in the sequence of interest, f_n represents the frequency (within the set of sequences of the experimentally verified binding sites for a given splicing protein) of the oligomer found at the position i in the sequence of interest, and $ln(4^k f_{n_i})$ is a log-odds representation of the degree to which the particular oligomer was enriched within the set of sequences of the experimentally verified binding sites for a given splicing protein. As proposed, we counted only the frequency of all the possible pentamers in the sequence of interest and used the frequency of pentamers from the set of sequences of the experimentally verified binding sites for a given splicing protein as the reference[46].

15. RIP-Seq data analysis

Quality assessment and pre-processing of the RIP-Seq raw data was carried out as described in section 1. The mapping of RIP-Seq reads to the reference genome and summarization of reads were performed as described in section 2 and section 5, respectively. Using the resulting count matrix, the differential abundance of the SRSF1-bound RNA molecules was determined using the functionality of the R/Bioconductor libraries edgeR v.3.22.3[27,47] and limma v.3.34.9[17,24], as described in section 12.3 for differential gene expression.

Example flow cytometry gating strategy for competition assays



References

1. McAlister, G. C. *et al.* MultiNotch MS3 enables accurate, sensitive, and multiplexed detection of differential expression across cancer cell line proteomes. *Anal. Chem.* **86,** 7150–7158 (2014).
2. Polpitiya, A. D. *et al.* DAnTE: a statistical tool for quantitative analysis of -omics data. *Bioinformatics* **24,** 1556–1558 (2008).
3. Morgan, M., Pages, H., Obenchain, V. & Hayden, N. Rsamtools: Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import. R package 'Rsamtools' version 1.24.0. (2016).
4. Speir, M. L. *et al.* The UCSC Genome Browser database: 2016 update. *Nucleic Acids Res.* **44,** D717–725 (2016).
5. Liao, Y., Smyth, G. K. & Shi, W. The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Res.* **41,** e108 (2013).
6. Liao, Y., Smyth, G. K. & Shi, W. The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Res.* **41,** e108 (2013).
7. Morgan, M., Pages, H., Obenchain, V. & Hayden, N. Rsamtools: Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import. R package 'Rsamtools' version 1.24.0. (2016).
8. Lawrence, M. *et al.* Software for computing and annotating genomic ranges. *PLoS Comput. Biol.* **9,** e1003118 (2013).
9. Aboyoun, P., Pages, H. & Lawrence, M. Representation and manipulation of genomic intervals and variables defined along a genome. R package 'GenomicRanges' version 1.32.3. (2018).
10. Carlson, M. *et al.* Tools for making and manipulating transcript centric annotations. R package 'GenomicFeatures' version 1.32.0. (2018).
11. Carlson, M. *et al.* Tools for making and manipulating transcript centric annotations. R package 'GenomicFeatures' version 1.32.0. (2018).
12. Liao, Y., Smyth, G. K. & Shi, W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* **30,** 923–930 (2014).
13. Derrien, T. *et al.* Fast computation and applications of genome mappability. *PLoS ONE* **7,** e30377 (2012).
14. Boutz, P. L., Bhutkar, A. & Sharp, P. A. Detained introns are a novel, widespread class of post-transcriptionally spliced introns. *Genes & Development* **29,** 63–80 (2015).
15. Braun, C. J. *et al.* Coordinated Splicing of Regulatory Detained Introns within Oncogenic Transcripts Creates an Exploitable Vulnerability in Malignant Glioma. *Cancer Cell* **32,** 411–426.e11 (2017).
16. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15,** 550 (2014).
17. Ritchie, M. E. *et al.* limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* **43,** e47 (2015).
18. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26,** 139–140 (2010).
19. Anders, S., Reyes, A. & Huber, W. Detecting differential usage of exons from RNA-seq data. *Genome Research* **22,** 2008–2017 (2012).
20. Anders, S. & Huber, W. Differential expression analysis for sequence count data. *Genome Biology* **11,** R106 (2010).
21. Love, M. I., Anders, S. & Huber, W. Differential gene expression analysis based on the negative binomial distribution. R package 'DESeq2' version 1.20.0. (2018).

22. Anders, S. *et al.* Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nat Protoc* **8,** 1765–1786 (2013).
23. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology* **15,** R29 (2014).
24. Smyth, G. *et al.* Linear models for microarray data. R package 'limma' version 3.36.2.
25. Trapnell, C. *et al.* Transcript assembly and abundance estimation from RNA-Seq reveals thousands of new transcripts and switching among isoforms. *Nat Biotechnol* **28,** 511–515 (2010).
26. Trapnell, C. *et al.* Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat Protoc* **7,** 562–578 (2012).
27. Chen, Y. *et al.* Empirical analysis of digital gene expression data in R. R package 'edgeR' version 3.22.3.
28. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics* **25,** 25–29 (2000).
29. Consortium, T. G. O. Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res.* **45,** D331–D338 (2017).
30. Merico, D., Isserlin, R., Stueker, O., Emili, A. & Bader, G. D. Enrichment map: a network-based method for gene-set enrichment visualization and interpretation. *PLoS ONE* **5,** e13984 (2010).
31. Cartegni, L., Wang, J., Zhu, Z., Zhang, M. Q. & Krainer, A. R. ESEfinder: A web resource to identify exonic splicing enhancers. *Nucleic Acids Res.* **31,** 3568–3571 (2003).
32. Cartegni, L., Hastings, M. L., Calarco, J. A., de Stanchina, E. & Krainer, A. R. Determinants of exon 7 splicing in the spinal muscular atrophy genes, SMN1 and SMN2. *Am. J. Hum. Genet.* **78,** 63–77 (2006).
33. Smith, P. J. *et al.* An increased specificity score matrix for the prediction of SF2/ASF-specific exonic splicing enhancers. *Hum. Mol. Genet.* **15,** 2490–2508 (2006).
34. Galiana-Arnoux, D. *et al.* The CD44 alternative v9 exon contains a splicing enhancer responsive to the SR proteins 9G8, ASF/SF2, and SRp20. *J. Biol. Chem.* **278,** 32943–32953 (2003).
35. Gonçalves, V., Matos, P. & Jordan, P. Antagonistic SR proteins regulate alternative splicing of tumor-related Rac1b downstream of the PI3-kinase and Wnt pathways. *Hum. Mol. Genet.* **18,** 3696–3707 (2009).
36. Jang, H. N. *et al.* Exon 9 skipping of apoptotic caspase-2 pre-mRNA is promoted by SRSF3 through interaction with exon 8. *BBA - Gene Regulatory Mechanisms* **1839,** 25–32 (2014).
37. Yao, Z. *et al.* Discriminative motif analysis of high-throughput dataset. *Bioinformatics* **30,** 775–783 (2014).
38. Pages, H., Aboyoun, P., Gentleman, R. & DebRoy, S. Biostrings: String objects representing biological sequences, and matching algorithms. R package version 2.42.0. *Bioconductor* (2014).
39. Buxadé, M., Morrice, N., Krebs, D. L. & Proud, C. G. The PSF·p54nrb Complex Is a Novel Mnk Substrate That Binds the mRNA for Tumor Necrosis Factor α. *J. Biol. Chem.* **283,** 57–65 (2008).
40. Hall-Pogar, T., Liang, S., Hague, L. K. & Lutz, C. S. Specific trans-acting proteins interact with auxiliary RNA polyadenylation elements in the COX-2 3'-UTR. *RNA* **13,** 1103–1115 (2007).
41. Melton, A. A., Jackson, J., Wang, J. & Lynch, K. W. Combinatorial control of signal-induced exon repression by hnRNP L and PSF. *Mol. Cell. Biol.* **27,** 6972–6984 (2007).
42. Motta-Mena, L. B., Heyd, F. & Lynch, K. W. Context-dependent regulatory

mechanism of the splicing factor hnRNP L. *Mol. Cell* **37,** 223–234 (2010).
43. Schaal, T. D. & Maniatis, T. Selection and characterization of pre-mRNA splicing enhancers: identification of novel SR protein-specific enhancer sequences. *Mol. Cell. Biol.* **19,** 1705–1719 (1999).
44. Venables, J. P. *et al.* Up-regulation of the ubiquitous alternative splicing factor Tra2beta causes inclusion of a germ cell-specific exon. *Hum. Mol. Genet.* **14,** 2289–2303 (2005).
45. Murray, J. I., Voelker, R. B., Henscheid, K. L., Warf, M. B. & Berglund, J. A. Identification of motifs that function in the splicing of non-canonical introns. *Genome Biology* **9,** R97 (2008).
46. Murray, J. I., Voelker, R. B., Henscheid, K. L., Warf, M. B. & Berglund, J. A. Identification of motifs that function in the splicing of non-canonical introns. *Genome Biology* **9,** R97 (2008).
47. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26,** 139–140 (2010).