Combining automated processing and customized analysis for large-scale sequencing data

Michael Kluge

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

eingereicht von
Michael Kluge
aus Ebersberg

München, den 16.10.2020

Erstgutachter: Prof. Dr. Caroline C. Friedel
Zweitgutachter: Prof. Dr. Sven Rahmann
Drittgutachter: Prof. Dr. Dominik Heider
Tag der mündlichen Prüfung: 01.03.2021

Eidesstattliche Versicherung

Hiermit erkläre ich, Michael Kluge, an Eides statt, dass die vorliegende Dissertation ohne unerlaubte Hilfe gemäß Promotionsordnung vom 12.07.2011, § 8, Abs. 2 Pkt. 5, angefertigt worden ist.

München, den 16.10.2020

. . . . . .
Michael Kluge

Acknowledgements

At this point, I would like to thank everyone who supported me in any way during my time as a doctoral student.

First of all, I want to thank Prof. Dr. Caroline C. Friedel for giving me the opportunity to work on interesting topics, allowing me to pursue my ideas, helping me to improve my drafts, and most importantly for her valuable advice. Additionally, I want to express my gratitude to Prof. Dr. Dirk Eick and Dr. Dalibor Blazek for their helpful input during our scientific cooperation. I also want to thank their employees Dr. Michaela Rohrmoser and Dr. Anil P. Chirackal Manavalan for answering my questions about experimental details and specific biological processes. Moreover, I am grateful to Prof. Dr. Sven Rahmann and Prof. Dr. Dominik Heider for reviewing my thesis and to Prof. Dr. Andreas Butz and Prof. Dr. Matthias Schubert for being part of my dissertation committee.

I also want to thank Prof. Dr. Ralf Zimmer, Prof. Dr. Volker Heun and all my colleagues for all the interesting presentations and discussions, but also for off-topic conversations and non-work-related activities. In particular, I want to thank Dr. Csaba Gergely for learning me how to play squash and Dr. Luisa F. Jimenez-Soto for countless conversations and cheering me up. Furthermore, I would like to thank Frank Steiner for excellently administrating our computer systems and Franziska Schneider for helping with administrative matters and always having an open ear.

Finally, I would like to thank my friends and family for always supporting me. In particular, I owe special thanks to my wife Sonja for all her love, support and encouragement, especially in the past few months.

Zusammenfassung

Die umfassende Anwendung von Hochdurchsatzmethoden in den Biowissenschaften hat die damit verbundene Datenanalyse vor erhebliche Herausforderungen gestellt. Oft müssen viele verschiedene Schritte auf eine große Anzahl von Proben angewandt werden. Dabei können Workflow-Management-Systeme Wissenschaftler durch die automatisierte Ausführung entsprechender Analyse-Workflows unterstützen. Der erste Teil dieser kumulativen Dissertation konzentriert sich auf die Entwicklung von Watchdog, einem neuartigen Workflow-Management-System zur automatisierten Analyse umfangreicher experimenteller Daten. Zu den Hauptfunktionen von Watchdog gehören die einfache Verarbeitbarkeit von Replikaten, die Unterstützung verteilter Computersysteme, eine anpassbare Fehlererkennung und die Möglichkeit manuell in die Workflowausführung einzugreifen. Zudem ermöglicht eine grafische Benutzeroberfläche die Erstellung von Workflows ohne Programmiererfahrung, wobei einen vordefinierter Satz von Tools verwenden werden kann. Weiterhin erlaubt eine Community-Sharing-Plattform Wissenschaftlern, Tools und Workflows einfach mit anderen zu teilen. Darüber hinaus sind Methoden implementiert, um die Ausführung unterbrochener oder veränderter Workflows fortzusetzen und Software durch den Einsatz von Paketmanagern und Containervirtualisierung automatisiert bereitzustellen.

Mit Watchdog haben wir Standardanalyse-Workflows für typische Arten von biologischen Hochdurchsatz-Experimenten, wie RNA-seq und ChIP-seq, implementiert. Obwohl sie sich leicht auf neue Datensets desselben Typs anwenden lassen, stoßen solche Workflows irgendwann an ihre Grenzen, weswegen zur Klärung spezifischer Fragen angepasste Methoden erforderlich sind. Daher konzentriert sich der zweite Teil dieser Dissertation auf die Anwendung von Standardanalyse-Workflows kombiniert mit der Entwicklung anwendungsspezifischer bioinformatischer Methoden, um Fragen zu beantworten, die für unsere biologischen Kooperationspartner von Interesse sind. Die erste Studie beschäftigt sich mit der Identifizierung des Bindungsmotivs des Transkriptionsfaktors *ZNF768*, das aus zwei Ankerregionen besteht, die durch eine variable Linkerregion verbunden sind. Da Standard-Motivfindungsmethoden die Anker der Motive nur separat erkannten, wurde eine maßgeschneiderte Methode zur Bestimmung des bipartiten Bindemotivs entwickelt. Die zweite Studie befasst sich mit der Wirkung von *CDK12*-Hemmung auf die Transkription. Die aus der Standard-RNA-seq-Analyse erhaltenen Ergebnisse zeigten eine erhebliche Verkürzung vieler Transkripte nach Hemmung von *CDK12*. Wir haben daher eine neue Methode entwickelt, um den Grad der Transkriptverkürzung zu quantifizieren. Darüber hinaus wurde ein maßgeschneidertes Meta-Gen-Analyse-Framework entwickelt, um die Progression der RNA-Polymerase II unter Verwendung von ChIP-seq Daten zu modellieren. Dies zeigte, dass Hemmung von *CDK12* einen RNA-Polymerase II Prozessivitätsdefekt verursacht, der ursächlich für die beobachtete Transkriptverkürzung ist.

Zusammenfassend stellen die in dieser Arbeit entwickelten Methoden sowohl allgemeine Beiträge zur Analyse von Hochdurchsatz-Sequenzierungsdaten dar als auch Werkzeuge um spezifische Fragen bezüglich der Bindung von Transkriptionsfaktoren und der Regulation der verlängernden RNA-Polymerase II zu beantworten.

Summary

Extensive application of high-throughput methods in life sciences has brought substantial new challenges for data analysis. Often many different steps have to be applied to a large number of samples. Here, workflow management systems support scientists through the automated execution of corresponding large analysis workflows. The first part of this cumulative dissertation concentrates on the development of Watchdog, a novel workflow management system for the automated analysis of large-scale experimental data. Watchdog's main features include straightforward processing of replicate data, support for distributed computer systems, customizable error detection and manual intervention into workflow execution. A graphical user interface enables workflow construction using a pre-defined toolset without programming experience and a community sharing platform allows scientists to share toolsets and workflows efficiently. Furthermore, we implemented methods for resuming execution of interrupted or partially modified workflows and for automated deployment of software using package managers and container virtualization.

Using Watchdog, we implemented default analysis workflows for typical types of large-scale biological experiments, such as RNA-seq and ChIP-seq. Although they can be easily applied to new datasets of the same type, at some point such standard workflows reach their limit and customized methods are required to resolve specific questions. Hence, the second part of this dissertation focuses on combining standard analysis workflows with the development of application-specific novel bioinformatics approaches to address questions of interest to our biological collaboration partners. The first study concentrates on identifying the binding motif of the *ZNF768* transcription factor, which consists of two anchor regions connected by a variable linker region. As standard motif finding methods detected only the anchors of the motifs separately, a custom method was developed for determining the spaced motif with the linker region. The second study focused on the effect of *CDK12* inhibition on transcription. Results obtained from standard RNA-seq analysis indicated substantial transcript shortening upon *CDK12* inhibition. We thus developed a new measure to quantify the degree of transcript shortening. In addition, a customized meta-gene analysis framework was developed to model RNA polymerase II progression using ChIP-seq data. This revealed that *CDK12* inhibition causes an RNA polymerase II processivity defect resulting in the detected transcript shortening.

In summary, the methods developed in this thesis represent both general contributions to large-scale sequencing data analysis and served to resolve specific questions regarding transcription factor binding and regulation of elongating RNA Polymerase II.

Chapter 1

Introduction

1.1 Biological background

Deoxyribonucleic acid (DNA) is the carrier of the genetic information of almost all known organisms [1]. It is composed of a sugar-phosphate backbone and the four nucleotides adenine, thymine, cytosine and guanine, and consists of two complementary intertwined strands [2, 3]. The genome's functional units are genes, which encode for proteins or regulatory ribonucleic acids (RNAs). Proteins consist of amino acid chains that fold into three-dimensional structures depending on their amino acid sequence. The resulting structure enables proteins to transport molecules, interact with other proteins or catalyze metabolic reactions [4].

Figure 1.1 illustrates a simplified version of the complex process leading from a gene to a protein. To initiate protein synthesis, general transcription factors bind to the so-called promoter located upstream of the corresponding gene. These factors recruit a protein complex consisting of RNA polymerase II (RNAPII) and about 50 other proteins [5]. This machinery transcribes the gene from the transcription start site (TSS) to the transcription termination site (TTS) resulting in a single-stranded copy of the gene, the so-called messenger RNA (mRNA) [6]. Subsequently, non-coding regions, the so-called introns, are spliced out from the resulting mRNA, the 5′ end is capped by adding a modified nucleotide and a poly(A) tail is attached at the 3′ end [7, 8]. Once exported to the cytoplasm, the mature mRNA is translated into an amino acid chain by the ribosomes [9, 10]. Each amino acid is encoded by a triplet of nucleotides (codon). The amino acid sequence is encoded by a sequence of codons between the start and stop codon [11, 12]. The regions of the mRNA before the start and after the stop codon are referred to as 5′ untranslated region (UTR) and 3′ UTR, respectively and often contain regulatory elements [13].

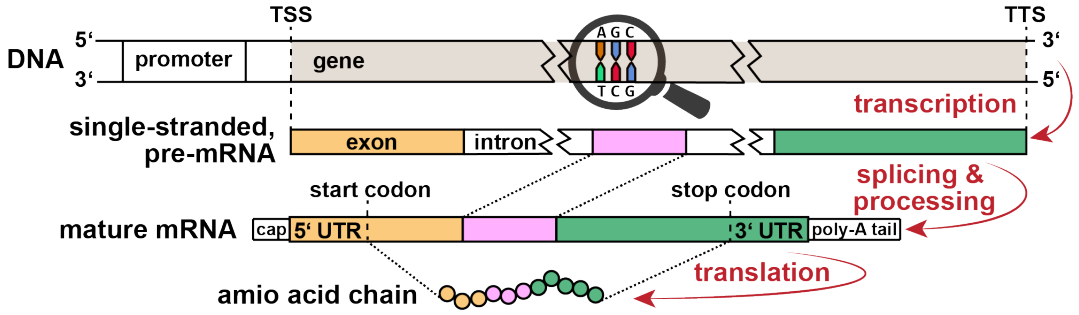


Figure 1.1: From gene to protein. A gene is transcribed from DNA to mRNA, non-coding regions are spliced out, modifications are applied and the mRNA is translated into a protein.

Transcription of most genes is heavily regulated on several levels, e.g. by cell- and gene-specific transcription factors that induce or repress transcription [14, 15]. Binding sites of specific transcription factors can be located near promoters or far removed from the gene (enhancer or silencer). Moreover, the DNA is wrapped around histones resulting in DNA-protein complexes, which makes it inaccessible to DNA-binding proteins. Hence, these structures have to be remodeled during transcription to make the DNA accessible [16]. Apart from transcription initiation, pausing of RNAPII progression and transcriptional bursts are important regulatory control mechanisms of transcription [17, 18]. Additionally, microRNAs (miRNAs) and other post-transcriptional regulation mechanisms can prevent mRNA translation or specifically target mRNAs for degredation [19].

1.2 Advances in sequencing technology and its applications

Sanger sequencing

In 1977, Frederick Sanger *et al.* laid the foundation for modern molecular biology by describing a method to determine the nucleotide sequence of DNA for the first time [20]. In the first step of the method, the single-stranded input DNA sample is divided into four test tubes, which each contain the four nucleotides. Subsequently, DNA polymerase is added to synthesize double-stranded DNA from the single-stranded DNA template. During double-strand synthesis, the DNA polymerase sequentially adds a new nucleotide at the 3′-OH group of the previously incorporated nucleotide. Each test tube additionally contains one type of nucleotide in low proportions that lacks the 3′-OH group. Once such a modified nucleotide is incorporated, the nucleotide chain is terminated. Following this step, each test tube contains incomplete copies of the template DNA, which all end with the chain-terminating nucleotide. To generate many copies ending at every possible position, the synthesis step is repeated multiple times. Here, heating the sample makes the DNA single-stranded again and allows another round of synthesis. Once the last cycle is finished, a gel with four separate lanes is used to separate the single-stranded DNA fragments by their length using electrophoresis. To make the resulting DNA bands visible, autoradiography is used as the chain-terminating nucleotides are radioactively labeled. Finally, the sequence of the input DNA can be read off the gel based on the relative positions of the bands in the four lanes.

To demonstrate the feasibility of the method, Sanger and colleagues sequenced the genome of bacteriophage ΦX174, which has a length of about five thousand nucleotides [20]. Only three years later, Frederick Sanger received the Nobel Prize in chemistry for this work [21]. Continuous improvements of his method led to state-of-the-art Sanger sequencers with very high accuracy [22]. Improvements include fluorescent labeling, capillary electrophoresis in one reaction, laser detection of fluorescence and automated sequencing of a few thousand fragments per day [23]. The availability of automated sequencers enabled the Human Genome Project to release the first human genome sequence in 2003. It took 13 years to complete and cost about $2.7 billion [24].

Next-generation sequencing

A breakthrough in sequencing technology was the development of methods for massively parallel sequencing, so-called next-generation sequencing (NGS) technologies. Today, massively dropped costs enable companies to offer genome sequencing for private customers within weeks for less than $1,000 [25]. Moreover, a vast number of sequencing library preparation protocols have emerged that capture diverse aspects of cells on a large scale [26]. With these protocols, it has become possible to quantify the expression of genes (RNA-seq), identify protein binding sites on DNA or RNA (ChIP-seq, CLIP-seq), study epigenetic marks (MeDIP-seq), detect open chromatin regions (ATAC-seq) or measure genome-wide chromatin interactions (CHIA-Pet), to name just a few examples. These methods have been applied intensively in the ENCODE project [27], which aimed to create

a catalog of all functional elements in the human genome. Moreover, an overwhelming amount of data has also been generated by other projects that apply NGS methods in large scale, e.g. the Cancer Genome Atlas [28], the 1000 Genomes Project [29], the Exome Sequencing Project [30], DiscovEHR [31] and the United Kingdom's 100,000 Genomes Project [32]. These projects aim to detect genetic variations in human genomes, explain genetic diseases or drug ineffectiveness and establish the basis for medical treatment adapted to specific characteristics of each patient [33].

Currently, Illumina is the market leader in NGS technology [34]. Illumina's sequencing technology is similar to the capillary electrophoresis approach of modern Sanger sequencers. However, the synthesis reactions occur in millions of nanowells located on a surface, the so-called flow cell. First, the DNA fragments in the sequencing library are immobilized through complementary adapter nucleotides bound to the flow cell. Afterwards, a few rounds of bridged amplification are performed to generate neighboring clusters of identical copies of the initially bound DNA fragments. Each cluster contains about 1,000 identical, single-stranded DNA fragments anchored at the flow cell. DNA polymerase and fluorescently labeled nucleotides are then added to the flow cell to generate double-stranded DNA. Here, the 3′-OH group of the amino acids is blocked that is required for the next nucleotide binding. Hence, only one complementary nucleotide can be initially incorporated. Subsequently, a laser excites the fluorescent labeling of the newly incorporated nucleotide and a microscope measures the spectrum of the emitted light. Next, the fluorescent labeling is cleaved and the 3′-OH group of the previously incorporated nucleotide is regenerated. These steps are repeated until a predefined number of nucleotides has been determined, usually between 50 and 300 base pair (bp) [35]. This procedure returns one short read per cluster, which matches the sequence of one end of the original DNA fragment. An optional bridged amplification step allows to sequence the other end of the original DNA fragment, leading to so-called paired-end reads. Extensive reviews on Illumina's and other NGS technologies are provided in [36, 37].

Illumina's most advanced instrument, the NovaSeq 6000, can sequence up to 20 billion paired-end, 150 bp-long reads within two days of operating time [38]. With that number of reads, about 48 full human genomes can be sequenced with running costs of $12-18 per gigabase pair [34]. Hence, sequencing of an incredibly large number of genomes has become possible, which could improve prevention, diagnosis and treatment of many diseases [39]. In 2020, Genomics England announced plans to sequence up to 500,000 whole genomes over the next five years to support the national health service (NHS) in introducing whole genome sequencing into routine healthcare [40].

Transcriptome profiling with RNA-seq

The term transcriptome refers to the collection of all RNA molecules present in a cell. In contrast to DNA, RNA is single-stranded and contains the nucleotide uracil instead of thymine. The transcriptome consists of numerous types of RNA. In mammalian cells, the most prevalent type is ribosomal RNA (rRNA), which accounts for about 80% of the RNA [41]. It is required to assemble the ribosomes, which serve as protein building factories. Another 15% are transfer RNAs (tRNAs) that transport amino acids required for protein synthesis to the ribosomes. The remaining 5% are mRNAs, miRNAs, small nuclear RNAs (snRNAs), long non-coding RNAs (lncRNAs) or other low abundant RNAs [41]. The transcriptome can be sequenced using RNA-sequencing (RNA-seq) in order to detect differentially expressed genes, quantify the expression of miRNAs or other small RNAs, identify alternative splicing events and catalog disease-related somatic mutations [42]. Here, RNA is not directly sequenced, but converted to complementary DNA (cDNA) libraries to allow use of DNA sequencing technology.

As the first step of sequencing library preparation, the total RNA is isolated from the cells. Next, the rRNA is removed as rRNA is the most abundant type, but of little interest [43, 44]. There are two ways to achieve this. One option, poly(A) enrichment uses oligo(T) primers bound to a surface to capture the poly(A) tails of mature mRNAs. Another method, rRNA depletion,

uses rRNA antisense transcripts bound to magnetic or biotinylated beads to capture and remove the rRNA from a sample. Afterwards, the remaining types of RNA are separated based on their length using gel electrophoresis. Subsequently, the RNA is converted to cDNA, fragmented, size selected for a specific fragment length and amplified by polymerase chain reaction [43].

Measuring protein-DNA interactions with ChIP-seq

Many proteins are known that bind DNA in a sequence-specific, shape-specific or even unspecific manner. The functions of DNA-binding proteins include activation or repression of expression (transcription factors), cleaving of DNA (nucleases), copying of DNA (polymerases) and DNA packing (histones) [45]. Some functions require the protein to move along the DNA from the original binding site. One example is RNAPII, which moves from the TSS of the gene to its TTS to transcribe the mRNA.

In order to determine regions in the genome at which a specific protein binds, ChIP-sequencing (ChIP-seq) can be used [46]. Here, DNA-bound proteins are first cross-linked to the DNA using formaldehyde and the DNA is sheared into smaller parts by sonication. Then, protein-specific antibodies bound to beads separate the target proteins together with the bound DNA fragments from the cell debris. The sensitivity and specificity of the antibody are crucial for the enrichment of DNA bound to the target protein. Afterwards, the proteins are unlinked from the DNA and the remaining DNA is purified. Finally, the resulting DNA fragments are prepared for sequencing. Multiple variants of this method exist that allow better resolution (e.g. ChIP-exo, ChIP-nexus) [47] or capture interactions of RNA with proteins (e.g. CLIP-Seq, PAR-CLIP) [48].

1.3 Standard bioinformatics analysis of NGS data

The next sections describe bioinformatics analysis steps commonly applied to all NGS datasets (1.3.1) as well as steps exclusively required for analysis of RNA-seq (1.3.2) or ChIP-seq (1.3.3) datasets.

1.3.1 General steps

Quality assessment and filtering

Quality assessment of the raw sequencing data is highly recommended before further analyses are performed as sequencing library preparation is a complicated process with many potential error sources [49]. Interesting properties include base quality scores, adapter contamination, over-represented sequences, GC content of the reads and read length distribution. An example for a quality assessment software is FastQC [50], which generates multiple quality control statistics for raw sequencing data. Depending on the degree of adapter contamination or uncertainty of base calls, reads are often trimmed or entirely discarded. Popular programs for that task are cutadapt [51] and Trimmomatic [52].

Mapping to reference genome

As a next step, the genomic origin of the reads has to be determined. This is done by aligning reads to a reference genome, which is also referred to as mapping. Most read mapping programs allow mismatches, insertions or deletions in alignments and clipping of both read ends. The alignment process has to be implemented very efficiently as alignments between every single read and a complete genome have to be calculated. Bowtie [53] and BWA [54] are frequently used programs for mapping reads that were obtained from DNA. Both programs utilize a FM-index to efficiently perform a vast number of string searches against a fixed genome sequence [55].

Mapping of reads obtained from mRNA is more complicated than mapping DNA sequencing reads due to splicing of the introns after transcription. Thus, two exons separated by an intron on

the genome can be directly connected in a read. Popular software for mapping RNA-seq reads is TopHat2 [56], HISAT2 [57] and STAR [58]. TopHat2 uses Bowtie as short read aligner and splits reads into smaller segments to map them across splice junctions. HISAT2 implements a hierarchical FM-index consisting of a gobal index and thousands of smaller indices that fit in the high-speed cache of modern computer processors. The global index serves for identifying all potential mapping regions within the complete genome, while the small ones are used to determine the exact read alignment. STAR uses an uncompressed suffix array to sequentially identify prefixes of a read that exactly match a region in the genome (maximal mappable prefixes). The resulting hits are then combined in a second step to an alignment for the complete read.

Another challenge of mapping NGS reads is that they are relatively short and may align equally well to multiple regions. In case of RNA-seq, pseudogenes are especially problematic. These are copies of real genes that lost their function, but are still very similar to the original gene [59]. Most mapping programs report multiple alignments for ambiguously mapping reads. In contrast, the RNA-seq mapper ContextMap2 assigns each read to exactly one location by considering the context around ambiguously mapping reads [60]. For this purpose, ContextMap2 aligns first reads to the reference genome using a short read alignment program. Afterwards, all fully mapped reads are clustered to define contexts, which are used to calculate support scores for reads that mapped ambiguously or partially. Finally, the remaining reads are assigned to the best-supported location based on these support scores.

1.3.2 Additional analysis steps for RNA-seq data

Expression level estimation

Once the genomic origin of RNA-seq reads has been determined, the expression of genomic regions of interest can be quantified. For example, quantification of protein-coding mRNAs allows drawing conclusions about the abundance of proteins. To estimate the expression level of genomic regions, mapped reads that overlap a particular region are counted. Here, reads are typically assigned to a gene if they overlap with an exon of that gene. Issues occur if a read maps equally well to several genomic locations or overlaps with multiple annotated genes.

The two most commonly used read counting methods are htseq-count part of the HTSeq framework [61] and featureCounts [62]. A benefit of using htseq-count is that it can be easily integrated into analysis pipelines implemented in Python. However, featureCounts is about 20 times faster than htseq-count and consumes only one-fifth of the memory. To speed up the test for overlapping intervals, it uses a two-level hierarchical data structure combined with chromosome hashing.

Differential gene expression analysis

Differential gene expression analysis compares the expression levels of genes between different conditions. For instance, biological systems are often deliberately permuted, e.g. by treatment with a (potential) drug and then compared to the natural state. In theory, the gene read counts for each condition could be divided by each other to calculate gene expression fold-changes. In practice, however, this is not that simple and requires additional steps. One problem is that the sequencing depth of different samples varies and thus does not allow a direct comparison. Accordingly, most programs for differential gene expression analysis normalize read counts by library size. Another problem is that sequencing biases (e.g. GC bias [63] or gene length bias [64]) cause overrepresentation of some transcripts within the library. To compensate for that, sequencing libraries are normalized using various assumptions about read distributions. Finally, statistical tests are used to calculate the fold-changes between conditions per gene and the statistical significance of these fold-changes [65]. The power of statistical tests for detecting differentially expressed genes is affected by the library sequencing depth and the number of replicates per condition [66, 67]. Since one statistical test is

carried out for each gene, the statistical significance has to be corrected for multiple testing [68].

Many programs for the detection of differentially expressed genes have been developed in the last decade. They differ in the used normalization method and the underlying statistical model. Popular programs include are edgeR [69], limma [70], DESeq [71] and DESeq2 [72]. Systematic comparisons between different methods can be found in [73, 74, 75, 76, 65].

Alternative splicing analysis

So far, splicing has been described as a process in which all introns of a gene are spliced out from the pre-mRNA (see Fig. 1.1). However, for many genes it is far more complicated because different mRNA isoforms are produced from the same gene by alternative splicing. The expression of different isoforms is frequently condition- and tissue-specific [77]. Known alternative splicing events include exon skipping, intron retention, alternative $5'$ or $3'$ splice site usage and inclusion of mutually exclusive exons [78, 79]. Since reads obtained with NGS technologies are too short to sequence complete transcripts, bioinformatic methods are required to investigate alternative splicing.

Two general strategies exist to identify and quantify differentially spliced genes in RNA-seq data. Isoform-based methods try to reconstruct and quantify full-length transcripts and apply differential expression analysis afterwards. Programs implementing this approach are cuffdiff2 [80], DiffSplice [81] and IsoDE2 [82]. The second general approach is to analyze splicing events either based on exon read counts (e.g. DEXSeq [83], JunctionSeq [84] and limma [70]) or individual splicing events (e.g. rMATS [85], MAJIQ [86], SUPPA2 [87] and MISO [88]). Some programs depend on comprehensive gene annotations, while others can recognize novel splicing events. More detailed comparisons and evaluations of differential splicing analysis programs can be found in [89, 90, 91].

1.3.3 Additional analysis steps for ChIP-seq data

Peak calling

After ChIP-seq reads are mapped to the reference genome, regions covered by a significant number of reads have to be identified. This process is referred to as peak calling and reveals genomic positions at which the target protein was bound. As unspecific binding and different accessibility of the DNA introduces noise, commonly a sample without antibody is prepared, sequenced and used as background control sample [92]. The success of the experiment heavily depends on the specificity of the antibody, while the sensitivity mainly depends on the sequencing depth [93]. Depending on the target protein, sharp (e.g. transcription factors [94]) or broad peaks (e.g. histone modifications [95]) are observed in the experiment.

Again, many different programs exist for peak calling. They differ in the algorithm used to identify potential peaks, the applied normalization method, the statistical test to measure the significance and their ability to detect sharp or broad peaks. The most popular software is MACS and its improved version MACS2 [96]. The algorithm uses the fact that both strands of the enriched DNA are sequenced from their $5'$ ends. Hence, a bound protein causes one peak up- and downstream of its binding site due to the fragment length. Both peaks together result in a bimodal read distribution around the actual binding site of the target protein. Other commonly used methods are SISSRs [97], SICER [98], F-Seq [99], FindPeaks [100], PeakSeq [101] and GEM [102]. For a feature comparison of 30 peak callers and an evaluation of six of them, please refer to [103].

Binding motif identification

In many cases, the binding motif of the target protein is still unknown. Hence, the next task after peak calling often is to computationally identify the binding motif based on the genomic sequences at peaks. Several issues make motif identification difficult. Due to unspecific binding or other artifacts, not all of the identified peaks contain the motif. Moreover, the resulting peaks can be

up to a few hundred nucleotides long and hence only roughly indicate the binding position [104]. Additionally, binding motifs of other proteins or nearby repetitive elements that overlap with the peaks can obstruct the correct identification of the motif [105].

Generally, two different approaches exist for motif discovery in ChIP-seq data. The first one is to enumerate and count all possible motifs, which guarantees that the optimal solution is found. A shortcoming of this approach is the exhaustively big search space. The runtime grows exponentially and depends on the number of input sequences, the maximum number of allowed mismatches and the maximal supported motif length. Usually, the user has to define these parameters as they determine the required runtime of the algorithm. Other programs implement probabilistic approaches that are able to process big datasets and require less user input. However, probabilistic approaches are more complex and might only find a local maximum. Popular enumeration- and probabilistic-based programs are HOMER [106], MEME [107], DREME [108], STEME [109] and GLAM2 [110]. Classification of 119 motif discovery algorithms and a general feature comparison can be found in [111].

Peak annotation and hypothesis generation

Once the position of the peaks is known, the peaks usually are annotated with the features of a gene (e.g. promoter, exon, intron, 5′ UTR, 3′ UTR) they overlap with to infer their function. For instance, transcription factors that modulate the expression of genes, often bind in the promoter region upstream of the TSS (see Fig. 1.1). In this way, peak annotation can help to generate hypotheses about the function of the target protein, which can be verified experimentally. For example, the R package ChIPseeker [112] can annotate, compare and visualize peaks.

1.3.4 Challenges for NGS data analysis

As outlined in the previous sections, bioinformatics analyses of NGS datasets consist of several interdependent steps. In addition to the problem of choosing an appropriate algorithm for each of these steps, several technical aspects have to be addressed on top. These include the amount of manual interaction required during an analysis, sufficient flexibility to allow an analysis to be reused for similar datasets and reproducibility of the obtained results [113]. There are various approaches with inherent advantages and disadvantages to address these points.

The most straightforward way is to execute all required steps manually or with the help of scripts. However, it is difficult to efficiently execute and monitor many long-running steps manually, in particular since they often depend on each other. Furthermore, it is not uncommon that new samples are added to a project after the initial analysis has been completed (e.g. new replicates or conditions) or that the way the data is processed needs to be changed (e.g. further steps or use of alternative algorithms). In both cases, all dependent steps have to be identified and executed again.

Another problem is that it is laborious to properly document a manually executed analysis and actually reproduce it based on the documentation. In 2016, a survey among 1,500 scientists revealed that archiving reproducibility is a major problem in science. For instance, 70% of all respondents failed at least once to reproduce an experiment initially performed by other scientists. To improve the situation, the authors recommended better documentation and standardization, among other things [114]. However, reproducibility is also an issue in computational science and bioinformatics [115, 116, 117]. Points that complicate analyses and potentially hinder reproducibility are differences between software or database versions, altered software behavior on different operating systems or unnoticed software crashes. An alternative to manual execution is implementing pipelines designed to perform a specific analyses [118, 119]. The advantage of this approach is that an analysis can be repeated without much effort. In addition, it requires much less manual interaction and therefore is less prone to errors. However, depending on the implementation, components

of such analysis pipelines might not be easily reusable in other pipelines. Code duplication and subsequent adaption might allow reusing parts of the pipeline, but complicate code maintainability.

The best approach is to use a workflow management system (WMS) to execute the analysis. For this purpose, the analysis pipeline has to be defined as a workflow according to the specification of the corresponding WMS. A workflow consists of a collection of tasks that depend on each other. A WMS parses the workflow and schedules, executes and monitors tasks in an unsupervised manner. Usually, workflow parameters can be configured such that a workflow can be easily reused in other projects. Other benefits are that WMSs typically encapsulate tasks in reusable components, are able to distribute time- and resource-consuming jobs on a computer cluster, handle various types of storage or automatically deploy software. Thus, WMSs relieve the user of many tasks that he would otherwise have to take care of. The choice of a specific WMS depends on different criteria including the required training period, implemented set of features, target audience, availability of a graphical user interface (GUI) and license fees to be paid. Popular WMSs are Galaxy [120], KNIME [121], Snakemake [122] and Nextflow[123], which are briefly described in the following.

Galaxy is a scientific analysis platform for designing and executing workflows in the web browser. Users can upload their data to a Galaxy server, combine available analysis programs and configure the workflow parameters without programming experience. Public Galaxy servers provide accessibility to everyone and allow easy sharing of tools and workflows. Tools are defined in an XML format specifying the program to execute and its input parameters.

KNIME is a data analysis platform that provides a GUI for workflow construction, execution and result visualization. A workflow consists of so-called nodes that are arranged by drag-and-drop and then configured in the GUI. Nodes for data manipulation, modeling and visualization are distributed together with KNIME. New nodes have to be defined in Java by extending multiple Java classes to integrate existing software in a workflow.

Snakemake is a WMS inspired by GNU Make. A workflow is defined as a set of rules, which specify how input files are used to create output files. Rules can execute shell commands, Python code or external scripts. Programmers can define own Snakemake workflows in a language that extends Python. Snakemake supports execution on workstations, computer clusters and cloud environments and can deploy the required software automatically.

Nextflow is build around the Unix pipeline model that uses streams to transfer data between consecutive tasks. Nextflow extends the model to allow streaming of complex data structures instead of plain text. Workflows are defined as a succession of processes in a proprietary scripting language based on Groovy. Similar to Snakemake, programming experience is required to define own workflows as no GUI is available.

1.4 Thesis outline

Four peer-reviewed articles contribute to this cumulative dissertation. The first two articles focus on the development of the WMS Watchdog. The second two articles cover the analysis of two high-throughput NGS datasets. In both cases, application of standard analysis workflows was combined with the development of application-specific bioinformatics approaches. In the following, the content of each article is briefly summarized and my contributions are indicated.

Contributing articles for Section 2.1

M. Kluge and C. C. Friedel. Watchdog - a workflow management system for the distributed analysis of large-scale experimental data. *BMC Bioinformatics*, 19(1):97, March 2018

M. Kluge, M.-S. Friedl, A. L. Menzel, and C. C. Friedel. Watchdog 2.0: New developments for reusability, reproducibility, and workflow execution. *GigaScience*, 9(6):giaa068, June 2020

In Section 2.1, I present the WMS Watchdog. The initial version of Watchdog was published in 2018 and a much extended version in 2020 [124, 125]. Watchdog targets experimentalists with only basic high-throughput data analysis experience as well as experienced bioinformaticians. Its main features include straightforward processing of replicate data, support for distributed computer systems, customizable error detection and manual intervention into workflow execution. To enable non-programmers to design and execute workflows, I implemented a comprehensive GUI.

The updated version introduced a community sharing platform for modules and workflows, a searchable module reference book, two new execution modes for more comfort and flexibility during workflow execution and support for automated software deployment using container virtualization and package managers. Additionally, Amrei L. Menzel implemented a GUI to semi-automatically create new modules from help or man pages of command-line programs as part of her Bachelor's thesis. For both publications, I produced the figures and wrote the initial draft. Caroline C. Friedel and for the second article Marie-Sophie Friedl, helped with revising the manuscripts and tested the software.

Contributing articles for Section 2.2

M. Rohrmoser, M. Kluge, Y. Yahia, A. Gruber-Eber, M. A. Maqbool, I. Forné, S. Krebs, H. Blum, A. K. Greifenberg, M. Geyer, N. Descostes, A. Imhof, J.-C. Andrau, C. C. Friedel, and D. Eick. MIR sequences recruit zinc finger protein ZNF768 to expressed genes. *Nucleic acids research*, 47(2):700–715, January 2019

In Section 2.2, I describe the methodology for the analysis of high-throughput datasets generated to characterize the function of the zinc finger protein *ZNF768* [126]. Our collaboration partners Dirk Eick and Michaela Rohrmoser performed RNA-seq of wild-type cells and *ZNF768*-knockout mutants to quantify the effect of *ZNF768* on the transcriptome. In addition, they performed ChIP-seq to determine the binding positions of the protein. *ZNF768* attracted their interest as it contains a domain highly similar to a domain of RNAPII that is involved in initiation of transcription, splicing and regulation of RNAPII activity.

I analyzed the RNA-seq and ChIP-seq datasets and developed an application-specific analysis method under the supervision of Caroline C. Friedel with some valuable input from Dirk Eick and Michaela Rohrmoser. A key result I obtained was the identification of the binding motif with a new method specially developed for this purpose. The motif consists of two anchor regions connected by a variable linker of fixed length and was experimentally validated by Michaela Rohrmoser. My RNA-seq data analysis indicated that *ZNF768* might act as a regulator of many other transcription factors. For the manuscript, I created figures visualizing the core findings of my analyses, helped draft the bioinformatics methods and analysis section together with Caroline C. Friedel and helped to revise the complete manuscript.

Contributing articles for Section 2.3

A. P. Chirackal Manavalan, K. Pilarova, M. Kluge, K. Bartholomeeusen, M. Rajecky, J. Oppelt, P. Khirsariya, K. Paruch, L. Krejci, C. C. Friedel, and D. Blazek. CDK12 controls G1/S progression by regulating RNAPII processivity at core DNA replication genes. *EMBO reports*, 20(9):e47592, September 2019

In Section 2.3, I present methods developed for the integrated analysis of RNA-seq and ChIP-seq data measured with and without *CDK12* inhibition [127]. *CDK12* encodes for a protein kinase known to regulate transcription of DNA damage and stress response genes. Our collaboration partners Dalibor Blazek and Anil P. Chirackal Manavalan used ChIP-seq to measure occupancy of RNAPII and two different phosphorylation states of the C-terminal domain of RNAPII, which are associated with initiation and elongation of transcription.

I performed the analysis of RNA-seq and ChIP-seq data under the supervision of Caroline C. Friedel and with constructive suggestions from Dalibor Blazek and Anil P. Chirackal Manavalan. More specifically, I developed and implemented a method to quantify transcript shortening. This revealed that many transcripts of predominantly long genes are shortened upon *CDK12* inhibition. Furthermore, I implemented a customized meta-gene analysis framework to model RNAPII progression using ChIP-seq data. By combining both methods, I was able to show that transcript shortening is accompanied by a shift of RNAPII occupancy. For the manuscript, I prepared the figures with regard to the high-throughput data analysis, helped draft the bioinformatics methods and analysis section and helped in revising the complete manuscript.

Chapter 2

Summary of contributing articles

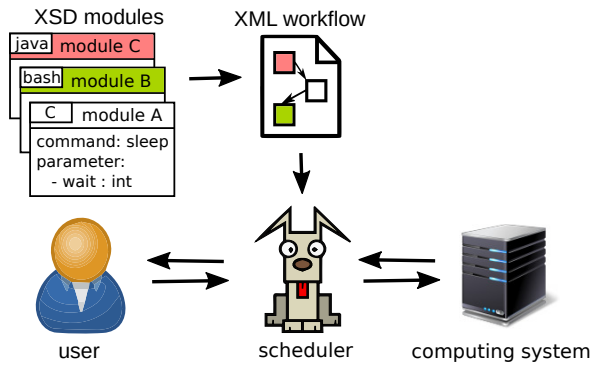
2.1 Processing of large-scale experimental data with Watchdog

We developed the workflow management system (WMS) Watchdog to support scientists in the analysis of high-throughput datasets [124, 125]. Similar to other WMS, a Watchdog workflow consists of a set of tasks and dependencies between these tasks that define the task execution order. A workflow is defined in an XML format, which supports the configuration of constants and environment variables, replicate data processing, distributed task execution and automatic software deployment. An example for a simple workflow is given in Figure 2.1a. Individual tasks executed in Watchdog workflows are encapsulated within so-called modules. A module is defined by an XSD schema file, which specifies the command to execute and its input and output parameters. Thus, there is no restriction regarding the programming languages used to implement the functionality of a module (see upper part of Fig. 2.1b). Apart from the XSD schema file, a module can optionally contain scripts, compiled binaries and test data.

Workflows are executed by the Watchdog scheduler, which is implemented in Java and thus platform-independent. The Watchdog scheduler determines the task execution order, schedules tasks for execution, continuously monitors their execution status and informs the user on success and error (see lower part of Fig. 2.1b).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <watchdog watchdogBase="/install/dir/" ...>
3     <settings>
4        <constants>
5           <const name="WAIT_TIME">30s</const>
6        </constants>
7     </settings>
8     <tasks mail="my@mail.net">
9        <sleepTask id="1" name="sleep">
10          <parameter>
11             <wait>${WAIT_TIME}</wait>
12          </parameter>
13       </sleepTask>
14    </tasks>
15 </watchdog>
```

(a) Simple workflow in XML format



(b) Workflow and scheduler

Figure 2.1: Watchdog workflows. (a) The example shows a simple Watchdog workflow, which executes a sleep task. A constant defines the sleep duration of 30 seconds. (b) Modules encapsulate reusable components that perform individual tasks in workflows and specify the command to execute and available parameters. The Watchdog scheduler executes these tasks, monitors their execution status, informs the user on errors and allows him to intervene into workflow execution.

In the following, a more detailed overview of Watchdog's features is provided and these features are briefly compared against the WMSs Galaxy, KNIME, Snakemake and Nextflow. For more details, please refer to the original publications [124, 125].

2.1.1 Defining and sharing of modules and workflows

Programs for module and workflow construction

Watchdog workflows can be created and edited in any XML editor. Alternatively, Watchdog provides a GUI for workflow construction, the so-called workflow designer. It enables users without programming experience to construct workflows using pre-defined modules. The application window of the workflow designer is divided into three main parts (see Fig. 2.2). On the left, the module library lists available modules and allows to filter them based on a text search. The right side is reserved for defining and configuring environment variables, distributed task execution, replicate processing, constants and automatic software deployment. The central area visualizes all tasks in the workflow and their status (e.g. correctly configured or execution status). New tasks can be added to a workflow by dragging the corresponding module from the module library, moving it to an empty position and dropping it there. Dependencies between tasks are also created using drag-and-drop. Task parameters, in- or output streams and more can be configured in a pop-up window by clicking on the task itself.

Before a workflow can be assembled, the required modules have to be created. The user can either define the XSD module file manually using any XML editor or use the provided helper script. It allows creating the module XSD file in an interactive manner and optionally generates a skeleton Bash script that the developer can extend. Another possibility is to use the so-called moduleMaker GUI, which was developed by Amrei L. Menzel as part of her Bachelor's thesis. The moduleMaker extracts parameters and flags of command-line programs from their help or man page to semi-automatically create module XSD files. For this purpose, different sets of regular expressions matching common help page formats are used. Once the user selected the best matching set of regular expressions, he can make adjustments concerning the detected parameters and flags on the GUI and finally save the resulting XSD file.

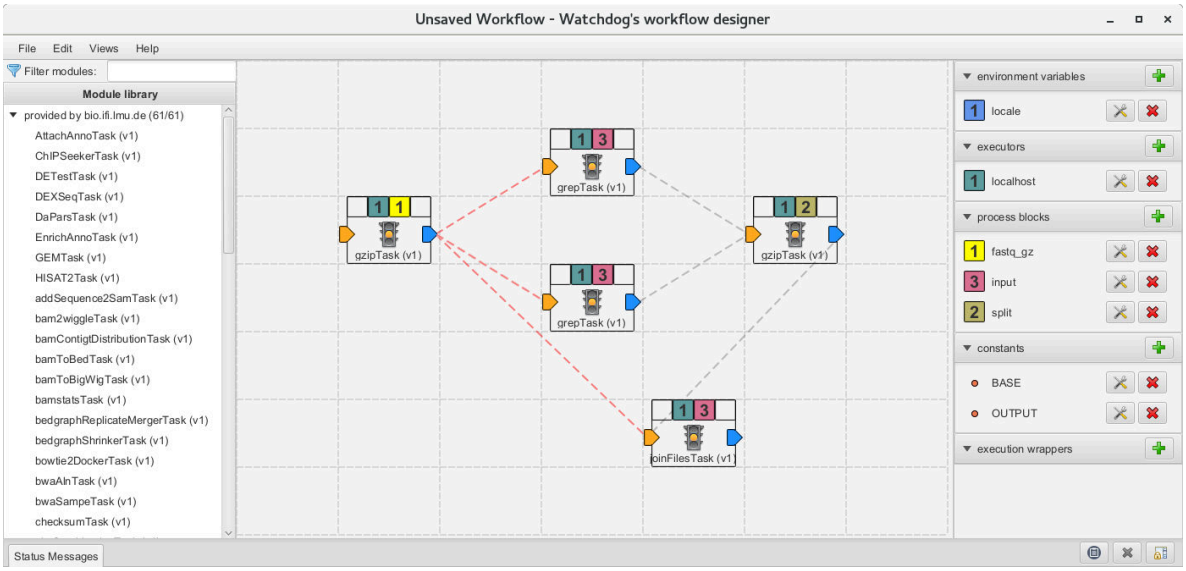


Figure 2.2: GUI for workflow construction. Screenshot of the workflow designer GUI during the creation of a sample workflow. For a description please refer to the main text.

Comparison: Galaxy offers a web application to create and execute workflows, while KNIME provides a GUI based on the IDE Eclipse. In both cases, the workflows are created via drag-and-drop and configured in separate windows. For Snakemake and Nextflow, no comparable solutions are available. To support the user during module creation, Galaxy also offers a command-line program similar to the helper script provided by Watchdog. KNIME provides a wizard that generates the required backbone classes in Java, which have to be massively extended by the user.

Public repositories for module and workflow sharing

To support and encourage the exchange of modules and workflows, there are two public sharing repositories available on Github under the *watchdog-wms* organization. Users can contribute to these repositories by making changes to a copy of the repository (fork) and then requesting that the changes are integrated into the original repository (pull request). An integration pipeline ensures that certain requirements are fulfilled before the proposed changes can be accepted.

During the development of Watchdog, two analysis workflows were established to process and analyze RNA-seq and ChIP-seq datasets. Both workflows were designed to be highly reusable by supporting any number of replicates and by using constants for parameters that vary between datasets. The workflows can be executed without having to manually install external software dependencies by using Watchdog's automatic software deployment support. These workflows and all required modules are publicly available at `https://github.com/watchdog-wms/watchdog-wms-workflows` and `https://github.com/watchdog-wms/watchdog-wms-modules`, respectively.

Comparison: Galaxy and KNIME operate dedicated platforms to share components of workflows or complete workflows with others. Snakemake provides reusable wrappers in source code repositories that everyone can contribute to. For KNIME and Nextflow, sharing platforms are operated by the respective communities.

Module reference book

Both for sharing and use of modules, it is very valuable to have a reference book containing information about all available modules and how to use them. Watchdog provides a program for generating a so-called module reference book from standardized module documentation files (XML format). The reference book is implemented as a nicely formatted and searchable HTML web page. Its start page provides an overview of all modules and allows to filter modules based on text search or categories. In addition, a detailed view for each module contains a brief functional description as well as information on third-party software dependencies, input and output parameters, citation information and web resources.

Comparison: Snakemake, KNIME and Galaxy also enable documentation of workflow components and their parameters in XML or YAML format. In case of Snakemake, a reference book for reusable wrappers can be generated containing software dependencies, an example Snakefile, author information and the source code of the wrapper. KNIME and Galaxy visualize the documentation of components on their GUI or respective web interface during workflow creation. Both offer similar documentation options as Watchdog.

2.1.2 Workflow execution and control by the user

Different execution modes

In addition to running a complete workflow, Watchdog offers several execution modes. First, it provides the option to only process a consecutive range of tasks or specifically selected tasks. This is e.g. useful if workflow execution was interrupted unexpectedly or a workflow was modified. However, this procedure is error-prone as the user has to manually decide which tasks to rerun. Alternatively, the resume mode can be used to automatically resume processing of interrupted workflows or to

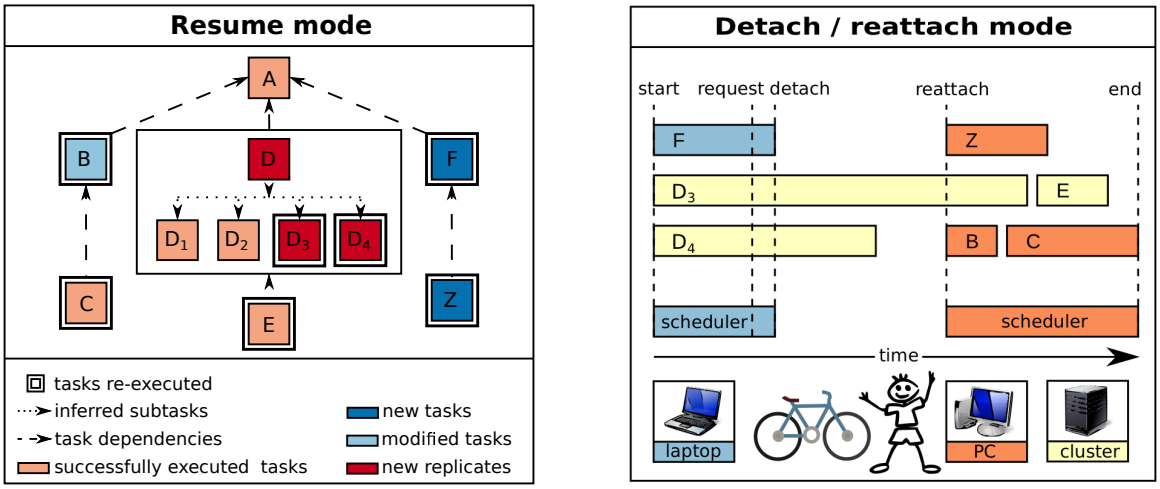
process workflows with altered parameters, additional tasks or more samples. In all cases, only tasks that require (re-)execution are scheduled (see Fig. 2.3a). The third execution mode allows to detach the Watchdog scheduler from workflow execution while tasks distributed to a computer cluster continue running. The scheduler can be reattached to the workflow execution at a later time from any computer with access to the shared file system (see Fig. 2.3b). Use cases for the detach mode include rebooting the computer running the Watchdog scheduler or changing location with a laptop. A detach request can be sent by the user at any time using a keystroke combination.

Comparison: Apart from Galaxy, all WMSs are able to resume execution of partly executed workflows and to execute only tasks of a modified workflow that require (re-)execution. In case of Snakemake and Nextflow, new samples can be added to a workflow without having to reprocess all samples. Similar execution modes to Watchdog's detach mode are at least partly implemented in all WMSs apart from Nextflow.

Manual intervention into workflow execution

In addition to different execution modes for the user, Watchdog provides multiple ways to intervene into workflow execution. First, the Watchdog scheduler and the workflow designer GUI allow the user to keep track of task processing using the standard output or the visual representation of the workflow. Second, Watchdog provides a web-interface that displays the execution status of all tasks in a table-based form. The web-interface additionally allows stopping or restarting a task or modifying its parameters. If errors occur during workflow execution, the user is optionally notified per email and has two additional options to proceed. He can decide to manually resolve the problem and mark it as solved. Alternatively, he can ignore all tasks that depend on the failed task during further processing.

Comparison: Galaxy and KNIME also allow the user to intervene into workflow execution in their respective GUIs. Snakemake and Nextflow provide no intervention possibilities during workflow execution, but allow to restart the modified workflow without reprocessing successfully executed tasks.



(a) Resume mode

(b) Detach / reattach mode

Figure 2.3: Execution modes. (a) The resume mode allows workflow execution after interruption or modification without rerunning successfully executed tasks not affected by changes. (b) The detach and reattach mode allows detaching the Watchdog scheduler and reattaching to workflow execution later on while tasks continue running on computer clusters.

Automated reporting

When analyzing data, it is important that each step of an analysis is described precisely such that others can reproduce the analysis. To support analysis documentation, Watchdog creates a time-stamped log file for each workflow execution containing input parameters and return values of each successfully executed task. Based on the log file, Watchdog is able to generate a step-by-step report of the analysis that could serve as basis for the methods section of an article. The report contains for each task the module description, citation information and optionally task parameters or used software versions in the order the corresponding tasks were executed.

Comparison: KNIME supports static workflow descriptions, but does not generate reports on executed workflows. In contrast, Snakemake, Nextflow and Galaxy create reports on executed analyses in table or list format. None of these WMSs generate a step-by-step report as basis for a manuscript draft. Galaxy allows exporting citation information of used programs after workflow execution.

2.1.3 Features supporting efficient and reproducible workflow execution

All of the following features can be extended by programmers using either Watchdog's plugin system or its dynamic class loading feature.

Processing of replicate data

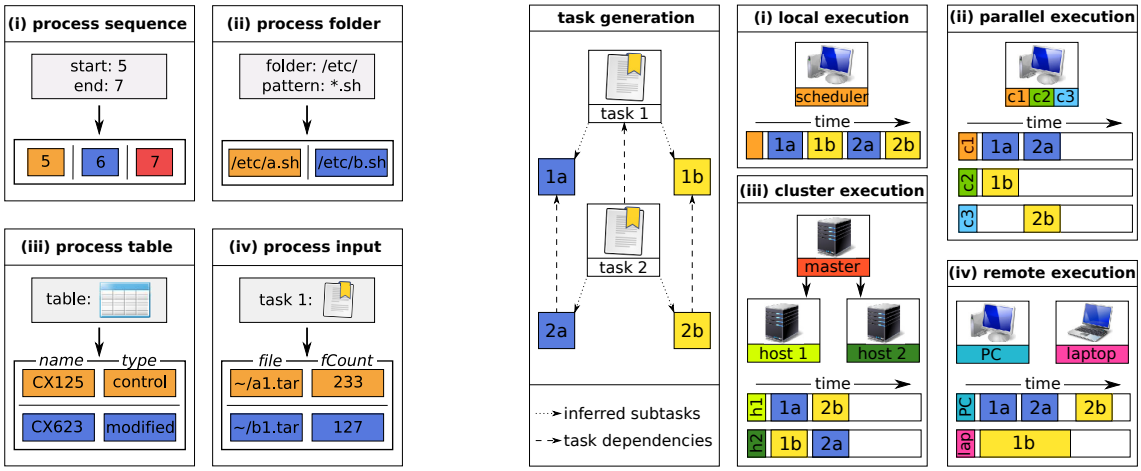
In Watchdog, so-called process blocks can automatically produce many instances of a task that differ only in the values of parameters. When a process block is applied on a task, subtasks are created by the Watchdog scheduler for each instance. During this process, special placeholders part of the task definition (e.g. input parameter) are replaced with varying values provided by the process block. This feature is extremely valuable when the same task has to be applied to many different samples or has to be executed multiple times with modified parameters. So far, four types of process blocks are implemented for processing (i) numerical sequences, (ii) a set of files defined by a filename pattern, (iii) tables or (iv) output parameters obtained from module dependencies (see Fig. 2.4a). Developers can implement custom process blocks by using Watchdog's plugin system.

Comparison: All four WMSs provide ways to process replicate data. However, for Galaxy, a collection has to be created manually and for KNIME special nodes have to be used that control the structure of the workflow.

Distributed task execution

By default, all tasks of a workflow are executed one after the other locally on the host running the Watchdog scheduler. However, additional computing capacity is beneficial for parallel processing of resource-intensive or long-running tasks such as mapping of RNA-seq data. Thus, Watchdog supports task execution on the (i/ii) local host, (iii) computer clusters using the Sun Grid Engine, the Slurm workload manager or the generic Distributed Resource Management Application API (DRMAA) and (iv) remote host via SSH (see Fig. 2.4b). Moreover, Watchdog's plugin system enables users with programming skills to integrate new types of executors. Different executors can be used in a workflow to meet resource requirements of the respective tasks and at the same time minimize the occupied computing power.

Comparison: Galaxy, Nextflow and Snakemake support various cluster engines and cloud solutions. Furthermore, Snakemake allows usage of any cluster engine offering scriptable job submission if a shared file system is provided. In case of KNIME, commercial extensions are available that support execution on the Sun Grid Engine or servers dedicated to KNIME.



(a) Supported types of process blocks

(b) Parallel and distributed task execution

Figure 2.4: Task generation and execution. (a) With the help of process blocks, multiple subtasks (colored rectangles) that differ only in the parameter values can be created without defining all of them separately. (b) Four subtasks 1a, 1b, 2a and 2b are created by the scheduler based on tasks 1 and 2 using process blocks. In this example, these tasks are then executed in (i) serial or (ii) parallel mode on the local host that runs the Watchdog scheduler, on a (iii) computer cluster or on (iv) remote hosts via SSH.

Automatic software deployment

Having to install all software required for modules used in a workflow can be time-consuming and cumbersome. Furthermore, to exactly reproduce the result of a previously executed workflow run, users have to ensure that they have installed the same software versions as previously. To address these problems, so-called execution wrappers support automatic deployment of software via package managers or container virtualization. These wrappers can be assigned to any executor and different types of wrappers can be used in a workflow. It is also possible to use a package manager inside a container. Currently, Docker, Podman and Singularity are supported for container virtualization and Conda as package manager, but additional execution wrappers can be implemented using Watchdog's plugin system.

Comparison: Apart from KNIME, all three other WMSs support controlling external software dependencies with Conda or Docker.

Customizable error detection

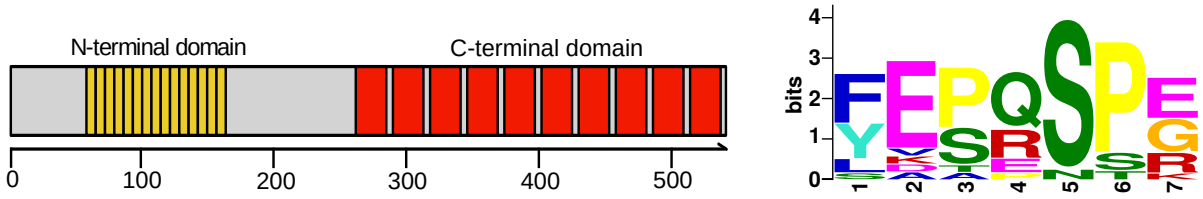
A number of errors can occur during the execution of a workflow. These include hardware failures, software errors or insufficient computer resources that cause a program to crash. Unexpected interruption of programs can lead to corrupted files, which are either unreadable (e.g. binary formats) or truncated. An analysis workflow might complete without further errors in the latter case, but the results are incorrect. Thus, Watchdog implements a two-stage error checking system. First, Watchdog checks if the exit code of the executed module indicates that the command succeeded. However, checking the exit code alone is not sufficient since not every software implements it correctly. Furthermore, a command can succeed from a technical point of view without resulting in the intended result (e.g. the wrong index used for mapping of RNA-seq data results in a low mapping rate). Thus, as second step, custom success and error checkers can be automatically applied on finished tasks. Developers can implement a simple Java interface to define their own success or error checkers. During workflow execution these Java classes are instanced by Watchdog's dynamic class loading feature.

Comparison: Apart from KNIME, all three other WMS automatically check the exit code of executed commands. In case of KNIME, node developers have to implement the exit code check by themselves. Customizable error checks are not available in any of these WMSs.

2.2 Uncovering the role of *ZNF768* in gene regulation

The *ZNF768* gene belongs to the family of zinc finger proteins, which interact specifically with DNA or RNA [128, 129]. They evolved during mammalian evolution and represent one of the largest gene families in the human genome with more than 700 members [130]. The family members encode for diverse functions including DNA binding site recognition, RNA packaging, apoptosis regulation, protein folding and activation of transcription [131]. As a consequence, zinc finger proteins are involved in the development and progression of several severe diseases such as cancer, neurodegeneration and diabetes [132, 133, 134, 135, 136].

Figure 2.5a shows a visualization of the domain structure of the *ZNF768* protein. Its C-terminal domain (CTD) consists of ten zinc fingers (red) as annotated in the Uniprot database [137]. A repeating pattern of seven amino acids, so-called heptad-repeats (yellow), is located at the N-terminus of the protein. Figure 2.5b shows the information content for each amino acid of these heptad-repeats per position. Interestingly, the array of heptad-repeats has a high similarity to the CTD of RNAPII, which is involved in initiation of transcription and splicing, and is essential for the regulation of RNAPII activity [138, 139]. Due to this similarity, our collaboration partners Michaela Rohrmoser and Dirk Eick hypothesized that *ZNF768* might act as transcription factor with gene regulatory function. Knockdown experiments showed that the *ZNF768* protein is at least required for viability and proliferation of cells.



(a) Annotated domains of *ZNF768* **(b)** Sequence logo of heptad-repeats

Figure 2.5: The *ZNF768* gene. (a) The *ZNF768* protein contains an array of heptad-repeats at the N-terminus (yellow) similar to the CTD of RNAPII and ten zinc fingers at the C-terminus (red). The tick marks on the x-axis indicate the amino acid position within the protein. (b) The sequence logo was calculated for a gapless alignment of the heptad-repeats located at the N-terminus of the *ZNF768* protein.

2.2.1 Experimental setup and initial data processing

Experimental setup

To characterize the influence of *ZNF768* on gene expression and its binding sites, our collaboration partners Michaela Rohrmoser and Dirk Eick carried out RNA-seq and ChIP-seq experiments. The experiments were performed in the two cell lines Raji [140] and U2OS [141], which were initially derived from human cancer cells in the 1960s [142].

For RNA-seq, four biological replicates were prepared from total RNA for Raji and U2OS wild-type cells. Moreover, a mutated form of *ZNF768* (ZNF768-ΔN) was constructed in U2OS using a plasmid vector [143]. The mutant expresses an alternative form of *ZNF768* containing only the

C-terminal zinc finger domain. Expression of the mutated transcript is inducible by the addition of doxycycline. After 12 hours of ZNF768-ΔN doxycycline-induced overexpression, four biological libraries were prepared. In case of the ChIP-seq experiment, two library replicates each were prepared for the Raji and U2OS wild-type cell line with an antibody against *ZNF768*. Additionally, one library without immunoprecipitation was prepared for each cell line, which serves as background to detect unspecifically isolated DNA. All libraries were sequenced on Illumina HiSeq instruments.

For the analysis of both high-throughput datasets, I implemented Watchdog workflows containing the standard analysis steps described in Section 1.3. Both workflows are now publicly available at `https://github.com/watchdog-wms/watchdog-wms-workflows`. The workflows and crucial parameters are described in the following paragraphs.

RNA-seq data processing with Watchdog

First, the sequencing read files were decompressed and the sequencing quality of reads was assessed using FastQC [50]. Next, the reads were aligned to the human reference genome hg38 and human rRNA sequences with ContextMap2 [60]. Subsequently, the resulting alignment files were converted into a compressed and indexed format and various statistics about the mapped reads were calculated using samtools [144] and RSeQC [145]. Then, fragment counts per gene were estimated by featureCounts [62] with Gencode 25 [146] as annotation. Next, the statistics for all samples were merged and visualized with custom R scripts. Finally, differential gene expression analysis was performed with limma [70].

ChIP-seq data processing with Watchdog

After decompressing the sequencing read files, the read quality was assessed using FastQC [50]. Subsequently, the reads were aligned to hg38 using BWA [54] and unpaired read alignments and alignments with a mapping quality less than 20 were discarded. Once the resulting alignment was compressed and indexed, peaks were identified using GEM [102] and peaks with a q-value ≤ 0.01 were included in further analysis. The resulting peaks were then annotated using ChIPseeker [112] and some figures were generated that provide an overview of the data.

2.2.2 Identification of the *ZNF768* DNA binding motif

Peak identification and replicate integration

GEM does not provide a sophisticated way to handle biological replicates as it just combines the replicates before the analysis. Hence, the information is lost if a peak was measured repeatedly in biological replicates. To avoid that, all replicates were processed individually and the resulting peaks were then compared between replicates and cell lines.

To define peak regions, the peak center determined by GEM was extended by 100 bp in both directions. Afterwards, overlapping peak regions were merged across all samples. This approach identified 21,012 non-overlapping regions covered by at least one ChIP-seq peak in one sample (referred to as unique peak regions in the following). The number of detected peaks per sample was very heterogeneous. More than 15,000 peaks were identified in replicate 1 for the Raji cells. About 4,500 peaks were detected in replicate 2 for the U2OS cells and about 9,000 peaks for each of the other two samples. Nearly 2,800 peaks were identified in all four samples and more than 6,000 in both replicates for at least one cell line.

Identification and validation of the bipartite *ZNF768* binding motif

We first applied HOMER [106] and MEME-ChIP [147] for motif discovery, but both were not successful in identifying a motif contained in most of the peaks. HOMER consumed too many resources while searching for a motif longer than 26 bp and crashed. MEME-ChIP reported two 8

bp-long binding motifs that occur in about 45% and 35% of all unique peak regions, respectively. A manual search for the corresponding consensus sequences CCTCTCTG and GCTGTGTG in the peak sequences revealed that they often occurred together, separated by 20 bp ± a few bp. This observation led to the hypothesis that *ZNF768* binds both of these regions (denoted as anchors), which are connected by a less conserved linker sequence with a length of around 20 bp.

To verify that hypothesis, I implemented a program that specifically searches for the two anchor sequences separated by a linker of 20 bp. Since the partial motif hits found by MEME-ChIP implied that certain positions are variable and the inspection of the data showed that varying linker length occur, the program allowed up to M mismatches in the anchor motifs as well as up to D bp deviation from the mean linker length. M and D are parameters of the program. Many motif occurrences will remain unidentified in the peak sequences if parameters for M and D are too stringent. However, numerous false positive motif hits will be found if too many errors are allowed. Consequently, to determine reasonable values for M and D, a parameter screening was performed.

For this purpose, 200 bp-long DNA sequences were randomly selected from the human genome using bedtools [148] and R [149] to obtain realistic nucleotide distributions. The sequences were shuffled with MEME's sequence shuffler [150] to remove all occurrences of the binding motif. Two test datasets, each containing 250,000 sequences, were created by maintaining nucleotide (1-mer) or dinucleotide frequencies (2-mer) during shuffling. Afterwards, the motif search program was applied on the test datasets with various M and D values. The results of the parameter screening are visualized in Figure 2.6. A linker length of 20 with a deviation of ±3 bp and up to 3 mismatches caused less than 1% randomly detected motif hits in both datasets. Hence, these parameters were used in all following analyses unless otherwise stated. Larger values for D are of little value from a biological perspective as the zinc fingers require a specific distance for binding. Similarly, M >3 is not useful as the fraction of randomly detected motif hits increases exponentially.

Application of the motif search program with $D = 3$ and $M = 3$ on all unique peak regions showed that >80% of the peaks contained the motif in each replicate. The only exception was replicate 1 of the Raji cells as only 56% of the identified peaks contained the motif, suggesting that many of these represent either weaker binding or false positives. Notably, 98% of the 2,747 peaks identified in all samples contained a motif hit. Figure 2.7 shows the resulting sequence logo [151] for all 12,205 identified motif hits.

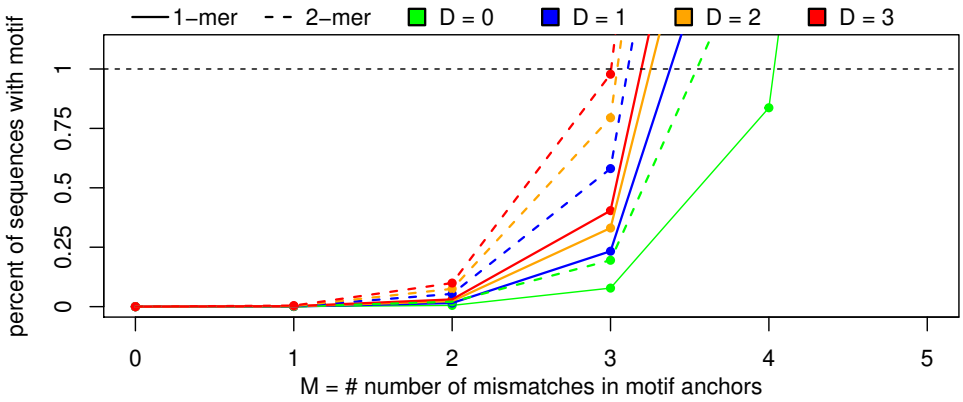


Figure 2.6: Parameter screening. The motif search program was invoked with different parameters for the number of allowed mismatches in the anchors (M) and the maximal deviation in bp from the optional linker length (D). M is plotted on the x-axis, while the percent of sequences with an identified motif hit is plotted on the y-axis. Colors indicate the deviation D, solid lines belong to the 1-mer dataset and dashed to the 2-mer dataset.

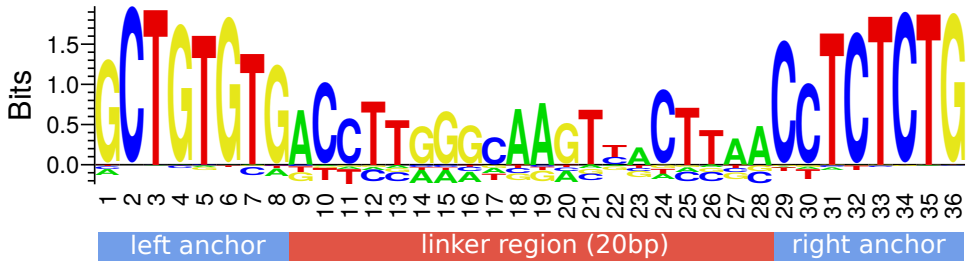


Figure 2.7: Logo of *ZNF768* binding motif. Sequence logo of the 12,205 motif hits found with $D = 3$ and $M = 3$ in all unique peak regions created with WebLogo [153]. The center of the linker region was ignored for logo frequency calculation if the linker length of a motif hit was not 20 bp long.

Michaela Rohrmoser confirmed that *ZNF768* indeed binds to GCTGTGTG(N_{20})CCTCTCTG using an electrophoretic mobility shift assay for detecting protein-DNA interactions [152]. The experiment also demonstrated that the linker sequence is not required for the binding specificity of *ZNF768*. The effect of different linker lengths was not evaluated.

2.2.3 Origin and conservation of the *ZNF768* binding site in mammalian genomes

Mammalian-wide interspersed repeats

A comparison of the unique peaks to repeats annotated by RepeatMasker [154] revealed that more than half of the peak regions overlapped with mammalian-wide interspersed repeat (MIR) elements. Such repetitive sequences were included in genomes through the process of reverse transcription millions of years ago. MIRs are tRNA-derived retrotransposons that are only present in mammals and approximately 260 bp long [155, 156]. Four different subfamilies have been described: MIR, MIR3, MIRb and MIRc. These will be referred to as MIR* in the following [157]. MIR* regions cover about 2.6% of the human genome with more than 550,000 individual copies [158, 154, 159].

Further analysis showed that the *ZNF768* binding motif is contained in a 70 bp-long conserved core region that is common to all MIR* members [155]. A motif search applied to all MIR* occurrences revealed that the motif is detectable in 12% of these occurrences with $D = 3$ and $M = 3$. With more strict search parameters ($D \in [0,1]$ and $M = 1$), only 15,378 motif hits remain, 54% of which were recovered by the *ZNF768* ChIP-seq experiment.

Binding site conservation in mammalian genomes

To investigate if the *ZNF768* binding sites within MIR* regions are conserved in mammalian genomes, the RepeatMasker software was used to align the consensus sequences of the four MIR subfamilies against the human genome. Next, genomic coordinates were extracted for all MIR* regions that were covered by gapless alignments in the binding motif region ±25 bp. Afterwards, the distributions of phyloP100 conservation scores per position were evaluated to estimate the conservation of these genomic regions. PhyloP100 scores are calculated from multiple alignments of 99 vertebrate genomes against the human genome and were obtained from the UCSC database [160]. Finally, conservation of gaplessly aligned MIR* regions was compared between regions bound by *ZNF768* and regions not bound by it.

The analysis showed that both anchors of the binding motif are conserved in bound MIR* regions. In contrast, the linker sequence and the regions up- and downstream of the binding motif are mostly not conserved. For unbound MIR* regions, no specific conservation of the anchors was observed. This finding indicates that *ZNF768* binding resulted in the conservation of particular positions of a subset of mammalian MIR* sequences during evolution.

2.2.4 Effect of *ZNF768* on gene expression

Genomic peak annotation

To test if the unique peaks are enriched in regulatory elements of genes, they were first assigned to promoters, UTRs, exons or introns of genes using ChIPseeker [112]. Subsequently, random peaks were sampled and used to calculate an enrichment between the real peaks and the random background for each annotation type. The analysis shows that peaks are enriched more than 2-fold in promoters compared to the random background. Interestingly, peaks showing the motif but without MIR* overlap are stronger enriched in the promoter and also slightly in introns and exons. Either these *ZNF768* binding sites evolved independently of the evolution of MIR* or the former MIR* regions were changed too strongly to be detectable by RepeatMasker.

Association of *ZNF768* binding with gene expression

Differential gene expression analysis of the ZNF768-ΔN RNA-seq dataset revealed that 500 genes were down- and 155 genes were up-regulated at least 2-fold compared to wild-type U2OS. A functional enrichment analysis for UniProt [161] keywords on the down-regulated genes showed that more than 20% of these were DNA-binding or zinc finger proteins. This observation suggests that *ZNF768* functions as a regulator upstream of a network of transcription factors.

Analysis of the wild-type RNA-seq dataset showed that genes with *ZNF768* peaks in promoters or 5′ UTRs were more highly expressed than genes without peaks. The effect was more pronounced in the Raji than in the U2OS cell line. Another analysis revealed that genes with *ZNF768* peaks that were detected in both replicates of a cell line, but not in any other sample for the other cell line were more highly expressed in the corresponding cell line. These findings together indicate that *ZNF768* binding is associated with actively transcribed genes and partially occurs in a cell-type-specific manner.

2.3 Integrative RNA-seq and ChIP-seq analysis of *CDK12* function

Fundamental research in the 1980s revealed the existence of kinases important for cell cycle regulation in yeast [162]. Kinases are enzymes that catalyze the transfer of a phosphate group from one molecules to another. After it was confirmed that these kinases are also relevant for mammalian cell cycle control and depend on cyclin, the family of cyclin-dependent kinases (CDKs) was defined in 1991 [163]. The human genome encodes 21 CDK proteins and five more distant, so-called CDK-like proteins [164]. More recent studies indicated that some CDK family members are also involved in the regulation of transcription and mRNA splicing [165].

In 2001, Ko *et al.* reported the discovery of the *CrkRS* protein and characterized it as CDK-related kinase [166]. Five years later, it was confirmed that *CrkRS* activity depends on cyclin. Consequently, *CrkRS* was renamed to *CDK12* and included in the CDK family [167]. On a molecular level, *CDK12* acts by transferring a phosphate group from adenosine triphosphate (ATP) to serine or threonine. More recent investigations on *CDK12* showed that it regulates transcription of DNA damage and stress response genes [168, 169], is frequently mutated in cancer [170, 171, 172, 173, 174] and might be a valuable biomarker or therapeutic target [175, 176, 177].

2.3.1 Experimental setup and initial data processing

Experimental setup

To further characterize the function of *CDK12*, our collaborators Dalibor Blazek *et al.* constructed an analog-sensitive version of *CDK12* using the CRISPR-Cas technology [178]. The change of a single nucleotide on both *CDK12* alleles in the HCT116 cell line enables the ATP analog 3-MB-PP1

to occupy the binding site of ATP. As a consequence, treatment with 3-MB-PP1 allows inhibiting the activity of *CDK12* in a specific and rapid manner [179]. This approach is an improvement over previous studies on *CDK12* that used long-term depletion of *CDK12*, which is prone to induce compensatory effects [180]. After cell synchronization, *CDK12* was inhibited for five hours using 3-MB-PP1 as inhibitor. Untreated cells were used as control condition. Afterwards, different high-throughput datasets were prepared (three replicates per condition) and sequenced on the Illumina platform as described in the following using 50 bp-long, single-end reads.

In total, two strand-specific RNA-seq datasets were prepared using different library preparation protocols. For the first dataset, oligo(T) primers were used to enrich RNAs with poly(A) tails from the total RNA fraction. In this case, only the 3′ ends of the remaining transcripts were used to prepare the sequencing library. For the second dataset, newly synthesized transcripts were enriched by extracting the nuclear RNA fraction from cells and subsequent rRNA depletion.

ChIP-seq experiments were carried out with antibodies against either RNAPII or its phosphorylated forms P-Ser5 and P-Ser2. RNAPII can be phosphorylated at its CTD at many positions. Regulation of RNAPII by phosphorylation is known to be a complex process [138, 139]. However, it is broadly accepted that phosphorylating RNAPII at P-Ser5 is required for transcription initiation, whereas phosphorylation of P-Ser2 is associated with elongation of transcription [181, 182]. After the first analysis results were available, ChIP-seq experiments were also performed with an antibody against the transcription elongation factor *SPT6*.

Automated data processing with Watchdog

All datasets were processed with Watchdog workflows similar to the ones used for the analysis of the *ZNF768* data (see Section 2.2.1). Hence, only deviating steps are briefly outlined here.

RNA-seq reads were mapped to the hg38 human reference genome and human rRNA sequences using ContextMap2 [60]. Then, read counts per gene and exon were strand-specifically determined by featureCounts [62] using Gencode 27 [146] as annotation. Differential gene expression analysis was performed using DESeq2 [72] and differential exon usage with DEXSeq [83]. The ChIP-seq reads were aligned to hg38 using BWA [54]. Next, reads with an alignment score smaller than 20 were discarded and the read coverage per genome position was determined using bedtools [148].

Based on the initial data processing, more specific analyses described in the following were applied to both datasets. The same workflows and analyses were also applied to publicly available datasets from the NCBI sequence read archive (SRA) [183] to check if similar effects are detectable in other *CDK12* inhibition datasets.

2.3.2 Identification and quantification of transcript shortening

Analysis of differentially expressed genes

Differential gene expression analysis of the poly(A)-selected RNA-seq dataset revealed that 1,491 and 611 genes were at least 2-fold down- or up-regulated, respectively. Overrepresentation analysis of gene sets defined by the Gene Ontology (GO) [184] on down-regulated genes performed with the GOrilla webserver [185] and gene set enrichment analysis [186] on $\log_2$ fold-changes of all genes showed that DNA repair-, DNA replication- and cell cycle-related genes were significantly enriched among the down-regulated genes.

Evidence for transcript shortening from differential exon usage analysis

Some members of the CDK family, and possibly also *CDK12* itself, are involved in the regulation of splicing [165, 167, 187]. However, the 3′ end protocol used in the RNA-seq experiment did not allow to differentiate between down-regulation of genes and differential splicing of the last exon. Therefore, our collaboration partners carried out RNA-seq using entire transcripts obtained from

the nuclear RNA fraction. Comparison of differentially expressed genes in both RNA-seq datasets showed a substantial overlap for at least 2-fold regulated genes, especially for down-regulated ones. Moreover, the $\log_2$ fold-changes of differentially expressed genes were highly correlated.

Differential exon usage analysis of the nuclear RNA-seq data using DEXSeq [83] revealed that 7,341 exons were used differentially upon *CDK12* inhibition (adjusted p-value cutoff 0.01). Analysis of the relative positions of differentially used exons within genes showed that down-regulated exons were more frequently in proximity to 3′ gene ends. Up-regulated exons, on the other hand, tended to be close to 5′ gene ends. Manual inspection of mapped reads of genes with down-regulated exons at their 3′ ends revealed that the read coverage was lost before reaching the annotated transcript end. This observation led to the hypothesis that the observed gene down-regulation in poly(A)-selected and nuclear RNA was actually due to premature transcription termination and resulting transcript shortening rather than down-regulation of entire genes.

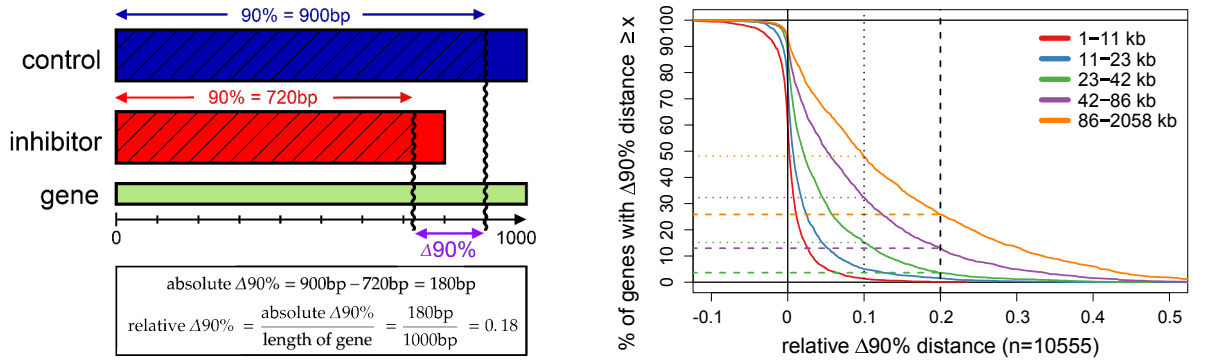
Quantification of transcript shortening

To quantify the extent of transcript shortening for a gene, we had to develop a new method. A very straightforward approach would be to calculate the distance from the TSS to the position where read coverage is lost in the RNA-seq data for each gene and compare these distances between control and *CDK12* inhibition. Unfortunately, this approach would massively underestimate the degree of the shortening as a single mapped read to the 3′ end of a gene for *CDK12* inhibition would prevent the detection of transcript shortening.

Instead, we developed the so-called X% distance to quantify transcript shortening. It measures the distance from the TSS to the position downstream of the gene at which X percent of the gene's total read coverage is observed when summing up the read coverage starting at the TSS (see Fig. 2.8a). The implementation of the X% distance calculation uses a binary search-based approach. First, the complete gene region is divided into two parts and read coverage from the TSS to the split position is calculated. If this value is larger than X% of the coverage of the complete gene, the search continues recursively in the first half of the region, otherwise it continues in the second half. If the X% distance is to be calculated for different values of X for the same sample, several recursion paths are followed simultaneously.

The absolute ΔX% distance is defined as the difference of the X% distance in control minus the X% distance after *CDK12* inhibition. A positive ΔX% distance value indicates that read coverage is shifted towards the 5′ end of a gene and therefore indicates shortening of the transcript. A negative value indicates transcript lengthening. The relative ΔX% distance is calculated by dividing the absolute ΔX% distance by gene length and can be used to compare genes with different lengths. Figure 2.8a shows an example for calculating the 90% distance for a transcript with a length of 1,000 bp. For simplicity, the example assumes a uniform read coverage across the transcript. For the control sample, the 90% distance is 900 bp and after inhibition it is 720 bp. As a consequence, the absolute Δ90% distance is 180 bp and the relative Δ90% distance is 0.18.

Using this program we calculated the 90% distances for the nuclear RNA-seq dataset. For each gene, we selected the transcript with maximal RNAPII coverage in the ChIP-seq experiment. Figure 2.8b shows the cumulative distributions of the relative Δ90% distance for five equally sized gene groups. The genes were grouped based on their length. This analysis revealed that longer genes were predominantly affected by premature shortening, whereas short genes remained mostly unaffected. For instance, nearly 50% of transcripts with length $\geq$ 86 kilobase pair (kb) were shortened by more than one-tenth of their length.



(a) Exemplary calculation of the 90% distance

(b) Distribution of the relative Δ90% distance

Figure 2.8: X% distance. (a) The example illustrates how the 90% distance is calculated and how it can be used to compare two samples with each other. The absolute Δ90% distance measures the difference between the 90% distances of two samples in bp. To obtain the relative Δ90% distance, the absolute Δ90% distance is divided by the length of the region of interest. (b) Relative Δ90% distances were calculated for the nuclear RNA-seq dataset for each replicate. The median function was used to aggregate the relative Δ90% values of replicates. Genes were then grouped into equally large sets based on their length. Cumulative distributions are shown for each group.

2.3.3 RNAPII processivity defect

Meta-gene analysis framework

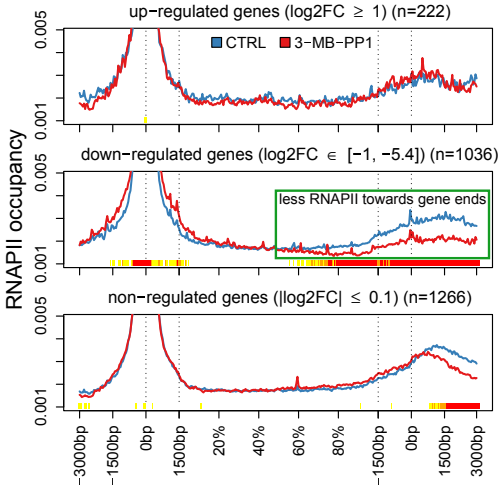
To analyze the ChIP-seq read coverage of many genes simultaneously, meta-gene plots are often used. Meta-gene plots visualize the aggregated (e.g. averaged) read coverage of a group of genes in a particular region. Thus, meta-gene plots allow to visualize a read coverage pattern common to many genes and can be used to compare this pattern between different conditions. Existing programs that create meta-gene plots often visualize the region around the TSS $\pm X$ bp. This approach is suitable for transcription factors, which usually bind within or near the promoter [15]. However, the ChIP-seq experiments performed for this study captured RNAPII, which travels from the TSS to a few thousand bp beyond the TTS during transcription. To visualize average RNAPII coverage for many genes in the same figure, I developed a framework that creates meta-gene plots for any regions of interest (e.g. complete genes) and applied it to model the progression of RNAPII.

As the length of the regions in a group can vary massively, all regions have to be scaled to the same length before creating a meta-gene plot. For this purpose, each region is divided into a predefined number of bins. To investigate the regions around the TSS and TTS without scaling, the framework provides the option to add fixed-sized bins at the start and end of each region. The remaining central parts of each region are divided into equally sized bins. To account for differences in sequencing depth and expression, different normalization methods have been implemented. The first approach normalizes only by sequencing depth. Since highly expressed genes would massively bias the average of all genes, a second normalization method was implemented, which normalizes all bins of a region by dividing by the total sum of all bins. Thus, the sum of all bins after normalization is one and all genes contribute equally. Finally, meta-gene plots are created by aggregating normalized values for each bin across all considered genes. The binning algorithm is implemented as a Java program and available as a Watchdog module. An R library handles normalization and creation of the meta-gene plots.

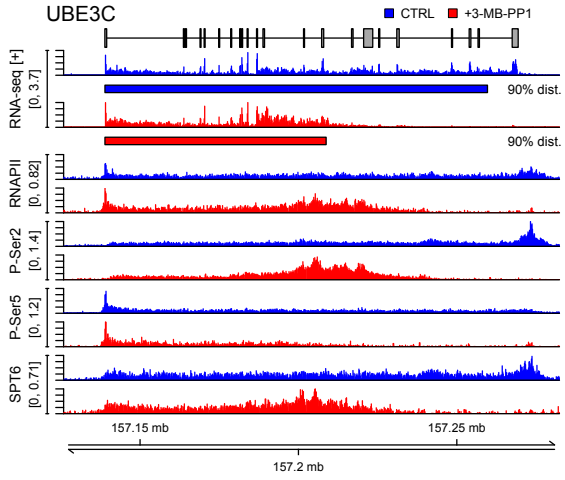
RNAPII occupancy shifts identified by meta-gene analysis

To apply the meta-gene framework on the ChIP-seq dataset, the transcript with the maximal RNAPII coverage was again selected for each gene. The analysis was restricted to high confidence transcripts of protein-coding genes annotated in Gencode 27. The region binning was performed as follows: The 4,500 bp regions around the TSS and TTS were each divided into 90 equally-sized bins. The remaining gene body was split into 180 bins of variable length, resulting in a total of 320 bins per gene. To compare RNAPII processivity between genes affected to different degrees on the RNA level, multiple gene sets were defined based on either differentially expressed genes in the nuclear RNA-seq dataset or the relative 90% distance. For each gene set, RNAPII occupancy was compared in meta-gene plots between inhibition and control condition for averaged replicate data. To determine the statistical significance of differences, paired Wilcoxon signed-rank tests were performed for each bin comparing the normalized coverage values for all genes with and without the inhibitor. *P*-values were adjusted for multiple testing using the Bonferroni correction [188].

Figure 2.9a shows the RNAPII occupancy for genes that are either up-, down- or non-regulated in the nuclear RNA-seq data between control and inhibitor. The meta-gene plot clearly shows a loss of RNAPII occupancy at 3′ ends for down-regulated genes with the inhibitor. The Wilcoxon tests confirm that this difference is highly significant in this region. This result is consistent with down-regulation being mostly caused by premature transcript shortening. The same analysis applied on the P-Ser2 dataset reveals a shift of P-Ser2 occupancy from the 3′ end towards the gene body (not shown here). Both observations indicate that the transcript shortening is caused by an RNAPII processivity defect. Analysis of gene sets with different lengths confirmed that predominantly long genes are affected by the shift of RNAPII occupancy.



(a) Meta-gene plot of RNAPII occupancy



(b) Transcript shortening for *UBE3C*

Figure 2.9: RNAPII processivity defect. (a) Meta-gene plots for RNAPII occupancy for three different gene sets, i.e. up-, down- and non-regulated genes. The meta-gene plots were created with the program described in the text. The colored bars at the bottom represent multiple-testing adjusted *p*-values for each bin (from low to high significance: yellow, orange, red). (b) Average read coverage (normalized to library size) for the nuclear RNA-seq dataset and the ChIP-seq datasets are visualized with Gviz [189] for the inhibitor (red) and control (blue) condition. The 90% distance is depicted as a bar below the RNA-seq coverage tracks. *UBE3C* exhibits premature transcript shortening. The new 3′ end is roughly located at the midpoint of the gene. This is accompanied by a shift of RNAPII, P-Ser2 and SPT6 into the gene body.

Visualization of read coverage

To visualize the transcription defect for individual genes, Gviz [189] was used to plot RNA-seq and ChIP-seq read coverage and 90% distances. For this purpose, read coverage was normalized to the total number of mapped reads per sample and then averaged between replicates. Because control over font size, positioning of labels and the general layout is limited in Gviz, I implemented a new program for generating read coverage plots based on Gviz. The program allows to define the general layout of tracks, the appearance of coverage and annotation tracks, title and sample labels and the appearance of the y-axis via configuration files. Additionally, I created a Watchdog module to allow generation of read coverage plots for regions of interest in workflows.

Figure 2.9b shows the read coverage plot for the *UBE3C* gene, which is an example for a gene strongly affected by the transcript shortening defect. Visual inspection of several affected genes showed that the location of the 3′ end P-Ser2 peak roughly corresponds to the point at which coverage is lost in the nuclear RNA-seq data after inhibition of *CDK12*.

Chapter 3

Discussion and outlook

3.1 Watchdog put to the test

The WMS Watchdog was developed to support the automated and distributed analysis of large-scale experimental data. So far, I developed and used Watchdog analysis workflows in several collaborations. This has led to the following publications in addition to the ones presented in this thesis.

Collaboration projects using Watchdog for large-scale sequencing data analysis

[190] T.-M. Decker, M. Kluge, S. Krebs, N. Shah, H. Blum, C. C. Friedel, and D. Eick. Transcriptome analysis of dominant-negative Brd4 mutants identifies Brd4-specific target genes of small molecule inhibitor JQ1. *Scientific reports*, 7(1):1684, May 2017

[191] R. Tejero, Y. Huang, I. Katsyv, M. Kluge, J.-Y. Lin, J. Tome-Garcia, N. Daviaud, Y. Wang, B. Zhang, N. M. Tsankova, C. C. Friedel, H. Zou, and R. H. Friedel. Gene signatures of quiescent glioblastoma cells reveal mesenchymal shift and interactions with niche microenvironment. *EBioMedicine*, 42:252–269, April 2019

[192] P. Metzger, S. V. Kirchleitner, M. Kluge, L. M. Koenig, C. Hörth, C. A. Rambuscheck, D. Böhmer, J. Ahlfeld, S. Kobold, C. C. Friedel, S. Endres, M. Schnurr, and P. Duewell. Immunostimulatory RNA leads to functional reprogramming of myeloid-derived suppressor cells in pancreatic cancer. *Journal for immunotherapy of cancer*, 7(1):288, November 2019

[193] X. Zhou, S. Wahane, M.-S. Friedl, M. Kluge, C. C. Friedel, K. Avrampou, V. Zachariou, L. Guo, B. Zhang, X. He, R. H. Friedel, and H. Zou. Microglia and macrophages promote corralling, wound compaction and recovery after spinal cord injury via Plexin-B2. *Nature neuroscience*, 23(3):337–350, March 2020

Admittedly, many other WMSs are now available to perform such analyses. In the following, I will briefly discuss five aspects that make Watchdog stand out from these WMSs. A detailed comparison of Watchdog to the WMSs Galaxy, KNIME, Snakemake and Nextflow can be found in the original publications [124, 125].

First, Watchdog provides a customizable error detection system to check whether the execution of a task completed successfully or not. Many, if not all WMSs, use only the exit code of an executed command to determine whether the command succeeded or failed. However, this procedure is not

sufficient as software often does not comply with the exit code convention. Moreover, a command can technically succeed despite some problems with the results (e.g. too low mapping rate of sequencing data). Such problems need to be identified before continuing with the analysis. Hence, Watchdog's customizable error detection system is a beneficial extension to ensure that all steps of the analysis really were properly executed.

Second, not every computational infrastructure and package manager is supported by a particular WMS. With its plugin system, Watchdog allows extending essential core features easily, in particular support for different types of executors, virtualizers and package managers, and process blocks. To create a new plugin, the developer has to define a new XML element for the feature and implement the classes required for parsing and processing the new element. The element can be tested and used without modifying the original Watchdog source code as Watchdog's plugin system dynamically loads plugins during workflow parsing.

Third, in addition to automatic deployment of software, Watchdog's automated reporting feature helps to maintain reproducibility. In particular, it enables other scientists to repeat an analysis as the generated step-by-step report ensures that no essential analysis steps are unintentionally omitted in a manuscript's methods section. Apart from KNIME, all other WMSs can export a list containing all executed tasks, but none of these can serve as easily as a draft of a manuscript's methods section. One additional feature of Watchdog that exceeds the capabilities of most WMSs is that used software versions, parameter values and citation information can be automatically included in the report.

Fourth, during the development of Watchdog particular emphasis was put on supporting straightforward processing of similar tasks. This feature is especially useful during NGS data analysis, as often the same steps have to be applied to a large set of samples. For this purpose, process blocks were introduced that automatically create multiple instances of a task template, so-called subtasks. The implemented types of process blocks obtain the parameters for these subtasks from filename patterns, tables, numeric ranges or output parameters of other tasks. If these comprehensive options are not sufficient, developers can implement new types of process blocks using Watchdog's plugin system. While Snakemake and Nextflow also support automatic processing of replicate data, it requires significant manual work in Galaxy and special nodes that control the structure of the workflow in KNIME.

The last important property is that the training time to work with Watchdog is relatively short as workflows can be designed with the GUI. Similar to Galaxy and KNIME, users can easily create new workflows without having to learn the used XML semantic. Moreover, by using the helper script program or the moduleMaker GUI, even non-programmers can create modules for already existing software. Compared to KNIME, where several Java classes have to be extended for integrating a new software, the overhead is small. However, Watchdog also offers programmers the option of creating workflows and modules in an XML editor, which may be more convenient for them. Thus, Watchdog enables users without programming experience as well as experienced developers to integrate already existing software, define workflows and execute them afterwards.

In summary, Watchdog now provides similar functionalities as existing WMSs with its recent second release. Furthermore, it offers valuable features not present in other WMSs and enables a wide range of users to perform large-scale bioinformatics analyses in a flexible and reproducible manner. For the future, I plan to focus on the development of new modules and workflows for sequencing data analysis. With the ongoing advances in NGS methods and the spread of new technologies such as third-generation [194, 195] and single-cell sequencing [196, 197], the need for reproducible and reusable analysis workflows will ever increase in the next years.

3.2 Open questions regarding *ZNF768*

As part of this project, I identified the binding motif of the zinc finger protein *ZNF768* based on ChIP-seq data generated by our collaboration partners. Interestingly, many of the observed binding sites are located within MIR sequences, which originate from transposable elements. In the early days of genomic research, MIRs and other repetitive sequences were described as "junk DNA". More recently, there have been reports that some of these elements fulfill crucial functions and also impacted the development of mammals [27, 198]. MIR sequences, for instance, have been reported to act as genome insulators [199], provide enhancer function [200] and correlate with tissue-specific gene expression [201]. In case of *ZNF768*, the spread of MIR sequences might have provided potential DNA-binding sites for the protein. However, only a small fraction was preserved over millions of years, potentially due to their role as *ZNF768* binding sites.

Some open questions remain that were beyond the scope of our original study. One concerns how similar *ZNF768* orthologs are to each other and how frequent MIR sequences and *ZNF768* binding sites occur in other species. Among 44 mammalian orthologs of *ZNF768*, the number of zinc fingers in the CTD is conserved for 37 of them, suggesting that these orthologs can bind the *ZNF768* motif. The number of heptad repeats in the N-terminus, however, ranges from 0 to 21. Hence, not all of them might have the same function. For instance, the platypus ortholog contains a SCAN domain at its N-terminus instead of the heptad repeats indicating that the protein mediates aggregation of homo- or heterodimers [202, 203].

In absence of suitable ChIP-seq data for other species, I applied my motif finding program and the RepeatMasker [154] software to genome assemblies of these 44 mammals obtained from NCBI [204]. Interestingly, the platypus genome contains ten times more *ZNF768* binding sites than the human genome and five times more MIR sequences. In contrast, the genomes of mouse and rat contain four times less MIR sequences than the human genome, while containing a comparable number of *ZNF768* binding sites. Further genomic comparisons could provide insight into the conservation of *ZNF768* binding sites and the evolution of *ZNF768*.

Another fascinating aspect is that the binding motif consists of two conserved anchor regions separated by a 20 bp-long linker. Most reported DNA-binding motifs are considerably shorter and do not contain poorly conserved linker regions. For instance, 99% of the motifs in the JASPAR database [205] are shorter than 22 bp. The longest reported binding motif has a length of 30 bp. Inspection of validated motives in JASPER's matrix clustering view detected only very few spaced motifs with linker regions. None of these motifs has a linker longer than 8 bp. Either long (spaced) binding motifs are very rare or commonly used software to identify the binding motifs cannot find them. To test the second hypothesis, I applied HOMER [106], MEME-CHIP [147], AMD [206], Bipad2 [207] and DIpartite [208] on the *ZNF768* dataset. Here, the last three programs were explicitly developed to find spaced motifs. However, none of the programs were able to identify the complete motif. When using only a small subset of all peaks as input, Bipad2 identified the motif. However, due to the long runtime caused by calculation of a bipartite multiple alignment, it is not suitable for processing large datasets in a reasonable time. Because of that, a bachelor student is currently developing a method to efficiently discover spaced motifs under my supervision. We plan to apply it to publicly available ChIP-seq datasets for zinc finger proteins to identify other zinc fingers that bind at spaced motifs. This includes a ChIP-seq dataset for 131 zinc finger proteins by Schmitges *et al.* and a ChIP-exo dataset for 222 zinc finger proteins by Imbeault *et al.* [209, 210].

In summary, this study provides an example of anciently introduced transposable elements being utilized as novel protein binding sites. This regulatory potential led to a protein required for the viability and proliferation of human cells, possibly by acting as a regulator of many other transcription factors.

3.3 Unraveling the function of *CKD12*

For this study, I developed new methods to identify and quantify the effect of *CDK12* inhibition on transcription. My analysis showed that the expression of a subset of genes terminated before reaching their annotated TTS. This observation helped to explain the genome instability defect often observed in cancers with mutated or lost *CDK12* function [170, 211] as transcription of many DNA repair and DNA replication genes was disrupted. Modeling of RNAPII progression based on ChIP-seq data revealed a loss of RNAPII occupancy at 3′ gene ends and a shift of P-Ser2 occupancy towards gene bodies for affected genes. Analysis of unpublished data generated after one hour of *CDK12* inhibition confirmed the RNAPII and P-Ser2 occupancy changes observed in this study. The transcript shortening defect was less pronounced, probably due to the shorter inhibition time, but already clearly visible. This indicates that an RNAPII processivity defect results in premature transcript termination.

However, the mechanisms remain unclear that cause RNAPII to prematurely terminate transcription. Typically, polyadenylation signals induce transcript cleavage and polyadenylation 10-30 bp downstream of the signal [212, 213]. An analysis of the 3′ ends of the poly(A)-selected RNA-seq dataset indicated that premature polyadenylation signals located in exons, introns and 3′ UTRs might be recognized instead of the usually used ones. This hypothesis is supported by the fact that transcript shortening was not observed for all genes but predominantly for long and polyadenylation-signal-rich genes. Other recently published studies came to similar results but focused only on intronic polyadenylation signals. For instance, Dubbury *et al.* proposed that *CDK12* can suppress the usage of intronic polyadenylation events [214]. Another study reported that premature cleavage and polyadenylation correlates with the presence of intronic polyadenylation signals when *CDK12* activity is lost [215]. However, my analysis did not reveal any particular enrichment of intronic polyadenylation signals over exonic ones when considering the length of introns and exons.

In any case, additional research on *CDK12* is required to better understand its versatile functions. *CDK12* was reported to influence several important cellular processes including mRNA splicing [167, 187, 214, 215], 3′ end processing [216, 217] and translation control [218]. Some of these functions can be explained by the ability of *CDK12* to phosphorylate the CTD of RNAPII [219, 220]. On the other hand, *CDK12* is also suspected to phosphorylate unknown substrates that are involved in transcription [177, 220]. Additionally, it has been proposed that *CDK12* maintains an RNA processing factory at the site of active transcription by participating in many protein-protein interactions [220]. Thus, we are currently collaborating with Blazek *et al.* on a project to identify proteins recruited by *CDK12* to *CDK12*-dependent genes that are required for normal transcription. In addition, Blazek *et al.* applied the same experimental setup as for this study for *CDK13*, which contains a domain that is very similar to *CDK12*'s kinase domain [220]. Just recently, Fan *et al.* reported a substantial functional redundancy between these two CDKs [221]. A preliminary analysis of this data showed neither transcript shortening nor RNAPII occupancy changes, which indicates that *CDK13* has a different role than *CDK12* or that its function is compensated by other kinases.

3.4 Conclusion

In summary, the WMS Watchdog developed in this thesis represents a general contribution to large-scale sequencing data analysis. The software itself and the established standard analysis workflows are publicly available and can therefore be used by experimentalists as well as bioinformaticians to analyze NGS data. Moreover, application of these workflows provided the basis for answering more specific biological questions. In particular, I identified the bipartite DNA-binding site of the zinc finger protein *ZNF768* and its likely evolutionary origin, and revealed a *CKD12*-dependent RNAPII processivity defect causing premature transcript shortening of hundreds of genes.

Acronyms

ChIP-seq ChIP-sequencing. 2, 4, 6, 7, 9, 10, 17, 18, 20, 22–26, 28, 29

RNA-seq RNA-sequencing. 2–6, 9, 10, 13, 15–18, 21–26, 30

ATP adenosine triphosphate. 21, 22

bp base pair. 3, 18–20, 22–25, 29, 30

CDK cyclin-dependent kinase. 21, 22, 30

cDNA complementary DNA. 3, 4

CTD C-terminal domain. 17, 22, 29, 30

DNA deoxyribonucleic acid. 1–4, 6, 10, 17–22, 28–30

GO Gene Ontology. 22

GUI graphical user interface. 8, 9, 12–14, 28

kb kilobase pair. 23

lncRNA long non-coding RNA. 3

MIR mammalian-wide interspersed repeat. 20, 21, 28, 29

miRNA microRNA. 2, 3

mRNA messenger RNA. 1–6, 21, 30

NGS next-generation sequencing. 2–8, 28, 30

RNA ribonucleic acid. 1–4, 17, 22, 23, 25, 30

RNAPII RNA polymerase II. 1, 2, 4, 9, 10, 17, 22–25, 29, 30

rRNA ribosomal RNA. 3, 4, 22

snRNA small nuclear RNA. 3

SRA NCBI sequence read archive. 22

tRNA transfer RNA. 3, 20

TSS transcription start site. 1, 4, 7, 23–25

TTS transcription termination site. 1, 4, 24, 25, 29

UTR untranslated region. 1, 21, 30

WMS workflow management system. 8, 9, 11, 12, 14–17, 27, 28, 30

References

[1] O. T. Avery, C. M. Macleod, and M. McCarty. Studies on the chemical nature of the substance inducing transformation of pneumococcal types. Induction of transformation by a desoxyribonucleic acid fraction isolated from Pneumococcus type III. *The Journal of experimental medicine*, 79(2):137–158, February 1944.

[2] J. D. Watson and F. H. Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, April 1953.

[3] R. E. Franklin and R. G. Gosling. Molecular configuration in sodium thymonucleate. *Nature*, 171(4356):740–741, April 1953.

[4] H. R. Petty. Overview of the physical state of proteins within cells. *Current protocols in cell biology*, Chapter 5:Unit 5.1, May 2001.

[5] E. Nogales, R. K. Louder, and Y. He. Structural Insights into the Eukaryotic Transcription Initiation Machinery. *Annual review of biophysics*, 46:59–83, May 2017.

[6] F. Jacob and J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of molecular biology*, 3:318–356, June 1961.

[7] P. A. Sharp. Split genes and RNA splicing. *Cell*, 77(6):805–815, June 1994.

[8] L. Herzel, D. S. M. Ottoz, T. Alpert, and K. M. Neugebauer. Splicing and transcription touch base: co-transcriptional spliceosome assembly and function. *Nature reviews. Molecular cell biology*, 18(10):637–650, October 2017.

[9] F. Gros, H. Hiatt, W. Gilbert, C. G. Kurland, R. W. Risebrough, and J. D. Watson. Unstable ribonucleic acid revealed by pulse labelling of Escherichia coli. *Nature*, 190:581–585, May 1961.

[10] S. Brenner, F. Jacob, and M. Meselson. An unstable intermediate carrying information from genes to ribosomes for protein synthesis. *Nature*, 190:576–581, May 1961.

[11] F. H. Crick, L. Barnett, S. Brenner, and R. J. Watts-Tobin. General nature of the genetic code for proteins. *Nature*, 192:1227–1232, December 1961.

[12] M. Nirenberg, T. Caskey, R. Marshall, R. Brimacombe, D. Kellogg, B. Doctor, D. Hatfield, J. Levin, F. Rottman, S. Pestka, M. Wilcox, and F. Anderson. The RNA code and protein synthesis. *Cold Spring Harbor symposia on quantitative biology*, 31:11–24, 1966.

[13] G. S. Wilkie, K. S. Dickson, and N. K. Gray. Regulation of mRNA translation by 5'- and 3'-UTR-binding factors. *Trends in biochemical sciences*, 28(4):182–188, April 2003.

[14] J. Soutourina. Transcription regulation by the Mediator complex. *Nature reviews. Molecular cell biology*, 19(4):262–274, April 2018.

[15] V. Haberle and A. Stark. Eukaryotic core promoters and the functional basis of transcription initiation. *Nature reviews. Molecular cell biology*, 19(10):621–637, October 2018.

[16] S. L. Klemm, Z. Shipony, and W. J. Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature reviews. Genetics*, 20(4):207–220, April 2019.

[17] A. Mayer, H. M. Landry, and L. S. Churchman. Pause & go: from the discovery of RNA polymerase pausing to its functional implications. *Current opinion in cell biology*, 46:72–80, June 2017.

[18] C. R. Bartman, N. Hamagami, C. A. Keller, B. Giardine, R. C. Hardison, G. A. Blobel, and A. Raj. Transcriptional Burst Initiation and Polymerase Pause Release Are Key Control Points of Transcriptional Regulation. *Molecular cell*, 73(3):519–532.e4, February 2019.

[19] R. E. Halbeisen, A. Galgano, T. Scherrer, and A. P. Gerber. Post-transcriptional gene regulation: from genome-wide studies to principles. *Cellular and molecular life sciences : CMLS*, 65(5):798–813, March 2008.

[20] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12):5463–5467, December 1977.

[21] G. B. Kolata. The 1980 Nobel Prize in Chemistry. *Science (New York, N.Y.)*, 210(4472):887–889, November 1980.

[22] J. Shendure and H. Ji. Next-generation DNA sequencing. *Nature biotechnology*, 26(10):1135–1145, October 2008.

[23] B. L. Karger and A. Guttman. DNA sequencing by CE. *Electrophoresis*, 30 Suppl 1:S196–S202, June 2009.

[24] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, October 2004.

[25] E. C. Hayden. Technology: The $1,000 genome. *Nature*, 507(7492):294–295, March 2014.

[26] A. Kahvejian, J. Quackenbush, and J. F. Thompson. What would you do if you could sequence everything? *Nature biotechnology*, 26(10):1125–1133, October 2008.

[27] E.N.C.O.D.E. Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, September 2012.

[28] Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, October 2008.

[29] 1000 Genomes Project Consortium, A. Auton, L. D. Brooks, R. M. Durbin, E. P. Garrison, H. M. Kang, J. O. Korbel, J. L. Marchini, S. McCarthy, G. A. McVean, and G. R. Abecasis. A global reference for human genetic variation. *Nature*, 526(7571):68–74, October 2015.

[30] H. K. Tabor, P. L. Auer, S. M. Jamal, J. X. Chong, J.-H. Yu, A. S. Gordon, T. A. Graubert, C. J. O'Donnell, S. S. Rich, D. A. Nickerson, N. E. S. Project, and M. J. Bamshad. Pathogenic variants for Mendelian and complex traits in exomes of 6,517 European and African Americans: implications for the return of incidental results. *American journal of human genetics*, 95(2):183–193, August 2014.

[31] F. E. Dewey, M. F. Murray, J. D. Overton, L. Habegger, J. B. Leader, S. N. Fetterolf, C. O'Dushlaine, C. V. Van Hout, J. Staples, C. Gonzaga-Jauregui, R. Metpally, S. A. Pendergrass, M. A. Giovanni, H. L. Kirchner, S. Balasubramanian, N. S. Abul-Husn, D. N. Hartzel, D. R. Lavage, K. A. Kost, J. S. Packer, A. E. Lopez, J. Penn, S. Mukherjee, N. Gosalia, M. Kanagaraj, A. H. Li, L. J. Mitnaul, L. J. Adams, T. N. Person, K. Praveen, A. Marcketta, M. S. Lebo, C. A. Austin-Tse, H. M. Mason-Suares, S. Bruse, S. Mellis, R. Phillips, N. Stahl, A. Murphy, A. Economides, K. A. Skelding, C. D. Still, J. R. Elmore, I. B. Borecki, G. D. Yancopoulos, F. D. Davis, W. A. Faucett, O. Gottesman, M. D. Ritchie, A. R. Shuldiner, J. G. Reid, D. H. Ledbetter, A. Baras, and D. J. Carey. Distribution and clinical impact of functional variants in 50,726 whole-exome sequences from the DiscovEHR study. *Science (New York, N.Y.)*, 354(6319), December 2016.

[32] G. N. Samuel and B. Farsides. The UK's 100,000 Genomes Project: manifesting policymakers' expectations. *New genetics and society*, 36(4):336–353, 2017.

[33] U. I. Schwarz, M. Gulilat, and R. B. Kim. The Role of Next-Generation Sequencing in Pharmacogenetics and Pharmacogenomics. *Cold Spring Harbor perspectives in medicine*, 9(2), February 2019.

[34] S. A. Jeon, J. L. Park, J.-H. Kim, J. H. Kim, Y. S. Kim, J. C. Kim, and S.-Y. Kim. Comparison of the MGISEQ-2000 and Illumina HiSeq 4000 sequencing platforms for RNA sequencing. *Genomics & informatics*, 17(3):e32, September 2019.

[35] Illumina. An introduction to Next-Generation Sequencing Technology. `https://www.illumina.com/documents/products/illumina_sequencing_introduction.pdf`. Accessed: 2020-01-21.

[36] J. M. Heather and B. Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1–8, January 2016.

[37] S. Goodwin, J. D. McPherson, and W. R. McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature reviews. Genetics*, 17(6):333–351, May 2016.

[38] Illumina. Illumina sequencing platforms. `https://emea.illumina.com/systems/sequencing-platforms.html`. Accessed: 2020-01-21.

[39] G. Lightbody, V. Haberland, F. Browne, L. Taggart, H. Zheng, E. Parkes, and J. K. Blayney. Review of applications of high-throughput sequencing in personalized medicine: barriers and facilitators of future progress in research and clinical application. *Briefings in bioinformatics*, 20(5):1795–1811, September 2019.

[40] G. England. Genomics England and Illumina partner to deliver whole genome sequencing for England's NHS Genomic Medicine Service. `https://www.genomicsengland.co.uk/genomics-england-illumina-partner-nhs-genomic-medicine-service/`. Accessed: 2020-01-21.

[41] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell. *Molecular Cell Biology*. W.H. Freeman, New York, 4th edition, 2000.

[42] A. C. C. Sá, W. Sadee, and J. A. Johnson. Whole Transcriptome Profiling: An RNA-Seq Primer and Implications for Pharmacogenomics Research. *Clinical and translational science*, 11(2):153–161, March 2018.

[43] S. R. Head, H. K. Komori, S. A. LaMere, T. Whisenant, F. Van Nieuwerburgh, D. R. Salomon, and P. Ordoukhanian. Library construction for next-generation sequencing: overviews and challenges. *BioTechniques*, 56(2):61–4, 66, 68, passim, 2014.

[44] W. Zhao, X. He, K. A. Hoadley, J. S. Parker, D. N. Hayes, and C. M. Perou. Comparison of RNA-Seq by poly (A) capture, ribosomal RNA depletion, and DNA microarray for expression profiling. *BMC genomics*, 15:419, June 2014.

[45] N. Chowdhury and A. Bagchi. An Overview of DNA-Protein Interactions. *Current Chemical Biology*, 9(2):73–83, 2015.

[46] P. J. Park. ChIP-seq: advantages and challenges of a maturing technology. *Nature reviews. Genetics*, 10(10):669–680, October 2009.

[47] Q. He, J. Johnston, and J. Zeitlinger. ChIP-nexus enables improved detection of in vivo transcription factor binding footprints. *Nature biotechnology*, 33(4):395–401, April 2015.

[48] R. B. Darnell. HITS-CLIP: panoramic views of protein-RNA regulation in living cells. *Wiley interdisciplinary reviews. RNA*, 1(2):266–286, 2010.

[49] Q. Sheng, K. Vickers, S. Zhao, J. Wang, D. C. Samuels, O. Koues, Y. Shyr, and Y. Guo. Multi-perspective quality control of Illumina RNA sequencing data analysis. *Briefings in functional genomics*, 16(4):194–204, July 2017.

[50] Babraham, Bioinformatics Institute. FastQC: A quality control tool for high throughput sequence data. `http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`, 2014.

[51] M. Martin. CUTADAPT removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17, 08 2011.

[52] A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics (Oxford, England)*, 30(15):2114–2120, August 2014.

[53] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3):R25, 2009.

[54] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, 25(14):1754–1760, July 2009.

[55] P. Ferragina and G. Manzini. Opportunistic Data Structures with Applications. In *FOCS*, pages 390–398. IEEE Computer Society, 2000.

[56] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):R36, April 2013.

[57] D. Kim, J. M. Paggi, C. Park, C. Bennett, and S. L. Salzberg. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature biotechnology*, 37(8):907–915, August 2019.

[58] A. Dobin and T. R. Gingeras. Mapping RNA-seq Reads with STAR. *Current protocols in bioinformatics*, 51:11.14.1–11.14.19, September 2015.

[59] Z. Zhang and M. Gerstein. Large-scale analysis of pseudogenes in the human genome. *Current opinion in genetics & development*, 14(4):328–335, August 2004.

[60] T. Bonfert, E. Kirner, G. Csaba, R. Zimmer, and C. C. Friedel. ContextMap 2: fast and accurate context-based RNA-seq mapping. *BMC bioinformatics*, 16:122, April 2015.

[61] S. Anders, P. T. Pyl, and W. Huber. HTSeq–a Python framework to work with high-throughput sequencing data. *Bioinformatics (Oxford, England)*, 31(2):166–169, January 2015.

[62] Y. Liao, G. K. Smyth, and W. Shi. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics (Oxford, England)*, 30(7):923–930, April 2014.

[63] D. Risso, K. Schwartz, G. Sherlock, and S. Dudoit. GC-content normalization for RNA-Seq data. *BMC bioinformatics*, 12:480, December 2011.

[64] A. Oshlack and M. J. Wakefield. Transcript length bias in RNA-seq data confounds systems biology. *Biology direct*, 4:14, April 2009.

[65] C. Evans, J. Hardin, and D. M. Stoebel. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. *Briefings in bioinformatics*, 19(5):776–792, September 2018.

[66] Y. Liu, J. Zhou, and K. P. White. RNA-seq differential expression studies: more sequence or more replication? *Bioinformatics (Oxford, England)*, 30(3):301–304, February 2014.

[67] S. Lamarre, P. Frasse, M. Zouine, D. Labourdette, E. Sainderichin, G. Hu, V. Le Berre-Anton, M. Bouzayen, and E. Maza. Optimization of an RNA-Seq Differential Gene Expression Analysis Depending on Biological Replicate Number and Library Size. *Frontiers in plant science*, 9:108, 2018.

[68] W. S. Noble. How does multiple testing correction work? *Nature biotechnology*, 27(12):1135–1137, December 2009.

[69] M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics (Oxford, England)*, 26(1):139–140, January 2010.

[70] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids research*, 43(7):e47, April 2015.

[71] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome biology*, 11(10):R106, 2010.

[72] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology*, 15(12):550, 2014.

[73] V. M. Kvam, P. Liu, and Y. Si. A comparison of statistical methods for detecting differentially expressed genes from RNA-seq data. *American journal of botany*, 99(2):248–256, February 2012.

[74] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, G. Guernec, B. Jagla, L. Jouneau, D. Laloë, C. Le Gall, B. Schaëffer, S. Le Crom, M. Guedj, F. Jaffrézic, and French StatOmique Consortium. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in bioinformatics*, 14(6):671–683, November 2013.

[75] F. Rapaport, R. Khanin, Y. Liang, M. Pirun, A. Krek, P. Zumbo, C. E. Mason, N. D. Socci, and D. Betel. Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. *Genome biology*, 14(9):R95, 2013.

[76] C. Soneson and M. Delorenzi. A comparison of methods for differential expression analysis of RNA-seq data. *BMC bioinformatics*, 14:91, March 2013.

[77] F. E. Baralle and J. Giudice. Alternative splicing as a regulator of development and tnumber identity. *Nature reviews. Molecular cell biology*, 18(7):437–451, July 2017.

[78] E. Park, Z. Pan, Z. Zhang, L. Lin, and Y. Xing. The Expanding Landscape of Alternative Splicing Variation in Human Populations. *American journal of human genetics*, 102(1):11–26, January 2018.

[79] H. Dvinge. Regulation of alternative mRNA splicing: old players and new perspectives. *FEBS letters*, 592(17):2987–3006, September 2018.

[80] C. Trapnell, D. G. Hendrickson, M. Sauvageau, L. Goff, J. L. Rinn, and L. Pachter. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature biotechnology*, 31(1):46–53, January 2013.

[81] Y. Hu, Y. Huang, Y. Du, C. F. Orellana, D. Singh, A. R. Johnson, A. Monroy, P.-F. Kuan, S. M. Hammond, L. Makowski, S. H. Randell, D. Y. Chiang, D. N. Hayes, C. Jones, Y. Liu, J. F. Prins, and J. Liu. DiffSplice: the genome-wide detection of differential splicing events with RNA-seq. *Nucleic acids research*, 41(2):e39, January 2013.

[82] I. Mandric, Y. Temate-Tiagueu, T. Shcheglova, S. Al Seesi, A. Zelikovsky, and I. I. Mandoiu. Fast bootstrapping-based estimation of confidence intervals of expression levels and differential expression from RNA-Seq data. *Bioinformatics (Oxford, England)*, 33(20):3302–3304, October 2017.

[83] S. Anders, A. Reyes, and W. Huber. Detecting differential usage of exons from RNA-seq data. *Genome research*, 22(10):2008–2017, October 2012.

[84] S. W. Hartley and J. C. Mullikin. Detection and visualization of differential splicing in RNA-Seq data with JunctionSeq. *Nucleic acids research*, 44(15):e127, September 2016.

[85] S. Shen, J. W. Park, Z.-x. Lu, L. Lin, M. D. Henry, Y. N. Wu, Q. Zhou, and Y. Xing. rMATS: robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proceedings of the National Academy of Sciences of the United States of America*, 111(51):E5593–E5601, December 2014.

[86] S. S. Norton, J. Vaquero-Garcia, N. F. Lahens, G. R. Grant, and Y. Barash. Outlier detection for improved differential splicing quantification from RNA-Seq experiments with replicates. *Bioinformatics (Oxford, England)*, 34(9):1488–1497, May 2018.

[87] J. L. Trincado, J. C. Entizne, G. Hysenaj, B. Singh, M. Skalic, D. J. Elliott, and E. Eyras. SUPPA2: fast, accurate, and uncertainty-aware differential splicing analysis across multiple conditions. *Genome biology*, 19(1):40, March 2018.

[88] Y. Katz, E. T. Wang, E. M. Airoldi, and C. B. Burge. Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nature methods*, 7(12):1009–1015, December 2010.

[89] J. E. Hooper. A survey of software for genome-wide discovery of differential splicing in RNA-Seq data. *Human genomics*, 8:3, January 2014.

[90] L. Ding, E. Rath, and Y. Bai. Comparison of Alternative Splicing Junction Detection Tools Using RNA-Seq Data. *Current genomics*, 18(3):268–277, June 2017.

[91] A. Mehmood, A. Laiho, M. S. Venäläinen, A. J. McGlinchey, N. Wang, and L. L. Elo. Systematic evaluation of differential splicing tools for RNA-seq studies. *Briefings in bioinformatics*, December 2019.

[92] H. Xu, L. Handoko, X. Wei, C. Ye, J. Sheng, C.-L. Wei, F. Lin, and W.-K. Sung. A signal-noise model for significance analysis of ChIP-seq with negative control. *Bioinformatics (Oxford, England)*, 26(9):1199–1204, May 2010.

[93] B. L. Kidder, G. Hu, and K. Zhao. ChIP-Seq: technical considerations for obtaining high-quality data. *Nature immunology*, 12(10):918–922, September 2011.

[94] S. A. Lambert, A. Jolma, L. F. Campitelli, P. K. Das, Y. Yin, M. Albu, X. Chen, J. Taipale, T. R. Hughes, and M. T. Weirauch. The Human Transcription Factors. *Cell*, 175(2):598–599, October 2018.

[95] T. Zhang, S. Cooper, and N. Brockdorff. The interplay of histone modifications - writers that read. *EMBO reports*, 16(11):1467–1481, November 2015.

[96] Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li, and X. S. Liu. Model-based analysis of ChIP-Seq (MACS). *Genome biology*, 9(9):R137, 2008.

[97] R. Jothi, S. Cuddapah, A. Barski, K. Cui, and K. Zhao. Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data. *Nucleic acids research*, 36(16):5221–5231, September 2008.

[98] C. Zang, D. E. Schones, C. Zeng, K. Cui, K. Zhao, and W. Peng. A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. *Bioinformatics (Oxford, England)*, 25(15):1952–1958, August 2009.

[99] A. P. Boyle, J. Guinney, G. E. Crawford, and T. S. Furey. F-Seq: a feature density estimator for high-throughput sequence tags. *Bioinformatics (Oxford, England)*, 24(21):2537–2538, November 2008.

[100] A. P. Fejes, G. Robertson, M. Bilenky, R. Varhol, M. Bainbridge, and S. J. M. Jones. FindPeaks 3.1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology. *Bioinformatics (Oxford, England)*, 24(15):1729–1730, August 2008.

[101] J. Rozowsky, G. Euskirchen, R. K. Auerbach, Z. D. Zhang, T. Gibson, R. Bjornson, N. Carriero, M. Snyder, and M. B. Gerstein. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature biotechnology*, 27(1):66–75, January 2009.

[102] Y. Guo, S. Mahony, and D. K. Gifford. High resolution genome wide binding event finding and motif discovery reveals transcription factor spatial binding constraints. *PLoS computational biology*, 8(8):e1002638, 2012.

[103] R. Thomas, S. Thomas, A. K. Holloway, and K. S. Pollard. Features that define the best ChIP-seq peak calling algorithms. *Briefings in bioinformatics*, 18(3):441–450, May 2017.

[104] E. G. Wilbanks and M. T. Facciotti. Evaluation of algorithm performance in ChIP-seq peak detection. *PloS one*, 5(7):e11471, July 2010.

[105] H. Zhang, L. Zhu, and D.-S. Huang. WSMD: weakly-supervised motif discovery in transcription factor ChIP-seq data. *Scientific reports*, 7(1):3217, June 2017.

[106] S. Heinz, C. Benner, N. Spann, E. Bertolino, Y. C. Lin, P. Laslo, J. X. Cheng, C. Murre, H. Singh, and C. K. Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular cell*, 38(4):576–589, May 2010.

[107] T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, 2:28–36, 1994.

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

[109] J. van Helden, B. André, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of molecular biology*, 281(5):827–842, September 1998.

[110] M. C. Frith, N. F. W. Saunders, B. Kobe, and T. L. Bailey. Discovering sequence motifs with arbitrary insertions and deletions. *PLoS computational biology*, 4(4):e1000071, May 2008.

[111] F. A. Hashim, M. S. Mabrouk, and W. Al-Atabany. Review of Different Sequence Motif Finding Algorithms. *Avicenna journal of medical biotechnology*, 11(2):130–148, 2019.

[112] G. Yu, L.-G. Wang, and Q.-Y. He. ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization. *Bioinformatics (Oxford, England)*, 31(14):2382–2383, July 2015.

[113] P. Kulkarni and P. Frommolt. Challenges in the Setup of Large-scale Next-Generation Sequencing Analysis Workflows. *Computational and structural biotechnology journal*, 15:471–477, 2017.

[114] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, May 2016.

[115] R. D. Peng. Reproducible research in computational science. *Science (New York, N.Y.)*, 334(6060):1226–1227, December 2011.

[116] S. Kanwal, F. Z. Khan, A. Lonie, and R. O. Sinnott. Investigating reproducibility and tracking provenance - A genomic workflow case study. *BMC bioinformatics*, 18(1):337, July 2017.

[117] Y.-M. Kim, J.-B. Poline, and G. Dumas. Experimenting with reproducibility: a case study of robustness in bioinformatics. *GigaScience*, 7(7), July 2018.

[118] W. Guo, N. Tzioutziou, G. Stephen, I. Milne, C. Calixto, R. Waugh, J. W. S. Brown, and R. Zhang. 3D RNA-seq - a powerful and flexible tool for rapid and accurate differential expression and alternative splicing analysis of RNA-seq data for biologists. *bioRxiv*, 2019.

[119] Z. Sundararajan, R. Knoll, P. Hombach, M. Becker, J. L. Schultze, and T. Ulas. Shiny-Seq: advanced guided transcriptome analysis. *BMC research notes*, 12(1):432, July 2019.

[120] E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Cech, J. Chilton, D. Clements, N. Coraor, B. A. Grüning, A. Guerler, J. Hillman-Jackson, S. Hiltemann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko, and D. Blankenberg. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic acids research*, 46(W1):W537–W544, July 2018.

[121] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel. KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007.

[122] J. Köster and S. Rahmann. Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics (Oxford, England)*, 28(19):2520–2522, October 2012.

[123] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame. Nextflow enables reproducible computational workflows. *Nature biotechnology*, 35(4):316–319, April 2017.

[124] M. Kluge and C. C. Friedel. Watchdog - a workflow management system for the distributed analysis of large-scale experimental data. *BMC Bioinformatics*, 19(1):97, March 2018.

[125] M. Kluge, M.-S. Friedl, A. L. Menzel, and C. C. Friedel. Watchdog 2.0: New developments for reusability, reproducibility, and workflow execution. *GigaScience*, 9(6):giaa068, June 2020.

[126] M. Rohrmoser, M. Kluge, Y. Yahia, A. Gruber-Eber, M. A. Maqbool, I. Forné, S. Krebs, H. Blum, A. K. Greifenberg, M. Geyer, N. Descostes, A. Imhof, J.-C. Andrau, C. C. Friedel, and D. Eick. MIR sequences recruit zinc finger protein ZNF768 to expressed genes. *Nucleic acids research*, 47(2):700–715, January 2019.

[127] A. P. Chirackal Manavalan, K. Pilarova, M. Kluge, K. Bartholomeeusen, M. Rajecky, J. Oppelt, P. Khirsariya, K. Paruch, L. Krejci, C. C. Friedel, and D. Blazek. CDK12 controls G1/S progression by regulating RNAPII processivity at core DNA replication genes. *EMBO reports*, 20(9):e47592, September 2019.

[128] J. M. Matthews and M. Sunde. Zinc fingers–folds for many occasions. *IUBMB life*, 54(6):351–355, December 2002.

[129] S. S. Krishna, I. Majumdar, and N. V. Grishin. Structural classification of zinc fingers: survey and summary. *Nucleic acids research*, 31(2):532–550, January 2003.

[130] H. D. Tadepally, G. Burger, and M. Aubry. Evolution of C2H2-zinc finger genes and subfamilies in mammals: species-specific duplication and loss of clusters, genes and effector domains. *BMC evolutionary biology*, 8:176, June 2008.

[131] J. H. Laity, B. M. Lee, and P. E. Wright. Zinc finger proteins: new insights into structural and functional diversity. *Current opinion in structural biology*, 11(1):39–46, February 2001.

[132] J. Jen and Y.-C. Wang. Zinc finger proteins in cancer progression. *Journal of biomedical science*, 23(1):53, July 2016.

[133] B. Doran, N. Gherbesi, G. Hendricks, R. A. Flavell, R. J. Davis, and L. Gangwani. Deficiency of the zinc finger protein ZPR1 causes neurodegeneration. *Proceedings of the National Academy of Sciences of the United States of America*, 103(19):7471–7475, May 2006.

[134] P. I. Joyce, P. Fratta, A. S. Landman, P. Mcgoldrick, H. Wackerhage, M. Groves, B. S. Busam, J. Galino, S. Corrochano, O. A. Beskina, C. Esapa, E. Ryder, S. Carter, M. Stewart, G. Codner, H. Hilton, L. Teboul, J. Tucker, A. Lionikas, J. Estabel, R. Ramirez-Solis, J. K. White, S. Brandner, V. Plagnol, D. L. H. Bennet, A. Y. Abramov, L. Greensmith, E. M. C. Fisher, and A. Acevedo-Arozena. Deficiency of the zinc finger protein ZFP106 causes motor and sensory neurodegeneration. *Human molecular genetics*, 25(2):291–307, January 2016.

[135] Y. Zhang, Z. Xie, L. Zhou, L. Li, H. Zhang, G. Zhou, X. Ma, P. L. Herrera, Z. Liu, M. J. Grusby, and W. J. Zhang. The zinc finger protein ZBTB20 regulates transcription of fructose-1,6-bisphosphatase 1 and β cell function in mice. *Gastroenterology*, 142(7):1571–1580.e6, June 2012.

[136] D. A. Buchner, A. Charrier, E. Srinivasan, L. Wang, M. T. Paulsen, M. Ljungman, D. Bridges, and A. R. Saltiel. Zinc finger protein 407 (ZFP407) regulates insulin-stimulated glucose uptake and glucose transporter 4 (Glut4) mRNA. *The Journal of biological chemistry*, 290(10):6376–6386, March 2015.

[137] UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, January 2019.

[138] N. J. Fuda, M. B. Ardehali, and J. T. Lis. Defining mechanisms that regulate RNA polymerase II transcription in vivo. *Nature*, 461(7261):186–192, September 2009.

[139] R. Schüller, I. Forné, T. Straub, A. Schreieck, Y. Texier, N. Shah, T.-M. Decker, P. Cramer, A. Imhof, and D. Eick. Heptad-Specific Phosphorylation of RNA Polymerase II CTD. *Molecular cell*, 61(2):305–314, January 2016.

[140] J. V. Pulvertaft. Cytology of Burkitt's Tumour (African Lymphoma). *Lancet (London, England)*, 1(7327):238–240, February 1964.

[141] J. Pontén and E. Saksela. Two established in vitro cell lines from human mesenchymal tumours. *International journal of cancer*, 2(5):434–447, September 1967.

[142] J. Fogh. Human tumor lines for cancer research. *Cancer investigation*, 4(2):157–184, 1986.

[143] G. W. Bornkamm, C. Berens, C. Kuklik-Roos, J.-M. Bechet, G. Laux, J. Bachl, M. Korndoerfer, M. Schlee, M. Hölzel, A. Malamoussi, R. D. Chapman, F. Nimmerjahn, J. Mautner, W. Hillen, H. Bujard, and J. Feuillard. Stringent doxycycline-dependent control of gene activities using an episomal one-vector system. *Nucleic acids research*, 33(16):e137, September 2005.

[144] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, 25(16):2078–2079, August 2009.

[145] L. Wang, S. Wang, and W. Li. RSeQC: quality control of RNA-seq experiments. *Bioinformatics (Oxford, England)*, 28(16):2184–2185, August 2012.

[146] A. Frankish, M. Diekhans, A.-M. Ferreira, R. Johnson, I. Jungreis, J. Loveland, J. M. Mudge, C. Sisu, J. Wright, J. Armstrong, I. Barnes, A. Berry, A. Bignell, S. Carbonell Sala, J. Chrast, F. Cunningham, T. Di Domenico, S. Donaldson, I. T. Fiddes, C. García Girón, J. M. Gonzalez, T. Grego, M. Hardy, T. Hourlier, T. Hunt, O. G. Izuogu, J. Lagarde, F. J. Martin, L. Martínez, S. Mohanan, P. Muir, F. C. P. Navarro, A. Parker, B. Pei, F. Pozo, M. Ruffier, B. M. Schmitt, E. Stapleton, M.-M. Suner, I. Sycheva, B. Uszczynska-Ratajczak, J. Xu, A. Yates, D. Zerbino, Y. Zhang, B. Aken, J. S. Choudhary, M. Gerstein, R. Guigó, T. J. P. Hubbard, M. Kellis, B. Paten, A. Reymond, M. L. Tress, and P. Flicek. GENCODE reference annotation for the human and mouse genomes. *Nucleic acids research*, 47(D1):D766–D773, January 2019.

[147] P. Machanick and T. L. Bailey. MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics (Oxford, England)*, 27(12):1696–1697, June 2011.

[148] A. R. Quinlan and I. M. Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics (Oxford, England)*, 26(6):841–842, March 2010.

[149] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[150] T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble. MEME SUITE: tools for motif discovery and searching. *Nucleic acids research*, 37(Web Server number):W202–W208, July 2009.

[151] T. D. Schneider and R. M. Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic acids research*, 18(20):6097–6100, October 1990.

[152] L. M. Hellman and M. G. Fried. Electrophoretic mobility shift assay (EMSA) for detecting protein-nucleic acid interactions. *Nature protocols*, 2(8):1849–1861, 2007.

[153] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner. WebLogo: a sequence logo generator. *Genome research*, 14(6):1188–1190, June 2004.

[154] A. Smit, R. Hubley, and P. Green. RepeatMasker Open-4.0 - 2013-2015. `http://www.repeatmasker.org`.

[155] A. F. Smit and A. D. Riggs. MIRs are classic, tRNA-derived SINEs that amplified before the mammalian radiation. *Nucleic acids research*, 23(1):98–102, January 1995.

[156] J. Jurka, E. Zietkiewicz, and D. Labuda. Ubiquitous mammalian-wide interspersed repeats (MIRs) are molecular fossils from the mesozoic era. *Nucleic acids research*, 23(1):170–175, January 1995.

[157] N. Gilbert and D. Labuda. Evolutionary inventions and continuity of CORE-SINEs in mammals. *Journal of molecular biology*, 298(3):365–377, May 2000.

[158] D. Jjingo, A. B. Conley, J. Wang, L. Mariño-Ramírez, V. V. Lunyak, and I. K. Jordan. Mammalian-wide interspersed repeat (MIR)-derived enhancers and the regulation of human gene expression. *Mobile DNA*, 5:14, 2014.

[159] R. Hubley, R. D. Finn, J. Clements, S. R. Eddy, T. A. Jones, W. Bao, A. F. A. Smit, and T. J. Wheeler. The Dfam database of repetitive DNA families. *Nucleic acids research*, 44(D1):D81–D89, January 2016.

[160] K. S. Pollard, M. J. Hubisz, K. R. Rosenbloom, and A. Siepel. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome research*, 20(1):110–121, January 2010.

[161] The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic acids research*, 46(5):2699, March 2018.

[162] T. Hunter. A thousand and one protein kinases. *Cell*, 50(6):823–829, September 1987.

[163] D. O. Morgan. Principles of CDK regulation. *Nature*, 374(6518):131–134, March 1995.

[164] M. Malumbres, E. Harlow, T. Hunt, T. Hunter, J. M. Lahti, G. Manning, D. O. Morgan, L.-H. Tsai, and D. J. Wolgemuth. Cyclin-dependent kinases: a family portrait. *Nature cell biology*, 11(11):1275–1276, November 2009.

[165] P. Loyer, J. H. Trembley, R. Katona, V. J. Kidd, and J. M. Lahti. Role of CDK/cyclin complexes in transcription and RNA splicing. *Cellular signalling*, 17(9):1033–1051, September 2005.

[166] T. K. Ko, E. Kelly, and J. Pines. CrkRS: a novel conserved Cdc2-related protein kinase that colocalises with SC35 speckles. *Journal of cell science*, 114(Pt 14):2591–2603, July 2001.

[167] H.-H. Chen, Y.-C. Wang, and M.-J. Fann. Identification and characterization of the CDK12/cyclin L1 complex involved in alternative splicing regulation. *Molecular and cellular biology*, 26(7):2736–2745, April 2006.

[168] D. Blazek, J. Kohoutek, K. Bartholomeeusen, E. Johansen, P. Hulinkova, Z. Luo, P. Cimermancic, J. Ule, and B. M. Peterlin. The Cyclin K/Cdk12 complex maintains genomic stability via regulation of expression of DNA damage response genes. *Genes & development*, 25(20):2158–2172, October 2011.

[169] X. Li, N. Chatterjee, K. Spirohn, M. Boutros, and D. Bohmann. Cdk12 Is A Gene-Selective RNA Polymerase II Kinase That Regulates a Subset of the Transcriptome, Including Nrf2 Target Genes. *Scientific reports*, 6:21455, February 2016.

[170] T. Popova, E. Manié, V. Boeva, A. Battistella, O. Goundiam, N. K. Smith, C. R. Mueller, V. Raynal, O. Mariani, X. Sastre-Garau, and M.-H. Stern. Ovarian Cancers Harboring Inactivating Mutations in CDK12 Display a Distinct Genomic Instability Pattern Characterized by Large Tandem Duplications. *Cancer research*, 76(7):1882–1891, April 2016.

[171] F. Menghi, F. P. Barthel, V. Yadav, M. Tang, B. Ji, Z. Tang, G. W. Carter, Y. Ruan, R. Scully, R. G. W. Verhaak, J. Jonkers, and E. T. Liu. The Tandem Duplicator Phenotype Is a Prevalent Genome-Wide Cancer Configuration Driven by Distinct Gene Mutations. *Cancer cell*, 34(2):197–210.e5, August 2018.

[172] D. A. Quigley, H. X. Dang, S. G. Zhao, P. Lloyd, R. Aggarwal, J. J. Alumkal, A. Foye, V. Kothari, M. D. Perry, A. M. Bailey, D. Playdle, T. J. Barnard, L. Zhang, J. Zhang, J. F. Youngren, M. P. Cieslik, A. Parolia, T. M. Beer, G. Thomas, K. N. Chi, M. Gleave, N. A. Lack, A. Zoubeidi, R. E. Reiter, M. B. Rettig, O. Witte, C. J. Ryan, L. Fong, W. Kim, T. Friedlander, J. Chou, H. Li, R. Das, H. Li, R. Moussavi-Baygi, H. Goodarzi, L. A. Gilbert, P. N. Lara, C. P. Evans, T. C. Goldstein, J. M. Stuart, S. A. Tomlins, D. E. Spratt, R. K. Cheetham, D. T. Cheng, K. Farh, J. S. Gehring, J. Hakenberg, A. Liao, P. G. Febbo, J. Shon, B. Sickler, S. Batzoglou, K. E. Knudsen, H. H. He, J. Huang, A. W. Wyatt, S. M. Dehm, A. Ashworth, A. M. Chinnaiyan, C. A. Maher, E. J. Small, and F. Y. Feng. Genomic Hallmarks and Structural Variation in Metastatic Prostate Cancer. *Cell*, 174(3):758–769.e9, July 2018.

[173] M. A. Reimers, S. M. Yip, L. Zhang, M. Cieslik, M. Dhawan, B. Montgomery, A. W. Wyatt, K. N. Chi, E. J. Small, A. M. Chinnaiyan, A. S. Alva, F. Y. Feng, and J. Chou. Clinical Outcomes in Cyclin-dependent Kinase 12 Mutant Advanced Prostate Cancer. *European urology*, October 2019.

[174] F. Peng, C. Yang, Y. Kong, X. Huang, Y. Chen, Y. Zhou, X. Xie, and P. Liu. CDK12 Promotes Breast Cancer Progression and Maintains Stemness by Activating c-myc/β-catenin Signaling. *Current cancer drug targets*, November 2019.

[175] R. Chilà, F. Guffanti, and G. Damia. Role and therapeutic potential of CDK12 in human cancers. *Cancer treatment reviews*, 50:83–88, November 2016.

[176] G. Y. L. Lui, C. Grandori, and C. J. Kemp. CDK12: an emerging therapeutic target for cancer. *Journal of clinical pathology*, 71(11):957–962, November 2018.

[177] S. H. Choi, S. Kim, and K. A. Jones. Gene expression regulation by CDK12: a versatile kinase in cancer with functions beyond CTD phosphorylation. *Experimental & molecular medicine*, May 2020.

[178] M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna, and E. Charpentier. A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science (New York, N.Y.)*, 337(6096):816–821, August 2012.

[179] L. Cipak, C. Zhang, I. Kovacikova, C. Rumpf, E. Miadokova, K. M. Shokat, and J. Gregan. Generation of a set of conditional analog-sensitive alleles of essential protein kinases in the fission yeast Schizosaccharomyces pombe. *Cell cycle (Georgetown, Tex.)*, 10(20):3527–3532, October 2011.

[180] N. J. Camlin and J. P. Evans. Auxin-inducible protein degradation as a novel approach for protein depletion and reverse genetic discoveries in mammalian oocytes†. *Biology of reproduction*, 101(4):704–718, October 2019.

[181] A. L. Mosley, S. G. Pattenden, M. Carey, S. Venkatesh, J. M. Gilmore, L. Florens, J. L. Workman, and M. P. Washburn. Rtr1 is a CTD phosphatase that regulates RNA polymerase II during the transition from serine 5 to serine 2 phosphorylation. *Molecular cell*, 34(2):168–178, April 2009.

[182] E. A. Bowman and W. G. Kelly. RNA polymerase II transcription elongation and Pol II CTD Ser2 phosphorylation: A tail of two kinases. *Nucleus (Austin, Tex.)*, 5(3):224–236, 2014.

[183] Y. Kodama, M. Shumway, R. Leinonen, and I. N. S. D. Collaboration. The Sequence Read Archive: explosive growth of sequencing data. *Nucleic acids research*, 40(Database number):D54–D56, January 2012.

[184] Gene Ontology Consortium. The Gene Ontology project in 2008. *Nucleic acids research*, 36(Database number):D440–D444, January 2008.

[185] E. Eden, R. Navon, I. Steinfeld, D. Lipson, and Z. Yakhini. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC bioinformatics*, 10:48, February 2009.

[186] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, October 2005.

[187] J. F. Tien, A. Mazloomian, S.-W. G. Cheng, C. S. Hughes, C. C. T. Chow, L. T. Canapi, A. Oloumi, G. Trigo-Gonzalez, A. Bashashati, J. Xu, V. C.-D. Chang, S. P. Shah, S. Aparicio, and G. B. Morin. CDK12 regulates alternative last exon mRNA splicing and promotes breast cancer cell invasion. *Nucleic acids research*, 45(11):6698–6716, June 2017.

[188] J. M. Bland and D. G. Altman. Multiple significance tests: the Bonferroni method. *BMJ (Clinical research ed.)*, 310(6973):170, January 1995.

[189] F. Hahne and R. Ivanek. Visualizing Genomic Data Using Gviz and Bioconductor. *Methods in molecular biology (Clifton, N.J.)*, 1418:335–351, 2016.

[190] T.-M. Decker, M. Kluge, S. Krebs, N. Shah, H. Blum, C. C. Friedel, and D. Eick. Transcriptome analysis of dominant-negative Brd4 mutants identifies Brd4-specific target genes of small molecule inhibitor JQ1. *Scientific reports*, 7(1):1684, May 2017.

[191] R. Tejero, Y. Huang, I. Katsyv, M. Kluge, J.-Y. Lin, J. Tome-Garcia, N. Daviaud, Y. Wang, B. Zhang, N. M. Tsankova, C. C. Friedel, H. Zou, and R. H. Friedel. Gene signatures of quiescent glioblastoma cells reveal mesenchymal shift and interactions with niche microenvironment. *EBioMedicine*, 42:252–269, April 2019.

[192] P. Metzger, S. V. Kirchleitner, M. Kluge, L. M. Koenig, C. Hörth, C. A. Rambuscheck, D. Böhmer, J. Ahlfeld, S. Kobold, C. C. Friedel, S. Endres, M. Schnurr, and P. Duewell. Immunostimulatory RNA leads to functional reprogramming of myeloid-derived suppressor cells in pancreatic cancer. *Journal for immunotherapy of cancer*, 7(1):288, November 2019.

[193] X. Zhou, S. Wahane, M.-S. Friedl, M. Kluge, C. C. Friedel, K. Avrampou, V. Zachariou, L. Guo, B. Zhang, X. He, R. H. Friedel, and H. Zou. Microglia and macrophages promote corralling, wound compaction and recovery after spinal cord injury via Plexin-B2. *Nature neuroscience*, 23(3):337–350, March 2020.

[194] E. L. van Dijk, Y. Jaszczyszyn, D. Naquin, and C. Thermes. The Third Revolution in Sequencing Technology. *Trends in genetics : TIG*, 34(9):666–681, September 2018.

[195] S. Oikonomopoulos, A. Bayega, S. Fahiminiya, H. Djambazian, P. Berube, and J. Ragoussis. Methodologies for Transcript Profiling Using Long-Read Technologies. *Frontiers in Genetics*, 11:606, 2020.

[196] X. Tang, Y. Huang, J. Lei, H. Luo, and X. Zhu. The single-cell sequencing: new developments and medical applications. *Cell & bioscience*, 9:53, 2019.

[197] G. Chen, B. Ning, and T. Shi. Single-Cell RNA-Seq Technologies and Related Computational Data Analysis. *Frontiers in Genetics*, 10:317, 2019.

[198] J. L. Garcia-Perez, T. J. Widmann, and I. R. Adams. The impact of transposable elements on mammalian development. *Development (Cambridge, England)*, 143(22):4101–4114, November 2016.

[199] J. Wang, C. Vicente-García, D. Seruggia, E. Moltó, A. Fernandez-Miñán, A. Neto, E. Lee, J. L. Gómez-Skarmeta, L. Montoliu, V. V. Lunyak, and I. K. Jordan. MIR retrotransposon sequences provide insulators to the human genome. *Proceedings of the National Academy of Sciences of the United States of America*, 112(32):E4428–E4437, August 2015.

[200] A. M. Smith, M.-J. Sanchez, G. A. Follows, S. Kinston, I. J. Donaldson, A. R. Green, and B. Göttgens. A novel mode of enhancer evolution: the Tal1 stem cell enhancer recruited a MIR element to specifically boost its activity. *Genome research*, 18(9):1422–1432, September 2008.

[201] D. Jjingo, A. Huda, M. Gundapuneni, L. Mariño-Ramírez, and I. K. Jordan. Effect of the transposable element environment of human genes on gene length and expression. *Genome biology and evolution*, 3:259–271, 2011.

[202] C. Schumacher, H. Wang, C. Honer, W. Ding, J. Koehn, Q. Lawrence, C. M. Coulis, L. L. Wang, D. Ballinger, B. R. Bowen, and S. Wagner. The SCAN domain mediates selective oligomerization. *The Journal of biological chemistry*, 275(22):17173–17179, June 2000.

[203] A. A. Fedotova, A. N. Bonchuk, V. A. Mogila, and P. G. Georgiev. C2H2 Zinc Finger Proteins: The Largest but Poorly Explored Family of Higher Eukaryotic Transcription Factors. *Acta naturae*, 9(2):47–58, 2017.

[204] P. A. Kitts, D. M. Church, F. Thibaud-Nissen, J. Choi, V. Hem, V. Sapojnikov, R. G. Smith, T. Tatusova, C. Xiang, A. Zherikov, M. DiCuccio, T. D. Murphy, K. D. Pruitt, and A. Kimchi. Assembly: a resource for assembled genomes at NCBI. *Nucleic acids research*, 44(D1):D73–D80, January 2016.

[205] O. Fornes, J. A. Castro-Mondragon, A. Khan, R. van der Lee, X. Zhang, P. A. Richmond, B. P. Modi, S. Correard, M. Gheorghe, D. Baranašić, W. Santana-Garcia, G. Tan, J. Chèneby, B. Ballester, F. Parcy, A. Sandelin, B. Lenhard, W. W. Wasserman, and A. Mathelier. JASPAR 2020: update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 48(D1):D87–D92, January 2020.

[206] J. Shi, W. Yang, M. Chen, Y. Du, J. Zhang, and K. Wang. AMD, an automated motif discovery tool using stepwise refinement of gapped consensuses. *PloS one*, 6(9):e24576, 2011.

[207] R. Lu, E. J. Mucaki, and P. K. Rogan. Discovery and validation of information theory-based transcription factor and cofactor binding site motifs. *Nucleic acids research*, 45(5):e27, March 2017.

[208] M. Vahed, J.-I. Ishihara, and H. Takahashi. DIpartite: A tool for detecting bipartite motifs by considering base interdependencies. *PloS one*, 14(8):e0220207, 2019.

[209] F. W. Schmitges, E. Radovani, H. S. Najafabadi, M. Barazandeh, L. F. Campitelli, Y. Yin, A. Jolma, G. Zhong, H. Guo, T. Kanagalingam, W. F. Dai, J. Taipale, A. Emili, J. F. Greenblatt, and T. R. Hughes. Multiparameter functional diversity of human C2H2 zinc finger proteins. *Genome research*, 26(12):1742–1752, December 2016.

[210] M. Imbeault, P.-Y. Helleboid, and D. Trono. KRAB zinc-finger proteins contribute to the evolution of gene regulatory networks. *Nature*, 543(7646):550–554, March 2017.

[211] Y.-M. Wu, M. Cieślik, R. J. Lonigro, P. Vats, M. A. Reimers, X. Cao, Y. Ning, L. Wang, L. P. Kunju, N. de Sarkar, E. I. Heath, J. Chou, F. Y. Feng, P. S. Nelson, J. S. de Bono, W. Zou, B. Montgomery, A. Alva, P. I. P. C. D. Team, D. R. Robinson, and A. M. Chinnaiyan. Inactivation of CDK12 Delineates a Distinct Immunogenic Class of Advanced Prostate Cancer. *Cell*, 173(7):1770–1782.e14, June 2018.

[212] E. Beaudoing, S. Freier, J. R. Wyatt, J. M. Claverie, and D. Gautheret. Patterns of variant polyadenylation signal usage in human genes. *Genome research*, 10(7):1001–1010, July 2000.

[213] A. J. Gruber, R. Schmidt, A. R. Gruber, G. Martin, S. Ghosh, M. Belmadani, W. Keller, and M. Zavolan. A comprehensive analysis of 3' end sequencing data sets reveals novel polyadenylation signals and the repressive role of heterogeneous ribonucleoprotein C on cleavage and polyadenylation. *Genome research*, 26(8):1145–1159, August 2016.

[214] S. J. Dubbury, P. L. Boutz, and P. A. Sharp. CDK12 regulates DNA repair genes by suppressing intronic polyadenylation. *Nature*, 564(7734):141–145, December 2018.

[215] M. Krajewska, R. Dries, A. V. Grassetti, S. Dust, Y. Gao, H. Huang, B. Sharma, D. S. Day, N. Kwiatkowski, M. Pomaville, O. Dodd, E. Chipumuro, T. Zhang, A. L. Greenleaf, G.-C. Yuan, N. S. Gray, R. A. Young, M. Geyer, S. A. Gerber, and R. E. George. CDK12 loss in cancer cells affects DNA damage response genes through premature cleavage and polyadenylation. *Nature communications*, 10(1):1757, April 2019.

[216] T. T. Eifler, W. Shao, K. Bartholomeeusen, K. Fujinaga, S. Jäger, J. R. Johnson, Z. Luo, N. J. Krogan, and B. M. Peterlin. Cyclin-dependent kinase 12 increases 3' end processing of growth factor-induced c-FOS transcripts. *Molecular and cellular biology*, 35(2):468–478, January 2015.

[217] K. Liang, X. Gao, J. M. Gilmore, L. Florens, M. P. Washburn, E. Smith, and A. Shilatifard. Characterization of human cyclin-dependent kinase 12 (CDK12) and CDK13 complexes in C-terminal domain phosphorylation, gene transcription, and RNA processing. *Molecular and cellular biology*, 35(6):928–938, March 2015.

[218] S. H. Choi, T. F. Martinez, S. Kim, C. Donaldson, M. N. Shokhirev, A. Saghatelian, and K. A. Jones. CDK12 phosphorylates 4E-BP1 to enable mTORC1-dependent translation and mitotic genome stability. *Genes & development*, 33(7-8):418–435, April 2019.

[219] B. Bartkowiak, P. Liu, H. P. Phatnani, N. J. Fuda, J. J. Cooper, D. H. Price, K. Adelman, J. T. Lis, and A. L. Greenleaf. CDK12 is a transcription elongation-associated CTD kinase, the metazoan ortholog of yeast Ctk1. *Genes & development*, 24(20):2303–2316, October 2010.

[220] A. L. Greenleaf. Human CDK12 and CDK13, multi-tasking CTD kinases for the new millenium. *Transcription*, 10(2):91–110, April 2019.

[221] Z. Fan, J. R. Devlin, S. J. Hogg, M. A. Doyle, P. F. Harrison, I. Todorovski, L. A. Cluse, D. A. Knight, J. J. Sandow, G. Gregory, A. Fox, T. H. Beilharz, N. Kwiatkowski, N. E. Scott, A. T. Vidakovic, G. P. Kelly, J. Q. Svejstrup, M. Geyer, N. S. Gray, S. J. Vervoort, and R. W. Johnstone. CDK13 cooperates with CDK12 to control global RNA polymerase II processivity. *Science Advances*, 6(18), 2020.

Appendices

Kluge and Friedel *BMC Bioinformatics* (2018) 19:97
https://doi.org/10.1186/s12859-018-2107-4

BMC Bioinformatics

SOFTWARE Open Access

Watchdog – a workflow management system for the distributed analysis of large-scale experimental data

Michael Kluge and Caroline C. Friedel*

Abstract

Background: The development of high-throughput experimental technologies, such as next-generation sequencing, have led to new challenges for handling, analyzing and integrating the resulting large and diverse datasets. Bioinformatical analysis of these data commonly requires a number of mutually dependent steps applied to numerous samples for multiple conditions and replicates. To support these analyses, a number of workflow management systems (WMSs) have been developed to allow automated execution of corresponding analysis workflows. Major advantages of WMSs are the easy reproducibility of results as well as the reusability of workflows or their components.

Results: In this article, we present *Watchdog*, a WMS for the automated analysis of large-scale experimental data. Main features include straightforward processing of replicate data, support for distributed computer systems, customizable error detection and manual intervention into workflow execution. *Watchdog* is implemented in Java and thus platform-independent and allows easy sharing of workflows and corresponding program modules. It provides a graphical user interface (GUI) for workflow construction using pre-defined modules as well as a helper script for creating new module definitions. Execution of workflows is possible using either the GUI or a command-line interface and a web-interface is provided for monitoring the execution status and intervening in case of errors. To illustrate its potentials on a real-life example, a comprehensive workflow and modules for the analysis of RNA-seq experiments were implemented and are provided with the software in addition to simple test examples.

Conclusions: *Watchdog* is a powerful and flexible WMS for the analysis of large-scale high-throughput experiments. We believe it will greatly benefit both users with and without programming skills who want to develop and apply bioinformatical workflows with reasonable overhead. The software, example workflows and a comprehensive documentation are freely available at www.bio.ifi.lmu.de/watchdog.

Keywords: Workflow management system, High-throughput experiments, Large-scale datasets, Automated execution, Distributed analysis, Reusability, Reproducibility, RNA-seq

Background

The development of high-throughput experimental methods, in particular next-generation-sequencing (NGS), now allows large-scale measurements of thousands of properties of biological systems in parallel. For example, modern sequencing platforms now allow simultaneously quantifying the expression of all human protein-coding genes and non-coding RNAs (RNA-seq [1]), active translation of genes (ribosome profiling [2]), transcription factor binding (ChIP-seq [3]), and many more. Dissemination of these technologies combined with decreasing costs resulted in an explosion of large-scale datasets available. For instance, the ENCODE project, an international collaboration that aims to build a comprehensive list of all functional elements in the human genome, currently provides data obtained in more than 7000 experiments with 39 different experimental methods [4]. While such large and diverse datasets still remain the exception, scientific studies now commonly combine two or more

*Correspondence: caroline.friedel@bio.ifi.lmu.de
Institute for Informatics, Ludwig-Maximilians-Universität München, Amalienstraße 17, 80333 München, Germany

high-throughput techniques for several conditions or in time-courses in multiple replicates (e.g. [5–7]).

Analysis of such multi-omics datasets is quite complex and requires a lot of mutually dependent steps. As a consequence, large parts of the analysis often have to be repeated due to modifications of initial analysis steps. Furthermore, errors e.g. due to aborted program runs or improperly set parameters at intermediate steps have consequences for all downstream analyses and thus have to be monitored. Since each analysis consists of a set of smaller tasks (e.g read quality control, mapping against the genome, counting of reads for gene features), it can usually be represented in a structured way as a workflow. Automated execution of such workflows is made possible by workflow management systems (WMSs), which have a number of advantages.

First, a workflow documents the steps performed during the analysis and ensures reproducibility. Second, the analysis can be executed in an unsupervised and parallelized manner for different conditions and replicates. Third, workflows may be reused for similar studies or shared between scientists. Finally, depending on the specific WMS, users with limited programming skills or experience with the particular analysis tools applied within the workflow may more or less easily apply complicated analyses on their own data. On the downside, the use of a WMS usually requires some initial training and some overhead for the definition of workflows. Moreover, the WMS implementation itself might restrict which analyses can be implemented as workflows in the system. Nevertheless, the advantages of WMSs generally outweigh the disadvantages for larger analyses.

In recent years, several WMS have been developed that address different target groups or fields of research or differ in the implemented set of features. The most well-known example, *Galaxy*, was initially developed to enable experimentalists without programming experience to perform genomic data analyses in the web browser [8]. Other commonly used WMSs are *KNIME* [9], an open-source data analysis platform which allows programmers to extend its basic functionality by adding new Java programs, and *Snakemake* [10], a python-based WMS. *Snakemake* allows definition of tasks based on rules and automatically infers dependencies between tasks by matching filenames. A more detailed comparison of these WMSs is given in the Results section.

In this article, we present *Watchdog*, a WMS designed to support bioinformaticians in the analysis of large high-throughput datasets with several conditions and replicates. *Watchdog* offers straightforward processing of replicate data and easy outsourcing of resource-intensive tasks on distributed computer systems. Additionally, *Watchdog* provides a sophisticated error detection system that can be customized by the user and allows manual intervention. Individual analysis tasks are encapsulated within so-called modules that can be easily shared between developers. Although *Watchdog* is implemented in Java, there is no restriction on which programs can be included as modules. In principle, *Watchdog* can be deployed on any operating system.

Furthermore, to reduce the overhead for workflow design, a GUI is provided, which also enables users without programming experience to construct and run workflows using pre-defined modules. As a case study on how *Watchdog* can be applied, modules for read quality checks, read mapping, gene expression quantification and differential gene expression analysis were implemented and a workflow for analyzing differential gene expression in RNA-seq data was created. *Watchdog*, including documentation, implemented modules as well as the RNA-seq analysis workflow and smaller test workflows can be obtained at www.bio.ifi.lmu.de/watchdog.

Implementation

Overview of *Watchdog*

The core features of *Watchdog* and their relationships are outlined in Fig. 1 and briefly described in the following. More details and additional features not mentioned in this overview are described in subsequent sections, Additional files 1, 2 and 3 and in the manual available at www.bio.ifi.lmu.de/watchdog.

Modules

Modules encapsulate re-usable components that perform individual tasks, e.g. mapping of RNA-seq data, counting reads for gene features or visualizing results of downstream analyses. Each module is declared in an XSD file containing the command to execute and the names and valid ranges of parameters. In addition to the XSD file, a module can contain scripts or compiled binaries required by the module and a test script running on example data. Module developers are completely flexible in the implementation of individual modules. They can use the programming language of their choice, include binaries with their modules or automatically deploy required software using Conda (https://conda.io/), Docker (https://www.docker.com/, an example module using a Docker image for Bowtie 2 [11] is included with *Watchdog*) or similar tools. Furthermore, *Watchdog* provides a helper bash script to generate the XSD definition file for new modules and (if required) a skeleton bash script that only needs to be extended by the program call.

Essentially, any program that can be run from the command-line can be used in a module and several program calls can be combined in the same module using e.g. an additional bash script. In principle, a module could

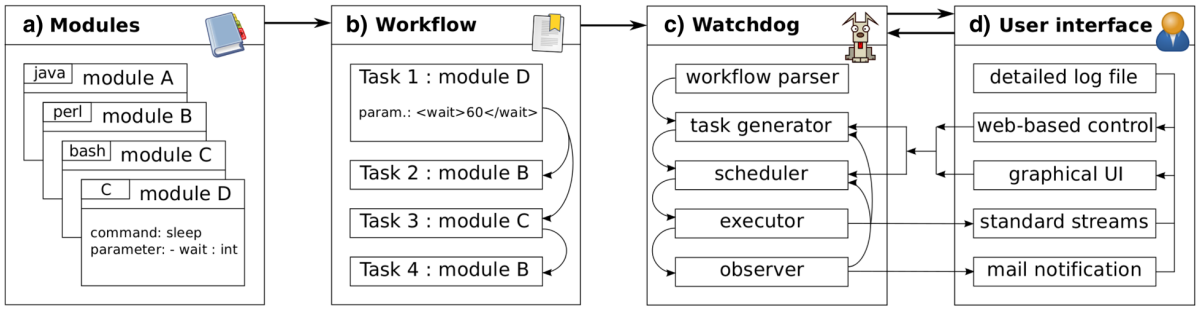


Fig. 1 Overview of *Watchdog*. **a** Modules are defined in an XSD format that describes the command to be executed and valid parameters. All modules together represent the software library that can be used in workflows and can be extended by defining new modules. **b** A workflow is defined in an XML format and consists of tasks that depend on each other. Among others, the XML format allows setting environment variables, defining different executors in the *settings* part of the workflow and processing replicate data in a straightforward way. **c** *Watchdog* parses the workflow, creates the corresponding tasks, executes them and verifies whether execution of each task terminated successfully or not. **d** Email notification (optional) and log files combined with either the GUI or a simple web-interface allow monitoring the execution of the workflow and intervening if necessary, e.g. by restarting tasks with modified parameters

even contain a whole pipeline, such as Maker-P [12], but this would run counter the purpose of a WMS. Here, it would make more sense to separate the individual steps of the pipeline into different modules and then implement the pipeline as a *Watchdog* workflow. Finally, *Watchdog* is not limited to bioinformatics analyses, but can be also used for workflows from other domains.

Workflows

Workflows are defined in XML and specify a sequence of tasks to be executed, the values of their input parameters and dependencies between them. An example for a simple workflow is given in Fig. 2. Among other features that are described later, it is possible to define constants, environment variables and execution hosts in a dedicated *settings* element at the beginning of the workflow, redirect the standard error and standard output for individual tasks or define how detailed the user is informed on the execution status of tasks.

The advantage of XML is that it is widely used in many contexts. Thus, a large fraction of potential *Watchdog* users should already be familiar with its syntax and only need to learn the *Watchdog* XML schema. Furthermore, numerous XML editors are available, including plugins for the widely used integrated development environment (IDE) *Eclipse* [13], which allow XML syntax checking and document structure highlighting. Finally, a number of software libraries for programmatically loading or writing XML are also available (e.g. Xerces for Java, C++ and Perl (http://xerces.apache.org/), ElementTree in Python).

In addition, *Watchdog* also provides an intuitive GUI (denoted *workflow designer*) that can be used to design a workflow, export the corresponding XML file afterwards and run the workflow in the GUI.

Watchdog

The core element of *Watchdog* that executes the workflow was implemented in Java and therefore is, in principle, platform-independent. Individual modules, however, may depend on the particular platform used. For instance, if a module uses programs only available for particular operating systems (e.g. Linux, macOS, Windows), it can only be used for this particular system.

As a first step, *Watchdog* validates the XML format of the input workflow and parses the XML file. Based on the XML file, an initial set of dependency-free tasks, i.e.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <watchdog watchdogBase="/install/dir/" ...>
3     <settings>
4        <constants>
5           <const name="WAIT_TIME">30s</const>
6        </constants>
7     </settings>
8     <tasks mail="my@mail.net">
9        <sleepTask id="1" name="sleep">
10          <parameter>
11             <wait>${WAIT_TIME}</wait>
12          </parameter>
13       </sleepTask>
14    </tasks>
15 </watchdog>
```

Fig. 2 Simple workflow in XML format. This example shows a simple *Watchdog* workflow executing a 30 second sleep task. A constant named *WAIT_TIME* is defined within the *settings* environment (line 5). Email notification of the user is enabled using the optional *mail* attribute of the *tasks* environment (line 8). Here, a task of type *sleepTask* with *id* 1 and *name* sleep is defined (lines 9-13). Either id or name can be used to refer to this task in dependency declarations of other tasks. Within the *parameter* environment of the *sleepTask*, values are assigned to required parameters (lines 10-12), which were specified in the XSD file of this particular module. In this case, the parameter *wait* is set to the value stored in the constant *WAIT_TIME* (line 11)

tasks that do not depend on any other tasks, is generated and added to the WMS scheduler to execute them. Subsequently, the scheduler continuously identifies tasks for which dependencies have been resolved, i.e. all preceding tasks the task depends on have been executed successfully, and schedules them for execution. Once a task is completed, *Watchdog* verifies that the task finished successfully. In this case, the task generator and scheduler are informed since dependencies of other tasks might have become resolved. In case of an error, the user is informed via email (optional) and the task is added to the scheduler again but is blocked for execution until the user releases the block or modifies its parameters. Alternatively, the user may decide to skip the task or mark the error as resolved.

User interfaces

Watchdog provides both a command-line version as well as a GUI that can be used to execute workflows and to keep track of their processing. Moreover, a web-interface is provided to GUI and command-line users that displays the status of all tasks in a table-based form and allows monitoring and interacting with the execution of tasks by releasing scheduled tasks, changing parameters after a failed task execution and more (see Fig. 3). The link to the web-interface is either printed to standard output or sent to the user by email if they enabled email notification. In the latter case, the user will also be notified per email about execution failure (always) or success (optional). Finally, the command-line interface also allows resuming a workflow at any task or limiting the execution of the workflow to a subset of tasks using the `-start` (start execution at specified task), `-stop` (stop execution after specified task), `-include` (include this task in execution) and `-exclude` (exclude this task for execution) options.

In the following more details are provided on principles and possibilities of workflow design in *Watchdog* and defining custom modules. The GUI is described in detail in Additional file 1.

Process blocks for creating subtasks

Analysis of high-throughput data often requires performing the same analysis steps in parallel for a number of samples representing different conditions or biological or technical replicates. To support these types of analyses, *Watchdog* uses so-called process blocks to automatically process tasks that differ only in values of parameters, e.g. short read alignment for all FASTQ files in a directory. For this purpose, process blocks define a set of instances, each of which contain one or more variables. For each instance, one subtask is created and subtask placeholders in the task definition are replaced with the variable values of the instance. For the example in which a task is executed for all FASTQ-files in a directory, each instance holds one variable containing the absolute file path of the file. The number of subtasks corresponds to the number of FASTQ-files in the directory.

Currently four different types of process blocks are supported by *Watchdog*: process sequences, process folders, process tables and process input (Fig. 4). In case of process sequences (Fig. 4a) and process folders (Fig. 4b), instances only hold a single variable. Process sequences are comparable to for-loops as they generate instances containing numerical values (integer or floating-point numbers) with a fixed difference between two consecutive numbers (default: 1). Instances generated by process folders contain the absolute path to files and are generated based on a parent folder and a filename pattern.

Process tables (Fig. 4c) and process input (Fig. 4d) blocks can generate instances with multiple variables. Instances generated by a process table are based on the content of a tab-separated file. The rows of the table define individual instances and the columns the variables for each instance. In case of process input blocks, variables and instances

ID	Name	Block argument	Executor	Hostname	Exec. counter	Status	Action
1-1	compress	/tmp/d.log	local	colibri	1	has finished	display parameters GO!
1-2	compress	/tmp/e.log	local	colibri	1	has failed	modify parameters GO!
1-3	compress	/tmp/b.log	local	colibri	1	is currently running	GO!
1-4	compress	/tmp/c.log	local	colibri	1	is currently running	GO!
1-5	compress	/tmp/g.log	local		1	is waiting for resource restrictions	release resource restrictions GO!

display parameters
modify parameters
restart task
ignore task
mark as resolved

Fig. 3 Web-interface of *Watchdog*. Each line of the table provides information on the status of a task or subtask. The drop-down menu at the end of each line allows to perform specific actions depending on the status of the task. The menu is shown for subtask *1-2*, which could not be executed successfully. To generate this screenshot the example workflow depicted in Fig. 6 was processed, which compresses all log-files stored in directory */tmp/*. Since the number of simultaneously running subtasks was set to at most 2 for this task, subtask *1-5* is put on hold until subtasks *1-3* and *1-4* have finished or the user manually releases the resource restriction

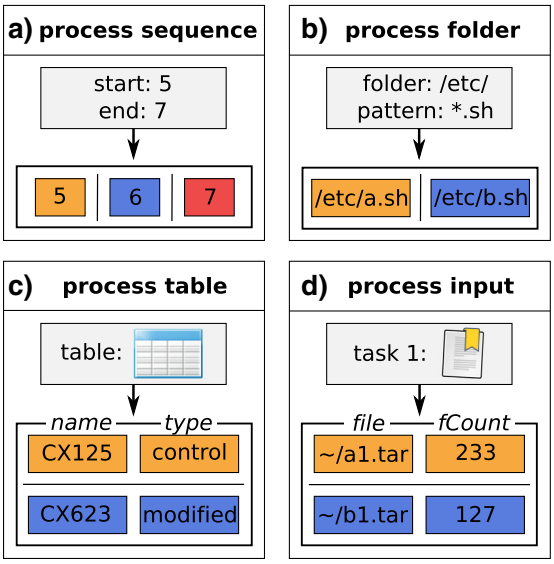


Fig. 4 Types of process blocks. With the help of process blocks, multiple tasks that differ only in the parameter values can be created without defining all of them separately. Four different types of process blocks are implemented that fall into two general classes. Instances of the first class contain only a single variable, either (**a**) a value from a numerical sequence (process sequence) or (**b**) a path to files (process folder). In (**a**), subtasks are created based on an integer sequence starting at 5 and ending at 7 with an increment of 1. In (**b**), a subtask is created for each sh-file in the folder */etc/*. Instances of the second type can contain multiple variables, either (**c**) instances derived from tables (process table) or (**d**) instances based on return values returned by previous tasks this task depends on (process input). In (**c**), a table with two columns named *name* and *type* and two rows is used as input for the process table. This results in two subtasks for this task, one for each row. The process input block in (**d**) depends on a task with id 1, which itself had two subtasks. Hence, this task returns two instances, each containing the variables *file* and *fCount* obtained from its return variables

```
1  <settings>
2     <processBlock>
3        <processSequence name="num" start="1"
              end="9" step="4" />
4        <processFolder name="logFiles"
              folder="/tmp/" pattern="*.log" />
5     </processBlock>
6  </settings>
```

Fig. 5 Definition of process blocks. In this example, two process blocks are defined within the *processBlock* environment (lines 2-5). In line 3, a process sequence named *num* is defined consisting of three instances (1, 5 and 9). In line 4, a process folder selecting all log-files in the */tmp/* directory is defined

are derived from return values of preceding tasks the task depends on.

Figure 5 shows an example how process blocks can be defined and Fig. 6 shows how they can be used for creation of subtasks. In Additional file 2, a detailed description with examples is provided on how to use process blocks for the analysis of data sets with several replicates or conditions. Furthermore, *Watchdog* provides a plugin system that allows users with programming skills to implement novel types of process blocks without having to change the original *Watchdog* code (see Additional file 3).

Dependencies

By default, all tasks specified in a *Watchdog* workflow are independent of each other and are executed in a non-deterministic order. Alternatively, dependencies on either task or subtask level (details in the next paragraphs) can be defined using the *id* or *name* attribute of a task (see Fig. 7). Dependency definitions impose a partial order on tasks, meaning that tasks depending on other tasks will only be executed after those other tasks have finished successfully. Tasks without dependencies or resolved dependencies will still be executed in a non-deterministic order.

Although explicit dependency definition adds a small manual overhead compared to automatic identification based on in- and output filenames as in *Snakemake*, it also provides more flexibility as dependencies can be defined that are not obvious from filenames. For instance, analysis of sequencing data usually involves quality control of sequencing reads, e.g. with FastQC [14], before mapping of reads, and users might want to investigate the results of quality control before proceeding to read mapping. However, output files of quality control are not an input to read mapping and thus this dependency could not be identified automatically. To provide more time to manually validate results of some intermediate steps, *Watchdog* allows adding checkpoints after individual tasks. After completion of a task with checkpoint, all dependent tasks are put on hold until the checkpoint is released. All checkpoints in

```
1  <gzipTask id="2" name="compress"
        processBlock="logFiles" maxRunning="2">
2     <parameter>
3        <input>{}</input>
4        <output>/tmp/[1]_compressed.gz</output>
5        <quality>7</quality>
6     </parameter>
7  </gzipTask>
```

Fig. 6 Usage of process blocks. The process block *logFiles* defined in Fig. 5 is used to generate several subtasks (line 1). These subtasks create compressed versions of the log-files stored in */tmp/*. In this case, at most two subtasks are allowed to run simultaneously. Additional file 2 describes how process block variables can be accessed. Here, the placeholder {} is replaced by the variable values stored in the process block, i.e. the complete file paths, and [1] is replaced with the file names (without the '.log' file-ending) (lines 3-4)

```
1  <task id="3" processBlock="logFiles">
2     <dependencies>
3        <depends>sleep</depends>
4        <depends separate="true">2</depends>
5     </dependencies>
6     ...
7  </task>
```

Fig. 7 Definition of dependencies. The task defined in this example creates subtasks using the process block *logFiles* from Fig. 5 (line 1) with both task and subtask dependencies. A task dependency on the task *sleep* defined in Fig. 2 is indicated in line 3. In addition, subtask dependencies to the task with id 2 defined in Fig. 6 are indicated in line 4. In this case, each subtask depends on the subtask of task 2 which was created using the same instance defined by the process block *logFiles*, i.e. the same file path

a workflow can be deactivated upon workflow execution with the `-disableCheckpoint` flag of the *Watchdog* command-line version.

Task dependencies

A task B can depend on one or more other tasks A_1 to A_n, which means that execution of task B is put on hold until tasks A_1 to A_n have finished successfully. If some of the dependencies A_1 to A_n use process blocks to create subtasks, task B is put on hold until all subtasks are finished successfully. Figure 8a illustrates the described behavior on a small example in which task B depends on three other tasks.

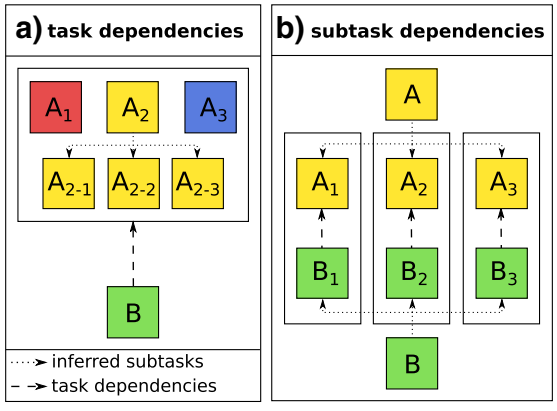


Fig. 8 Types of dependencies. Dependencies can either be defined on (**a**) task or (**b**) subtask level. **a** Task B depends on tasks A_1, A_2 and A_3. Task A_2 uses a process block to create the three subtasks A_{2-1}, A_{2-2} and A_{2-3}. Task B will be executed when A_1, A_2 (including all subtasks) and A_3 have finished successfully. **b** Tasks A and B create subtasks using a process block. For example, task A might decompress files stored in a folder (by using a process folder) and task B might extract data from the decompressed files afterwards (by using a process input block). Here, subtask B_x of B only depends on the subtask A_x of A based on whose return values it is created

Subtask dependencies

If a subtask B_x of a task B only depends on a particular subtask A_x of A instead of all subtasks of A, the definition of subtask dependencies in the workflow allows executing B_x as soon as A_x has finished successfully (but not necessarily other subtasks of A). This is illustrated in Fig. 8b and can be explained easily for the most simple case when the process block used for task B is a process input block containing the return values of subtasks of A. In that case, a subtask B_x depends only on the subtask A_x of A that returned the instance resulting in the creation of B_x. The use of subtask dependencies is particularly helpful if subtasks of A need different amounts of time to finish or cannot all be executed at the same time due to resource restrictions, such as a limited amount of CPUs or memory available. In this case, B_x can be executed as soon as A_x has finished but before all other subtasks of A have finished. An example application would be the conversion of SAM files resulting from read mapping (task A) to BAM files (task B).

Parallel and distributed task execution

By default all tasks are executed one after the other on the host running *Watchdog* (see Fig. 9a,b). In principle, however, tasks that are independent of each other or individual subtasks of a task can be executed in parallel. *Watchdog* implements three different types of executors that facilitate parallel execution of tasks: (i) local executor (Fig. 9c), (ii) remote executor (Fig. 9d) and (iii) cluster executor (Fig. 9e). All executors allow multi-threaded execution of tasks. In cases (i) and (ii) *Watchdog* uses multiple threads for parallel execution of tasks while in case (iii) the cluster master is utilized to distribute tasks on the cluster. Before execution or after completion or failure of tasks, files or directories can be created, deleted or copied to/from remote file systems (e.g. the file system of a remote or cluster executor) using so-called task actions. By default, *Watchdog* supports virtual file systems based on the protocols File, HTTP, HTTPS, FTP, FTPS and SFTP as well as the main memory (RAM). However, any file system with an implementation of the FileProvider interface from the Commons Virtual File System project of the Apache Software Foundation (http://commons.apache.org/proper/commons-vfs/) can also be used (see manual).

Executors and their resource limitations are declared in the *settings* element at the beginning of the workflow (see Fig. 10) and assigned to tasks based on their names. Within each workflow, an arbitrary number of executors of different types can be defined and any of these can be assigned to individual tasks. For instance, memory-intensive tasks might be executed on a dedicated high-memory computer using a remote executor while other tasks spawning many subtasks are distributed using

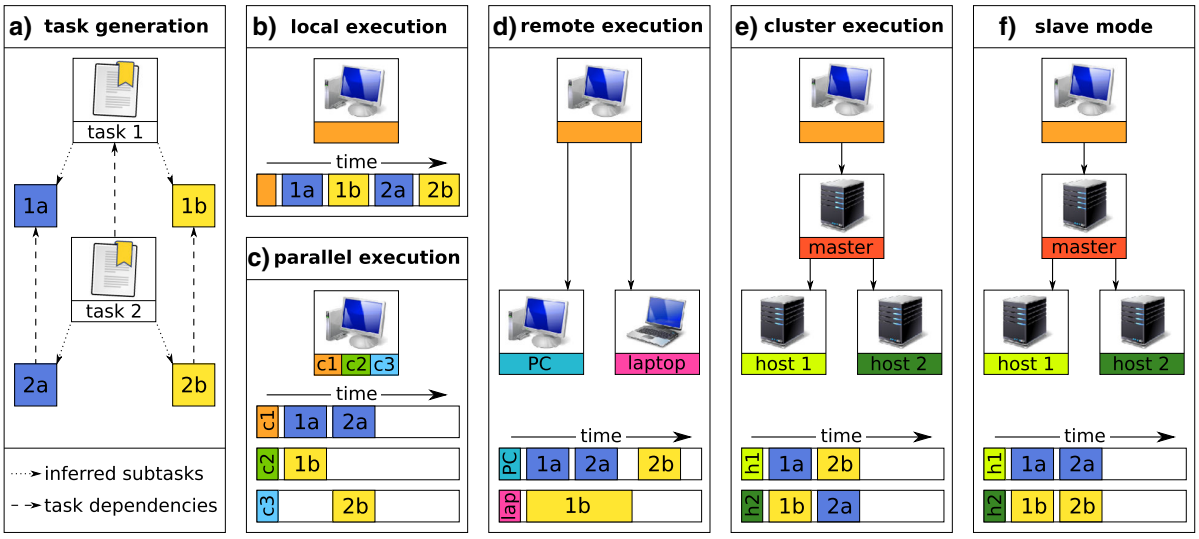


Fig. 9 Parallel and distributed task execution. Three different types of executors are implemented in *Watchdog*: (i) execution on the local host that runs *Watchdog*, (ii) remote execution via SSH or (iii) cluster execution using DRMAA or the Slurm Workload Manager. **a** In this example, the four subtasks 1a, 1b, 2a and 2b are created by *Watchdog* based on tasks 1 and 2 using process blocks. Task 2a depends on 1a, and 2b on 1b. All tasks are assumed to require the same runtime. **b** By default, one task is executed after the other on the host running *Watchdog*. **c** *Watchdog* also allows parallel execution in all three execution modes (local, remote and cluster execution). **d** For remote execution, *Watchdog* establishes a SSH connection to pre-defined execution hosts and randomly distributes the tasks that should be executed to these execution hosts. **e** For cluster execution, the DRMAA or Slurm master receives tasks to execute and redirects them to its execution hosts. *Watchdog* has no influence on which execution host is used for task execution because the tasks are distributed by the internal DRMAA or Slurm scheduler. **f** During slave mode (supported for remote and cluster execution), tasks or subtasks that depend on each other are scheduled on the same execution host, which allows using the local disk space of the host for storage of files that are needed only temporarily but by different tasks

a cluster executor and non-resource-intensive tasks are run using a local executor. Here, the number of simultaneously running (sub)tasks can be restricted on task (see Fig. 6) or executor level (see Fig. 10), e.g. to not occupy the whole cluster with many long-running tasks. Provided the name of a particular executor remains the same, everything else can be modified about this executor without having to change the *tasks* part of the workflow. This includes not only resource limitations or the maximum number of running tasks but even the type of executor, for instance when moving the workflow to a different system.

```
1  <settings>
2     <executors>
3        <local name="localhost"
              maxRunning="2" default="true"/>
4        <remote name="highP" host="goliath"
              user="me" privateKey="~/key.sec"/>
5        <cluster name="cluster" memory="1G"
              queue="short" maxRunning="16"/>
6     </executors>
7  </settings>
```

Fig. 10 Defining executors. This example defines three possible executors: (i) the local host running *Watchdog* using two parallel threads for task execution (line 3). This will be used by default for task execution if no other executor is specified in a task definition using the *executor* attribute. (ii) a remote host named *goliath* accessed by SSH and authenticated via a private key that should be protected by a passphrase (line 4). (iii) a cluster executor that schedules a maximum of 16 simultaneously running tasks on the *short* queue of a computer cluster supporting DRMAA (line 5)

Every host that accepts secure shell connections (SSH) can be used as a remote executor (see Fig. 9d). In this case, a passphrase-protected private key for user authentication must be provided. For cluster execution, any grid computing infrastructures that implement the Distributed Resource Management Application API (DRMAA) can be utilized (see Fig. 9e). By default, *Watchdog* uses the Sun Grid Engine (SGE) but other systems that provide a DRMAA Java binding can also be used. Furthermore, *Watchdog* provides a plugin system that allows users with programming skills to add new executor types without having to change the original *Watchdog* code. This plugin system is explained in detail in Additional file 3 and was used to additionally implement an executor for computing clusters or supercomputers running the Slurm Workload Manager (https://slurm.schedmd.com/). The plugin system can also be used to provide support for cloud computing services that do not allow SSH. Support for the Message Passing Interface (MPI) is not explicitly modeled

in *Watchdog*, but MPI can be used by individual modules if it is supported by the selected executor.

Finally, to allow storage of potentially large temporary files on the local hard disk of cluster execution hosts and sharing of these files between tasks, *Watchdog* also implements a so-called *slave mode* (see Fig. 9f). In slave mode, the scheduler ensures that tasks or subtasks depending on each other are processed on the same host allowing them to share temporary files on the local file system. For this purpose, a new slave is first started on an execution host, which establishes a network connection to the master (i.e. the host running *Watchdog*) and then receives tasks from the master for processing.

Error detection and handling

During execution of workflows, a number of errors can occur resulting either in aborted program runs or incorrect output. To identify such errors, *Watchdog* implements a sophisticated error checking system that allows flexible extension by the user. For this purpose, *Watchdog* first checks the exit code of the executed module. By definition an exit code of zero indicates that the called command was executed successfully. However, some tools return zero as exit code regardless of whether the command succeeded or failed. Thus, the exit code alone is not a reliable indicator whether the command was executed successfully. Furthermore, a command can technically succeed without the desired result being obtained. For instance, the mapping rate for RNA-seq data may be very low due to wrong parameter choices or low quality of reads. To handle such cases, the user has the option to implement custom success and error checkers in Java that are executed by *Watchdog* after a task is finished. Two steps must be performed to use custom checkers: implementation in Java and invocation in the XML workflow (see Fig. 11 for an example and the manual for details).

Once the task is finished, the checkers are evaluated in the same order as they were added in the XML workflow. In cases in which both success and error were detected by different checkers, the task will be treated as failed. When an error is detected, the user is informed via email notification (if enabled, otherwise the information is printed to standard output), including the name of the execution host, the executed command, the returned exit code and the detected errors.

```
1  <task name="error check" ...>
2     <checkers>
3        <checker classPath="/home/CErr.class"
              className="example" type="error">
4           <cArg type="string">sleep</cArg>
5           <cArg type="integer">1</cArg>
6        </checker>
7     </checkers>
8     ...
9  </task>
```

Fig. 11 Invocation of a custom error checker. The example illustrates how a custom error checker implemented in class *CErr* located in directory */home/* can be added to a task (line 3). In line 4 and 5, two arguments of type *string* and *integer* are forwarded to the constructor of the error checker

Information on failure or success is also available via the web-interface, which then allows to perform several actions: (i) modify the parameter values for the task and restart it, (ii) simply restart the task, (iii) ignore the failure of the task or (iv) manually mark the task as successfully resolved. In case of (iii), (sub)tasks that depend on that task will not be executed, but other (sub)tasks will continue to be scheduled and executed. To continue with the processing of tasks depending on the failed task, option (iv) can be used. In this case, values of return parameters of the failed task can be entered manually via the web-interface.

Option (i) is useful if a task was executed with inappropriate parameter values and avoids having to restart the workflow at this point and potentially repeating tasks that are defined later in the workflow but are not dependent on the failed task. As *Watchdog* aims to execute all tasks without (unresolved) dependencies as soon as executors and resource limitations allow, these other tasks might already be running or even be finished. Option (ii) is helpful if a (sub)task fails due to some temporary technical problem in the system, a bug in a program used in the corresponding module or missing software. The user can then restart the (sub)task as soon as the technical problem or the bug is resolved or the software has been installed without having to restart the other successfully finished or still running (sub)tasks. Here, the XSD definition of a module cannot be changed during a workflow run as XSD files are loaded at the beginning of workflow execution, but the underlying program itself can be modified as long as the way it is called remains the same. Option (iii) allows to finish an analysis for most samples of a larger set even if individual samples could not be successfully processed, e.g. due to corrupt data. Finally, option (iv) is useful if custom error checkers detect a problem with the results, but the user nevertheless wants to finish the analysis.

Defining custom modules

Watchdog is shipped with 20 predefined modules, but the central idea of the module concept is that every developer can define their own modules, use them in connection with *Watchdog* or share them with other users. Each module consists of a folder containing the XSD module definition file and optional scripts, binaries and test scripts. It should be noted here that while the complete encapsulation of tasks within modules is advantageous for larger tasks consisting of several steps or including additional checks on in- or output, the required module

creation adds some burden if only a quick command is to be executed, such as a file conversion or creation of a simple plot. However, to reduce the resulting overhead for module creation, a helper bash script is available for unix-based systems that interactively leads the developer through the creation of the XSD definition file.

For this purpose, the script asks which parameters and flags to add. In addition, optional return parameters can be specified that are required if the module should be used as process input block. If the command should not be called directly because additional functions (e.g. checks for existence of input and output files and availability of programs) should be executed before or after the invocation of the command, the helper script can generate a skeleton bash script that has to be only edited by the developer to include the program and additional function calls. Please note that modules shipped with *Watchdog* were created with the helper script, thus XSD files and large fractions of bash scripts were created automatically with relatively little manual overhead. Once the XSD file for a module is created, the module can be used in a workflow. By default, *Watchdog* assumes that modules are located in a directory named *modules/* in the installation directory of *Watchdog*. However, the user can define additional module folders at the beginning of the workflow.

Results and discussion

Example workflows

For testing and getting to know the potentials of *Watchdog* by first-time users, two longer example workflows are provided with the software, which are documented extensively within the XML file (contained in the *examples* sub-directory of the *Watchdog* installation directory after configuring the examples, see manual for details). All example workflows can also be loaded into the GUI in order to get familiar with its usage (see Additional file 1). In order to provide workflows that can be used for practically relevant problems, 20 modules were developed that are shipped together with *Watchdog*. In addition, several smaller example workflows are provided, each demonstrating one particular feature of *Watchdog*. They are explained in detail in the manual. The next paragraphs describe the two longer example workflows and the corresponding test dataset.

Test dataset

A small test dataset consisting of RNA-seq reads is included in the *Watchdog examples* directory. It is a subset of a recently published time-series dataset on HSV-1 lytic infection of a human cell line [5]. For this purpose, reads mapping to chromosome 21 were extracted for both an uninfected sample and a sample obtained after eight hours of infection. Both samples in total contain about 308,000 reads.

Workflow 1 - Basic information extraction

This workflow represents a simple example for testing *Watchdog* and uses modules encapsulating the programs *gzip*, *grep* and *join*, which are usually installed on unix-based systems by default. Processing of the workflow requires about 50MB of storage and less than one minute on a modern desktop computer. As a first step, gzipped FASTQ files are decompressed. Afterwards, read headers and read sequences are extracted into separate files. To demonstrate the ability of *Watchdog* to restrict the number of simultaneously running jobs, the sequence extraction tasks are limited to one simultaneous run, while the header extraction tasks are run in parallel (at most 4 simultaneously). Once the extraction tasks are finished, the resulting files from each sample are compressed and merged.

Workflow 2 - Differential gene expression

This workflow illustrates *Watchdog*'s potentials for running a more complex and practically relevant analysis. It implements a workflow for differential gene expression analysis of RNA-seq data and uses a number of external software programs for this purpose. Thus, although XSD files for corresponding modules are provided by *Watchdog*, the underlying software tools have to be installed and paths to binaries added to the environment before running this workflow. The individual modules contain dependency checks for the required software that will trigger an error if some of them are missing.

Software required by modules used in the workflow include *FastQC* [14], *ContextMap 2* [15], *BWA* [16], *samtools* [17], *featureCounts* [18], *RSeQC* [19], *R* [20], *DEseq* [21], *DEseq2* [22], *limma* [23], and *edgeR* [24]. The workflow can be restricted to just the initial analysis steps using the `-start` and `-stop` options of the *Watchdog* command-line version and individual analyses steps can be in- or excluded using the `-include` and `-exclude` options. Thus, parts of this workflow can be tested without having to install all programs. Please also note that the workflow was tested on Linux and may not immediately work on macOS due to differences in pre-installed software. Before executing the workflow a few constants have to be set, which are marked as *TODO* in the comments of the XML file. Processing of the workflow requires about 300MB of storage and a few minutes on a modern desktop computer.

The first step is again decompression of gzipped FASTQ files. Afterwards, quality assessment is performed for each replicate using *FastQC*, which generates various quality reports for raw sequencing data. Subsequently, the reads are mapped to chromosome 21 of the human genome using *ContextMap 2*. After read mapping is completed, the resulting SAM files are converted to BAM files and BAM files are indexed using modules based on *samtools*.

Afterwards, reads are summarized to read counts per gene using *featureCounts*. As methods for differential gene expression detection may require replicates, pseudo-replicates are generated by running *featureCounts* twice with different parameters. This was done in order to provide a simple example that can be executed as fast as possible and should not be applied when real data is analyzed. In parallel, quality reports on the read mapping results are generated using *RSeQC*. Finally, *limma*, *edgeR*, *DEseq* and *DEseq2* are applied on the gene count table in order to detect differentially expressed genes. All four programs are run as part of one module, *DETest*, which also combines result tables of the different methods. Several of the provided modules also generate figures using *R*.

Comparison with other WMSs

Most WMSs can be grouped into two types based on how much programming skills are required in order to create a workflow. If a well-engineered GUI or web interface is provided, users with basic computer skills should be able to create their own workflows. However, GUIs can also restrict the user as some features may not be accessible. Hence, a second group of WMSs addresses users with more advanced programming skills and knowledge of WMS-specific programming or scripting languages.

As a comprehensive comparison of all available WMS is outside the scope of this article, two commonly used representatives of each group were selected and compared with *Watchdog*. Figure 12 lists features of each WMS, which are grouped into the categories *setup*, *workflow design*, *workflow execution* and *integration* of new tools. As representative WMSs *Galaxy* [8], *KNIME* [9], *Snakemake* [10], and *Nextflow* [25] were chosen. In the following paragraphs, the selected WMSs are discussed. Because all four WMSs as well as *Watchdog* allow non-programmers to execute predefined workflows, this property is not further discussed. Furthermore, an analysis of the computational overhead of *Watchdog* and *Snakemake* showed that the computational overhead of using either WMS (and likely any other) is negligible compared to the actual runtime of the executed tasks (see Additional file 4).

Galaxy

The most well-known WMS for bioinformatic analyses is *Galaxy* [8]. It was initially developed to enable experimentalists without programming experience to perform genomic data analyses in the web browser. Users can upload their own data to a *Galaxy* server, select and combine available analysis tools from a menu and configure them using web forms. To automatically perform the same workflow on several samples in a larger data set, so-called collections can be used.

In addition to computer resources, *Galaxy* provides a web-platform for sharing tools, datasets and complete workflows. Moreover, users can set up private *Galaxy* servers. In order to integrate a new tool, an XML-file has to be created that specifies the input and output parameters. Optionally, test cases and the expected output of a test case can be defined. Once the XML-file has been prepared, *Galaxy* must be made aware of the new tool and be re-started. If public *Galaxy* servers should be used, all input data must be uploaded to the public *Galaxy* servers. This is especially problematic for users with only low-bandwidth internet access who want to analyze large high-throughput datasets but cannot set up their own server.

In summary, *Galaxy* is a good choice for users with little programming experience who want to analyze data using a comfortable GUI, might not have access to enough computer resources for analysis of large high-throughput data otherwise, appreciate the availability of a lot of predefined tools and workflows and do not mind the manual overhead.

KNIME

The Konstanz Information Miner, abbreviated as *KNIME* [9], is an open-source data analysis platform implemented in Java and based on the IDE *Eclipse* [13]. It allows programmers to extend its basic functionality by adding so-called nodes. In order to create a new node, at least three interfaces must be implemented in Java: (i) a model class that contains the data structure of the node and provides its functionality, (ii) view classes that visualize the results once the node was executed and (iii) a dialog class used to visualize the parameters of the node and to allow the user to change them.

One disadvantage for node developers is that the design of the dialog is labor-intensive, in particular for nodes that accept a lot of parameters. Another shortcoming of *KNIME* is that only Java code can be executed using the built-in functionality. Hence, wrapper classes have to be implemented in Java if a node requires external binaries or scripts. Furthermore, *KNIME* does not support distributed execution in its free version. However, two extensions can be bought that allow either workflow execution on the SGE or on a dedicated server.

Hence, the free version of *KNIME* is not suitable for the analysis of large high-throughput data. However, *KNIME* can be used by people without programming skills for the analysis of smaller datasets using predefined nodes, especially, if a GUI is required that can be used to interactively inspect and visualize the results of the analysis.

Snakemake

A workflow processed by *Snakemake* [10] is defined as a set of rules. These rules must be specified in *Snakemake*'s

		Watchdog	*Galaxy*	*KNIME*	*Snakemake*	*Nextflow*
setup	documentation	yes	yes	yes	yes	yes
	open source	yes	yes	yes[1]	yes	yes
	dependencies	Java	python2	Java	python3	Java/POSIX
	installation	unpack	install script	unpack	Conda, pip	install script
workf. design	GUI for workflow design	Java	web-based	Java	no	no
	dependency definition	id/graphical	graphical	graphical	filenames[2]	channel names
	built-in replicate detection[3]	process blocks	collections[4]	list file node	wildcards	channels
	return variables at runtime[5]	yes	limited[6]	yes	no	yes
	used syntax/semantic	XML/own	-	-	python/own	own/own
workflow execution	GUI for workflow monitoring	GUI, web-based	web-based	GUI	web-based	no
	parallel replicate execution	yes	yes	beta-status	yes	yes
	distributed execution	DRMAA, SSH, Slurm[7]	DRMAA, other[7]	no[8]	DRMAA, generic[7]	DRMAA, other[7]
	cloud computing support	manual (SSH, plugins[9])	yes	Azure, AWS	Kubernetes	AWS, Kubernetes
	remote storage support	yes[10]	yes[10]	yes[10]	yes[10]	yes[10]
	built-in error detection	exit code, checker	exit code[11]	no[12]	exit code	exit code
	local storage usage[13]	yes	no	no	no	yes
	user notification	GUI/console, email	GUI, email	GUI	console	console
	manual intervention on errors	yes	yes	yes	no	no
	resume previous execution	manual	manual	man., autom.	man., autom.	man., autom.
integration	supported languages	no restrictions	no restrictions	Java	no restrictions[14]	no restrictions[15]
	implementation overhead	small	small	high	small	small
	encapsulation[16]	module	tool	node	wrapper	-
	automatic software deployment	within module[17]	Tool Shed/Conda	within node	Conda, Singularity, Docker	Singularity, Docker
	sharing of integrated tools	copy module folder	sharing platform	export function	wrapper repository	copy part of workf.
	used syntax/semantic	XML/own	XML/own	Java/Java	python/own	own/own

Fig. 12 Comparison of *Watchdog* with other WMSs. Comparison was performed using features grouped into the categories setup, workflow design, workflow execution and integration. Workflow is abbreviated as *workf.* in this table. Integration refers to the integration of new data analysis tools into the particular WMS. Footnotes: [1]six non-free extensions are available; [2]since version 2.4.8, rules can also explicitly refer to the output of other rules; [3]explanation: includes a way to automatically run a predefined workflow for a variable number of replicates based on filename patterns; [4]have to be created manually in the web-interface from uploaded files; [5]explanation: finished steps of the workflow can return variables that are used by subsequent steps as input; [6]can only return the names of output files; [7]other supported executors: *Watchdog*: new executors can be added with the plugin system, *Galaxy*: PBS/Torque, Open Grid Engine, Univa Grid Engine, Platform LSF, HTCondor, Slurm, Galaxy Pulsar, *Snakemake*: can also use cluster engines with access to a common file system and a submit command that accepts shell scripts as first argument, *Nextflow*: SGE, LSF, Slurm, PBS/Torque, NQSII, HTCondor, Ignite; [8]non-free extensions for SGE or dedicated server support are available; [9]custom executors for cloud computing services can be created using the plugin system; [10]*Watchdog*: HTTP/S, FTP/S and SFTP by default, can be extended to any remote file system with an implementation of the FileProvider interface from the Commons Virtual File System project, *Galaxy*: Object Store plugins for S3, Azure, iRODS, *Snakemake*: S3, GS, SFTP, HTTP, FTP, Dropbox, XRootD, NCBI, WebDAV, GFAL, GridFTP. *Nextflow*: HTTP/S, FTP, S3; [11]a hard-coded error checker triggered on keywords 'exception' and 'error' in standard output and error is provided; [12]depends on the node implementation and left to developer; [13]explanation: usage of local storage during distributed execution in order to avoid unnecessary load on the shared storage system; [14]direct integration of python code is possible; [15]own scripting language available; [16]explanation: describes the concept used to separate workflow definition and functionality (e.g. *Watchdog*'s modules) in order to allow easy re-use of functionality; [17]modules can include binaries in the module directory or automatically deploy required software using Conda, Singularity, Docker or similar tools available on the used system

own language in a text file named *Snakefile*. Similar to *GNU Make*, which was developed to resolve complex dependencies between source files, each rule describes how output files can be generated from input files using shell commands, external scripts or native python code. At the beginning of workflow execution, *Snakemake* automatically infers the rule execution order and dependencies based on the names of the input and output files for each rule. From version 2.4.8 on, dependencies can also be declared by explicitly referring to the output of rules defined further above. Workflows can be applied automatically to a variable number of samples using wildcards, i.e. filename patterns on present files.

In *Snakemake*, there is no clear separation between the tool library and workflow definition as the command used to generate output files is defined in the rule definition itself. Starting with version 3.5.5, *Snakemake* introduced re-usable wrapper scripts e.g. around command-line tools. In addition, it provides the possibility to include either individual rules or complete workflows as sub-workflows. Thus, *Snakemake* now allows both encapsulation of integrated tools as well as quickly adding commands directly into the workflow.

By default, no new jobs are scheduled in *Snakemake* as soon as one error is detected based on the exit code of the executed command. Accordingly, the processing of the

complete workflow is halted until the user fixes the problem. This is of particular disadvantage if time-consuming tasks are applied on many replicates in parallel and one error for one replicate prevents execution of tasks for other replicates. While this default mode can be overridden by the `-keep-going` flag, this flag has to be set when starting execution of the workflow and applies globally independent of which particular parts of the workflow caused the error. In addition, the option `-restart-times` allows automatically restarting jobs after failure for a predefined number of times and each rule can specify how resource constraints are adapted in case of restarts. However, this option is only useful in case of random failure or failure due to insufficient resources. If errors result from incorrect program calls or inappropriate parameter values, restarting the task will only result in the same error again. Finally, *Snakemake* is the only one of the compared WMSs that does not provide return variables that can be used as parameters in later steps.

In summary, *Snakemake* is a much improved version of *GNU Make*. Programmers will be able to create and execute own workflows using *Snakemake* once they learned the syntax and semantic of the *Snakemake* workflow definition language. However, as *Snakemake* does not offer a GUI or editor for workflow design, most experimentalists without programming skills will not be able to create their own workflows.

Nextflow

The idea behind the WMS *Nextflow* [25] is to use pipes to transfer information from one task to subsequent tasks. In Unix, pipes act as shared data streams between two processes whereby one process writes data to a stream and another reads that data in the same order as it was written. In *Nextflow*, different tasks communicate through channels, which are equivalent to pipes, by using them as input and output. A workflow consists of several tasks, which are denoted as processes and are defined using *Nextflow*'s own language. The commands that are executed by processes can be either bash commands or defined in *Nextflow*'s own scripting language. *Nextflow* also provides the possibility to apply a task on a set of input files that follow a specific filename pattern using a channel that is filled with the filenames at runtime.

By default, all running processes are killed by *Nextflow* if a single process causes an error. This is particularly inconvenient if tasks with long runtimes are processed (e.g. transcriptome assembly based on RNA-seq reads). However, alternative error strategies can be defined for each task before workflow execution, which allow to either wait for the completion of scheduled tasks, ignore execution errors for this process or resubmit the process. In the latter case, computing resources can also be adjusted dynamically.

In *Nextflow*, there is no encapsulation of integrated tools at all since the commands to execute are defined in the file containing the workflow. While this is advantageous for quickly executing simple tasks, reusing tasks in the same or other workflows requires code duplication. Furthermore, *Nextflow* also does not offer a GUI for workflow design, which makes it hard for beginners to create their own workflows as they must be written in *Nextflow*'s own very comprehensive programming language.

Conclusion

In this article, we present the WMS *Watchdog*, which was developed to support the automated and distributed analysis of large-scale experimental data, in particular next-generation sequencing data. The core features of *Watchdog* include straightforward processing of replicate data, support for and flexible combination of distributed computing or remote executors and customizable error detection that allows automated identification of technical and content-related failure as well as manual user intervention.

Due to the wide use of XML, most potential users of *Watchdog* will already be familiar with the syntax used in *Watchdog* and only need to learn the semantic. This is in contrast to other WMSs that use their own syntax. Furthermore, *Watchdog*'s powerful GUI also allows non-programmers to construct workflows using predefined modules. Moreover, module developers are completely free in which software or programming language they use in their modules. Here, the modular design of the tool library provides an easy way for sharing modules by simply sharing the module folder.

In summary, *Watchdog* combines advantages of existing WMSs and provides a number of novel useful features for more flexible and convenient execution and control of workflows. Thus, we believe that it will benefit both experienced bioinformaticians as well experimentalists with no or limited programming skills for the analysis of large-scale experimental data.

Availability and requirements

- Project name: *Watchdog*
- Homepage: www.bio.ifi.lmu.de/watchdog; Bioconda package: anaconda.org/bioconda/watchdog-wms; Docker image: hub.docker.com/r/klugem/watchdog-wms/
- Operating system: Platform independent
- Programming language: Java, XML, XSD
- Other requirements: Java 1.8 or higher, JavaFX for the GUI
- License: GNU General Public License (GPL)
- Any restrictions to use by non-academics: none

Additional files

Additional file 1: Overview on the *Watchdog* GUI. Contains an overview on the *Watchdog* GUI for designing workflows and a step-by-step instruction on how to use it for creating a simple workflow. (PDF 1177 kb)

Additional file 2: Replicate data analysis in *Watchdog*. Describes how to use process blocks for the automated analysis of data sets with many different replicates or conditions. (PDF 126 kb)

Additional file 3: Extending *Watchdog*. Describes how to use the plugin system to extend *Watchdog* by new executors or process blocks without changing the original *Watchdog* code. (PDF 118 kb)

Additional file 4: Computational overhead of *Watchdog*. Contains an analysis of the computational overhead of *Watchdog* and *Snakemake* for executing a workflow with a variable number of samples. (PDF 157 kb)

Acknowledgements
Not applicable.

Funding
This work was supported by grants FR2938/7-1 and CRC 1123 (Z2) from the Deutsche Forschungsgemeinschaft (DFG) to CCF.

Availability of data and materials
Watchdog is freely available at http://www.bio.ifi.lmu.de/watchdog.

Authors' contributions
MK implemented the software and wrote the manuscript. CCF helped in revising the manuscript and supervised the project. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 24 August 2017 Accepted: 5 March 2018
Published online: 13 March 2018

References

1. Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet. 2009;10:57–63.
2. Ingolia NT. Ribosome profiling: new views of translation, from single codons to genome scale. Nat Rev Genet. 2014;15:205–13.
3. Johnson DS, Mortazavi A, Myers RM, Wold B. Genome-wide mapping of in vivo protein-DNA interactions. Science. 2007;316:1497–502.
4. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012;489:57–74.
5. Rutkowski AJ, Erhard F, L'Hernault A, Bonfert T, Schilhabel M, Crump C, et al. Wide-spread disruption of host transcription termination in HSV-1 infection. Nat Commun. 2015;6:7126.
6. Decker TM, Kluge M, Krebs S, Shah N, Blum H, Friedel CC, et al. Transcriptome analysis of dominant-negative Brd4 mutants identifies Brd4-specific target genes of small molecule inhibitor JQ1. Sci Rep. 2017;7:1684.
7. Davari K, Lichti J, Gallus C, Greulich F, Uhlenhaut NH, Heinig M, et al. Rapid genome-wide recruitment of RNA Polymerase II drives transcription, splicing, and translation events during T cell responses. Cell Rep. 2017;19:643–54.
8. Taylor J, Schenck I, Blankenberg D, Nekrutenko A. Using galaxy to perform large-scale interactive data analyses. Curr Protoc Bioinforma. 2007. Chapter 10:Unit 10.5.
9. Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, et al. KNIME: The Konstanz Information Miner. In: Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007). Heidelberg-Berlin: Springer; 2007. p. 319–26.
10. Köster J, Rahmann S. Snakemake–a scalable bioinformatics workflow engine. Bioinformatics. 2012;28:2520–2.
11. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9:357–9.
12. Campbell MS, Law M, Holt C, Stein JC, Moghe GD, Hufnagel DE, et al. MAKER-P: a tool kit for the rapid creation, management, and quality control of plant genome annotations. Plant Physiol. 2014;164:513–24.
13. McAffer J, Lemieux JM, Aniszczyk C. Eclipse Rich Client Platform, 2nd ed. Boston: Addison-Wesley Professional; 2010.
14. Babraham, Bioinformatics Institute. FastQC: A quality control tool for high throughput sequence data. 2014. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
15. Bonfert T, Kirner E, Csaba G, Zimmer R, Friedel CC. ContextMap 2: Fast and accurate context-based RNA-seq mapping. BMC Bioinformatics. 2015;16:122.
16. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics. 2009;25:1754–60.
17. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25:2078–9.
18. Liao Y, Smyth G, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics. 2014;30:923–30.
19. Wang L, Wang S, Li W. RSeQC: quality control of RNA-seq experiments. Bioinformatics. 2012;28:2184–5.
20. R Core Team. R: A language and environment for statistical computing. Vienna; 2014. Available from: http://www.R-project.org/.
21. Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biol. 2010;11:R106.
22. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 2014;15:550.
23. Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, et al. limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res. 2015;e47:43.
24. Robinson M, McCarthy D, Smyth G. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010;26:139–40.
25. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017;35:316–9.

OXFORD (GIGA)n SCIENCE

GigaScience, 9, 2020, 1–12

doi: 10.1093/gigascience/giaa068
TECHNICAL NOTE

TECHNICAL NOTE

Watchdog 2.0: New developments for reusability, reproducibility, and workflow execution

Michael Kluge, Marie-Sophie Friedl, Amrei L. Menzel and Caroline C. Friedel *

Institut für Informatik, Ludwig-Maximilians-Universität München, Amalienstr. 17, Munich 80333, Germany

*Correspondence address: Caroline C. Friedel, LFE Bioinformatik, Institut für Informatik, Ludwig-Maximilians-Universität München, Amalienstr. 17, Munich 80333, Germany. E-mail: caroline.friedel@bio.ifi.lmu.de http://orcid.org/0000-0003-3569-4877

Abstract

Background: Advances in high-throughput methods have brought new challenges for biological data analysis, often requiring many interdependent steps applied to a large number of samples. To address this challenge, workflow management systems, such as Watchdog, have been developed to support scientists in the (semi-)automated execution of large analysis workflows. **Implementation:** Here, we present Watchdog 2.0, which implements new developments for module creation, reusability, and documentation and for reproducibility of analyses and workflow execution. Developments include a graphical user interface for semi-automatic module creation from software help pages, sharing repositories for modules and workflows, and a standardized module documentation format. The latter allows generation of a customized reference book of public and user-specific modules. Furthermore, extensive logging of workflow execution, module and software versions, and explicit support for package managers and container virtualization now ensures reproducibility of results. A step-by-step analysis protocol generated from the log file may, e.g., serve as a draft of a manuscript methods section. Finally, 2 new execution modes were implemented. One allows resuming workflow execution after interruption or modification without rerunning successfully executed tasks not affected by changes. The second one allows detaching and reattaching to workflow execution on a local computer while tasks continue running on computer clusters. **Conclusions:** Watchdog 2.0 provides several new developments that we believe to be of benefit for large-scale bioinformatics analysis and that are not completely covered by other competing workflow management systems. The software itself, module and workflow repositories, and comprehensive documentation are freely available at https://www.bio.ifi.lmu.de/watchdog.

Keywords: workflow management system; bioinformatics; automated biological data analysis; next-generation sequencing; reusability; reproducibility; open science tools

Background

As a result of improvements in sequencing technologies, sequencing costs have decreased massively in recent years [1]. While the first human genome sequence cost ~$2.7 billion and took 13 years to complete [2], companies now offer genome sequencing to private customers using state-of-the-art next-generation sequencing (NGS) technologies for <$1,000. In addition, other cellular properties can now be measured at large scale using NGS. This includes, e.g., the expression of genes (RNA sequencing [RNA-seq]) [3], protein binding to DNA (chromatin immunoprecipitation sequencing [ChIP-seq]) [4], open chromatin regions (assay for transposase-accessible chromatin using sequencing [ATAC-seq]) [5], and many more.

As a consequence, data analysis has become more complex with new challenges for bioinformatics, often requiring multiple interdependent steps and integration of numerous replicates and several types of high-throughput data. Because manual execution of all required analysis steps is cumbersome, time-

Received: 27 November 2019; **Revised:** 26 April 2020

consuming, and laborious to repeat, several tools have been developed for performing large-scale bioinformatics analyses. One group of tools consists of static analysis pipelines specifically designed for 1 application, e.g., transcriptome analysis [6, 7]. While these pipelines have the advantage that a particular analysis can be repeated without great effort, components of these analysis pipelines are often not easily reusable for other related applications. As an alternative, workflow management systems (WMSs) have been developed that support creation of such analysis pipelines (denoted as workflows in this context) from reusable components and allow (semi-)automated execution of these workflows. Popular WMSs are Galaxy [8], KNIME [9], Snakemake [10], and Nextflow [11] and differ in the implemented set of features, target audience, the required training period, usage fees, and more (for more details see the comparison in the first article on Watchdog [12] and at the end of this article).

Previously, we presented the WMS Watchdog for the distributed analysis of large-scale experimental data originating, e.g., from NGS experiments [12]. The core features of Watchdog include straightforward processing of replicate data, support for and flexible combination of distributed computing or remote executors, customizable error detection, user notification on execution errors, and manual user intervention. In Watchdog, reusable components are encapsulated within so-called modules, which are defined by an XSD file specifying the program to execute, input parameters, and return values of the module. In addition, modules can contain scripts or compiled binaries that are invoked in the module. There are no restrictions on included software or on the programming language used in additional scripts. Modules may also deploy required software internally using Conda [13], Docker [14], or similar tools.

A Watchdog workflow is defined in an XML format and consists of a sequence of tasks and dependencies between tasks. Each task uses 1 module and the same task can be automatically run on multiple samples or with multiple parameter combinations using so-called process blocks. This creates several subtasks, 1 for each sample or parameter combination. A workflow can either be created manually using any XML editor or the Watchdog graphical user interface (GUI) for workflow construction. While XML may be more complex than, e.g., YAML or JSON, it is widely used and numerous XML editors are available, e.g., plugins for Eclipse [15]. Furthermore, using the GUI requires neither understanding of XML nor programming skills and thus allows easy construction of workflows from a pre-defined set of modules. In this case, the only Watchdog syntax that has to be learned is how to reference variables.

Workflows can be executed using the Watchdog scheduler via a command-line interface or the GUI, which are both implemented in Java and thus platform-independent. The Watchdog scheduler continuously monitors the execution status of tasks and schedules new tasks or subtasks for execution if all tasks that they depend on finished successfully. The execution status of tasks is reported to the user via standard output, a web interface that allows manual intervention and (optionally) email.

In the workflow, different executors can be specified for different tasks. Currently, 3 types of executors are supported (local host, remote host via SSH, or computer clusters using SGE or SLURM). Thus, resource-intensive or long-running tasks can, e.g., be submitted to a computer cluster while less demanding tasks may be executed on the local host. Furthermore, Watchdog provides a plugin system that allows users with programming skills to add new executor types, e.g., for cloud computing, without having to change the original Watchdog code (for details see [12]).

In this article, we present Watchdog 2.0, a new and improved version of Watchdog with several new developments for module creation and documentation, reusability of modules and workflows, and reproducibility of analysis results, as well as workflow execution.

Implementation

Overview

In the following, we describe only new developments that were added in Watchdog 2.0. The general principle of Watchdog and features already present in the previously published version remain unchanged; thus, we refer the reader to our previous publication for a detailed introduction to Watchdog [12]. The central improvements provided by Watchdog 2.0 are the following and are described in more detail in subsequent sections (see Fig. 1 for an overview). First, Watchdog 2.0 now provides a GUI for semi-automatically creating a new module from a software's help page. Second, a standardized documentation format for modules was introduced in Watchdog 2.0. From module documentation files, a searchable module reference book can then be generated providing an overview and details on existing modules. Third, a community platform was created for sharing Watchdog modules and workflows with other scientists.

Improvements for reproducibility of analysis results comprise extensive logging of executed steps, including module and software versions, and the possibility to automatically generate a summary of the executed workflow steps, e.g., as a draft for an article methods section. In addition, we added fully integrated support for container virtualization or package managers in the form of so-called execution wrappers, in particular for Docker containers and the Conda package management system.

Finally, 2 additional execution modes were implemented to provide more comfort and flexibility in workflow execution. The resume mode allows execution of a workflow to be restarted by (re-)running only tasks that previously did not run (successfully) or were added or modified compared to the original execution. The second mode allows the scheduler to be detached from workflow execution without aborting tasks running on a computer cluster and reattaching to execution at a later time on the same or a different computer.

The GUI for module creation and all new command-line tools described in the following are implemented in Java and thus platform-independent.

Semi-automated module generation

To make a software package available for use in Watchdog workflows, a new module has to be created. Watchdog already provides a helper script for creating the module XSD file and (optionally) a skeleton Bash script that only has to be extended by the program call. Nevertheless, this requires manually listing all parameters for the module. The newly developed GUI moduleMaker [16] now automatically extracts parameters and flags from a software help page to more conveniently create the corresponding module.

The moduleMaker GUI uses sets of regular expressions matching common help page formats to parse the help page of a software. Currently, 8 pre-defined regular expression sets are provided, but users can also define new sets using the GUI and add them to the pre-defined list. When creating a module with the GUI, users may either choose 1 particular regular expression set explicitly or let moduleMaker rank the regular ex-

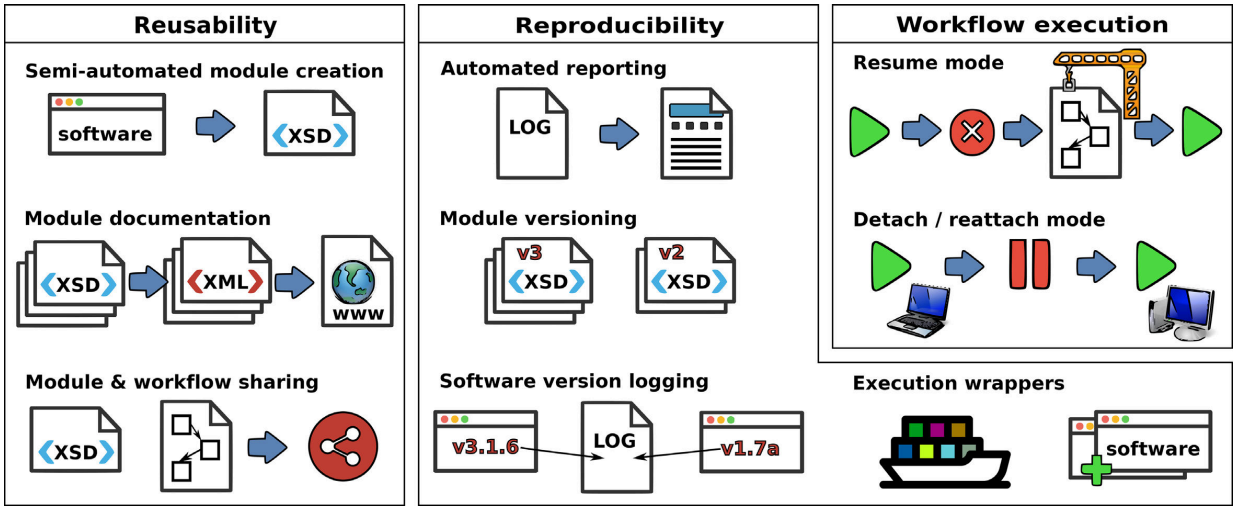


Figure 1: Overview on new developments in Watchdog 2.0. New features are broadly grouped into the categories reusability, reproducibility, and workflow execution. *Left:* New modules can now be developed in a semi-automated manner from software help pages using a GUI. A standardized documentation format was developed, allowing the automatic compilation of a reference book of available modules. Public repositories for sharing modules and workflows are now available. *Center:* Extensive logging of workflow execution ensures reproducibility of results and allows automated creation of a step-by-step report on analysis methods. Versioning of modules allows adaption to new requirements with backward compatibility without unnecessary module duplication. Software and module versions are now automatically reported in the log files. Execution wrappers now allow automatic deployment of software using container virtualization or package managers. *Right:* Workflow execution becomes more flexible with the resume and detach/reattach modes. The resume mode allows the resumption of interrupted or modified workflow execution without unnecessarily rerunning tasks. Detach/reattach allows the scheduler to be shut down on the local host while tasks on a computer cluster continue running and reattaching to workflow execution on the same or a different computer at a later time.

pression sets based on how well they match the help page. In the latter case, the user can then examine the results of the n best-matching regular expression sets (with n user-defined) and choose the result they consider best. Subsequently, the user can correct errors in the automatic detection, add additional flags or parameters, and modify or delete detected parameters. In a next step, existence checks for input files or directories can be added and return values for the module can be defined.

Once the user is finished, moduleMaker creates the module XSD file and a wrapper Bash script for the software that—in contrast to the skeleton Bash script created by the helper script—is almost complete. The only manual changes required by the developer involve assigning values to return values. This wrapper script checks that required software is installed, parses parameters, verifies that mandatory parameters are set, performs existence checks on required input files and directories, executes the program, performs default error checks after execution, and writes return values to a corresponding file read by the scheduler. Optionally, a project file can be saved that allows modules created with the moduleMaker to be reloaded and modified at a later time.

Thus, developing a module does not require understanding XML or the module XSD schema. Furthermore, little or no Bash scripting experience is required if the GUI or helper script is used, respectively. The GUI creates a Bash script that is finished apart from the return value assignment. If the helper script is used, there is no requirement to use a Bash script to execute the commands. Any type of executable can be called in the module, e.g., a Python script. Examples for modules using Python scripts are included in the new module repository (see below).

Module documentation

While the Watchdog scheduler, features of Watchdog workflows, and workflow creation are already comprehensively documented [12], no convenient way was so far available for documenting both individual Watchdog modules and the set of available modules. To address this problem, we developed (i) a standardized documentation format for modules and (ii) a program for creating a nicely formatted, searchable, and updatable catalog of modules, the so-called reference book (see Fig. 2 for an example), from the documentation files of individual modules. The module entry in the reference book describes software dependencies, parameters (i.e., input files and values) and their default values, return values (i.e., output files and values), and more. Thus, instead of inspecting the module XSD or input mask in the GUI to obtain this information, users can now simply browse the reference book.

Documentation format

Individual Watchdog modules are now documented using a standardized XML format. This contains general module information (e.g., author, description, dependencies) and properties of module parameters and return values (e.g., name, type, description). The allowed semantic is described by an XSD schema file, allowing the XML documentation files to be read and further processed by XML parsing software.

To limit the overhead for creating the module documentation, a command-line tool (docuTemplateExtractor) is provided by Watchdog 2.0. The docuTemplateExtractor extracts parameter and return value information from the module XSD file and generates a template documentation file. Module developers then only have to fill in parts of the XML documentation not contained in the module XSD file.

As noted above, modules may also contain additional scripts, which can contain further information useful for documentation. For example, many scripts utilize an argument parser that requires a description or default values for each parameter. To exploit this and guarantee consistency between documentation

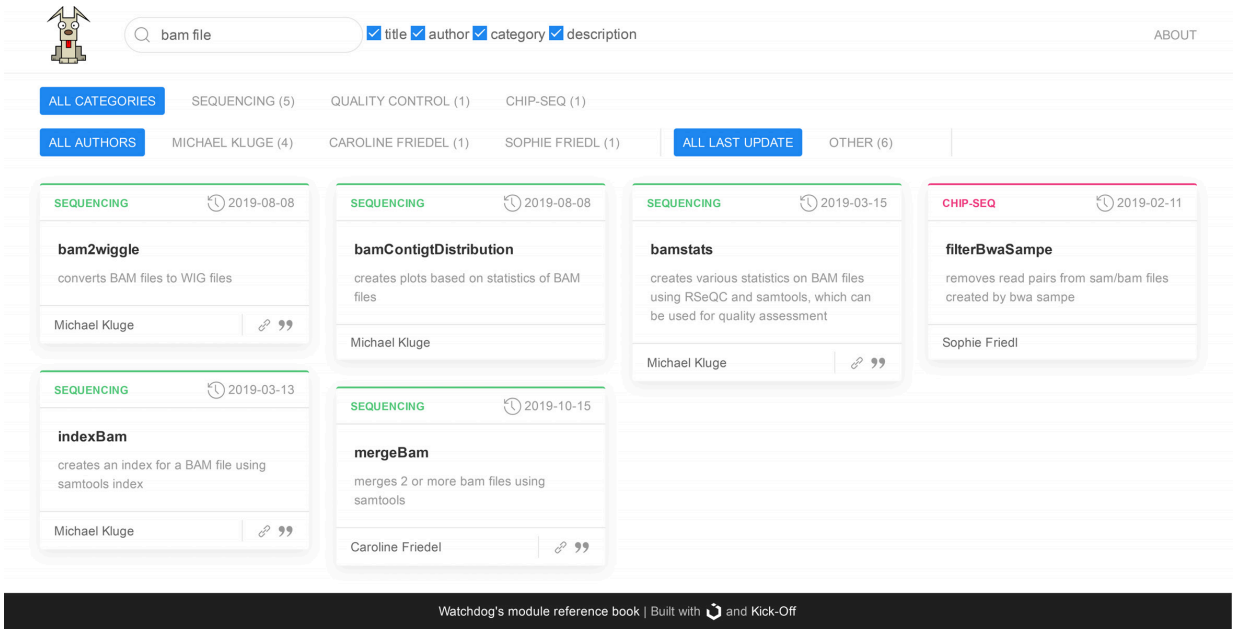


Figure 2: Overview page of the module reference book. The main section displays available modules as boxes, showing the module name, date of last change, a short description, links, and the author of the module. A search bar and category bar can be used to filter the displayed modules using text search or multi-category filters. In this example, all modules containing the term "bam file" in the description are shown.

and scripts, the docuTemplateExtractor also aims to extract this information. Because the syntax used by the argument parser strongly depends on both the scripting language used and the argument parser, this information cannot be obtained with a generalized approach. Instead, we developed a plugin system that allows developers to load custom parameter and return value extractors by implementing a simple Java interface. Currently, 2 parameter extractors for Bash- and Python-based modules are available, which obtain description and default value of parameters from argument parser definitions. For Bash scripts, the shFlags library is supported, and for Python, the argparse library.

Reference book

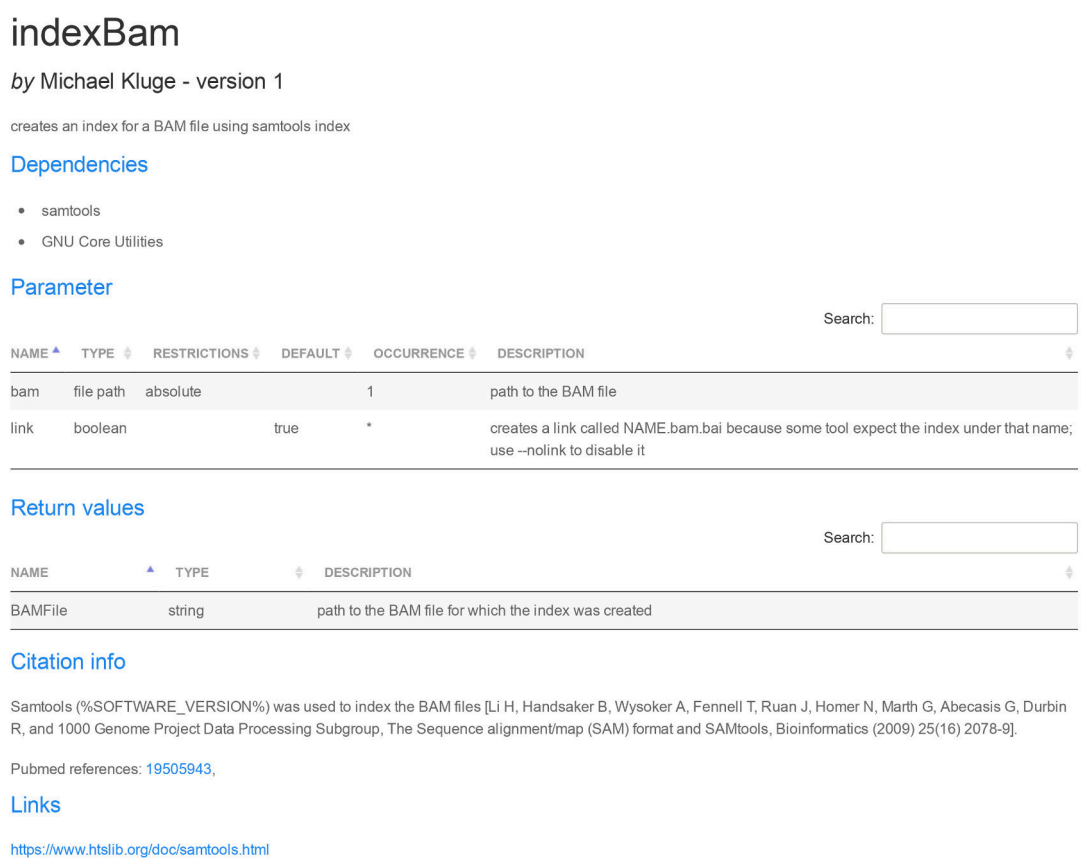
The reference book is implemented as an HTML web page based on the UIkit framework [17]. It can be opened with any browser supporting JavaScript and does not require a dedicated web server. The reference book can be created from the XML documentation files using the refBookGenerator command-line tool. The reference book can be created either for publicly available modules, personal modules of the user, or a combination of both. When new modules are added or existing modules are removed, the reference book can simply be regenerated using the refBookGenerator. Thus, every user can generate their personalized reference book containing the modules they work with or consider relevant to their work.

Fig. 2 shows the front page of the module reference book (generated for all publicly available modules) after searching for modules containing the term "bam file" in the description. The main section of the front page provides an overview on all available modules. Every module is visualized as a box that contains its name, author, assigned category, and a short description. The search bar at the top can be used to filter modules using a keyword search, which can be applied to title, author, category, and/or description. Alternatively, the modules displayed in the overview section can be filtered on the basis of authorship, category, and update date. Clicking on a module box opens a detailed view, showing module dependencies, parameters and valid input values, return values, and if applicable citation information and web links (see Fig. 3 for an example).

Public repositories for module and workflow sharing

Watchdog 2.0 now provides 2 repositories on Github under the watchdog-wms organization [18] that are dedicated for sharing modules [19] and workflows [20], respectively, by other users. To contribute either a module or workflow to one of the repositories, users have to first create a copy (fork) of the repository, change or add modules/workflows, commit the proposed changes to the repository copy, and submit these changes for review to the original repository via a pull request. An integration pipeline then checks whether the proposed changes adhere to essential requirements. If all automatic tests were successful, the proposed changes can be accepted by Watchdog team members.

Currently, the module repository contains 60 modules. Each module is located in a separate directory and must contain at least the XSD module file and an XML documentation file. Currently, most available modules focus on sequencing data analysis, in particular RNA-seq and ChIP-seq analysis. Some modules provide basic functionalities like file compression or text search while others fulfill more specific tasks, e.g., differential gene expression analysis (module DETest), peak detection in ChIP-seq data (module GEM), or identification of circular RNAs (modules circRNAfinder and ciri2). By default, modules are licensed under Apache License 2.0, but a different license can be assigned to a module by including it in the module folder. A reference book for all modules in the repository is available [21].



indexBam

by Michael Kluge - version 1

creates an index for a BAM file using samtools index

Dependencies

- samtools
- GNU Core Utilities

Parameter

Search:

NAME	TYPE	RESTRICTIONS	DEFAULT	OCCURRENCE	DESCRIPTION
bam	file path	absolute		1	path to the BAM file
link	boolean		true	*	creates a link called NAME.bam.bai because some tool expect the index under that name; use --nolink to disable it

Return values

Search:

NAME	TYPE	DESCRIPTION
BAMFile	string	path to the BAM file for which the index was created

Citation info

Samtools (%SOFTWARE_VERSION%) was used to index the BAM files [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, The Sequence alignment/map (SAM) format and SAMtools, Bioinformatics (2009) 25(16) 2078-9].

Pubmed references: 19505943,

Links

https://www.htslib.org/doc/samtools.html

Figure 3: Detailed view of a module in the reference book. As an example, the detailed view on the indexBam module is shown, containing a short description, dependencies on third-party software, parameters with valid ranges and descriptions, return values, citation information, and web links. The citation information will also be included into the step-by-step report automatically created from the workflow execution log file.

It is automatically updated with every commit to the master branch of the module repository.

Workflows shared in the watchdog-wms-workflows repository also have to be located in separate directories. Each workflow directory has to contain the XML workflow file, a readme file, and optionally example data. Workflows should be documented with inline comments. Furthermore, lines that require modifications to adapt, e.g., to different computing environments or input data should be highlighted to allow everyone to quickly adapt the workflow. We recommend, but do not enforce, that paths or constant parameter values are not hard-coded in the task section of the workflow, but rather that global constants are defined in the settings section. A constant "CONSTANT" can then be referenced as "${CONSTANT}" within process block or task definitions. If this recommendation is followed, the workflow can be quickly adapted to a new environment or data by modifying only constants and executors.

Currently, the workflow repository contains, e.g., the workflow for RNA-seq mapping and differential gene expression analysis from the original Watchdog release. Additionally, new workflows are available, e.g., for circular RNA detection with CIRI2 [22] and circRNA_finder [23], ChIP-seq analysis using GEM (GEM, RRID:SCR_005339) followed by ChIPseeker [24,25], and download of public NGS data from the NCBI SRA (SRA, RRID:SCR_004891) [26] followed by alignment with HISAT2 (HISAT2, RRID:SCR_015530) [27].

Methods for ensuring reproducibility

A critical aspect of any analysis of biological data is the reproducibility of the results. While the use of a WMS already contributes to reproducibility, workflows may be modified between different runs of the workflow, e.g., by changing parameter values or including or excluding some steps, or the underlying software may be changed, e.g., by updates to a new version. This may lead to uncertainty regarding the steps, parameters, and software environment of the analysis that produced specific results. Furthermore, when reporting the individual steps of an analysis, for instance in a publication, some steps may be unintentionally omitted, making it difficult for others to reproduce the results. To address these problems, Watchdog 2.0 includes a number of new developments to ensure reproducibility of analyses.

Logging and automated reporting

When executing a workflow, Watchdog 2.0 now produces a time-stamped log file (filename extension .resume) reporting on the successful execution of each individual task. This log file is also used for the resume mode (see below). If a task creates multiple subtasks, e.g., for multiple input samples, successful execution of each subtask is recorded. For each task/subtask the log file records the value of each input parameter, as well as return values.

Quality of the sequencing data was checked using FastQC (0.11.3). RNA-seq reads were mapped against the XXX genome using ContextMap (2.7.9) with BWA as short read aligner and default parameters. Samtools (1.9) was used to convert SAM to BAM files. Samtools (1.9) was used to index the BAM files. Quality of the resulting mappings was assessed using RSeQC (3.0.0). FeatureCounts (1.4.6) was applied to count read/fragment counts per gene/exon/other feature according to Ensembl_Homo_sapiens.GRCh38.78.chr21.gtf annotation. Differential gene expression analysis was performed using DESeq2 (1.22.2).

Figure 4: Result of automated report generation for example workflow. This example shows the step-by-step analysis report generated with the reportGenerator from the execution log file for the RNA-seq example workflow provided with Watchdog. The workflow was described in detail in our original Watchdog publication [12]. The annotation file name (a parameter to featureCounts) and software version numbers in parentheses are automatically obtained from the log file (see software version logging). For this example, the workflow was simplified to perform differential gene expression analysis only with DESeq2, instead of 4 different gene expression analysis methods as previously described. For modules without paper description (e.g., unzipping or replicate merging), the report would contain the text "No short description given in documentation of module *module name*". To shorten this example, these sentences were manually removed, as well as the citation information commonly included in the module descriptions.

Moreover, a report of the executed steps can be automatically created from the log file using the new command-line tool reportGenerator provided with Watchdog 2.0 (see Fig. 4 for an example of the report). For this purpose, the XML documentation file of each module contains the element paperDescription, which can be filled with a short description of the module and citation information. It can also contain references to parameters of the task or the software version (see below for software version logging). The reportGenerator concatenates these descriptions in the order the corresponding tasks were executed and replaces references by the values reported in the log file. There is also an option to include PubMed IDs from the module documentation. The resulting report can then be used as a step-by-step protocol of the analysis or be further revised for the methods section of an article.

Module versioning

Modules generally rely on third-party software that can be modified repeatedly to improve performance, fix bugs, or be adapted to changing requirements, for instance by adding support for new types of experimental data. As a consequence, a module will need to be adapted over time, e.g., by changing the parameters of the module to support new parameters or drop obsolete ones. At the same time, backward compatibility needs to be ensured such that previously defined workflows relying on the old module version can still be executed. One solution to this problem would be to duplicate the module and adapt the copy. However, this leads to unnecessary code duplication because most of the module XSD file will remain unchanged, and results in code that is difficult to maintain.

To avoid this problem, Watchdog 2.0 now allows different versions of a module within 1 module XSD file to be defined by specifying the minimum and maximum supported module version for each element in the XSD file. If neither minimum nor maximum supported version is indicated, the element is valid for all module versions. This allows input parameters, return values, or even the executed program call to be changed between different module versions. When executing a workflow, the module version for each task will also be recorded in the log file. By default, the first version of a module is used unless otherwise specified in the workflow XML file. This guarantees that workflows defined before a new module version was introduced do not have to be adapted.

Software version logging

Watchdog is very flexible with regard to how dependencies to third-party software in a module can be handled by module developers. Software can be shipped with the module, loaded via package and environment management systems like Conda [13], or be required to be installed on the system that will execute the corresponding task (e.g., the local host or a computer cluster). In any case, it is crucial to know which software versions were run for a particular analysis in order to reproduce the analysis results or understand differences in outputs between repeated runs because new software releases often correct errors or may change the behavior of the software.

Thus, Watchdog 2.0 now implements a general approach for reporting versions of third-party software used in a module in the log file. For this purpose, a new attribute in the module XSD file can be used to define the flag for version printing of third-party software. During workflow execution, after a task or subtask has been completed successfully on a particular computer, the program call defined in the corresponding module is invoked with the version flag on the same computer to retrieve the installed third-party software version. This software version is then reported for the task/subtask in the log file. If the version flag has not been defined in the module, this step is omitted for the corresponding tasks. This option is also useful for identifying differences in installed third-party software between different executors used for workflow execution, such as the local host, a computer cluster, or remote executors accessed by SSH.

Execution wrappers

A disadvantage of Watchdog's flexibility on how installation of third-party software is handled is that it complicates both reusability and reproducibility of workflows. Having to install all required software before modules or workflows can be used can be cumbersome. Furthermore, to fully reproduce results from a workflow, users would have to make sure that they (still) have the same software versions installed as in the original run of a workflow. Thus, we now implemented execution wrappers to explicitly support automatic deployment of software via package managers or container virtualization in Watchdog 2.0. Execution wrappers are initialized in the settings section of a Watchdog workflow and are then assigned to individual executors, which in turn use the wrapper to deploy the software for all tasks they run. Each executor can be assigned both a package manager and a container; thus, package managers can also be used within containers. Furthermore, different packager managers or containers can be assigned to different tasks by using different executors and corresponding execution wrappers for these tasks. Execution wrappers are implemented using Watchdog's plugin system; thus, the set of available execution wrappers can be extended by users without having to modify the Watchdog code.

Currently, Watchdog 2.0 provides execution wrappers for the Conda package manager (Conda, RRID:SCR_018317) [13] and for Docker container virtualization [14]. To enable use of Conda for a module, the module directory only has to contain a YAML file defining the default Conda environment (modulename.conda.yml). For different versions of a module, different Conda environments can be defined (ending in .v[0-9]+.conda.yml). If no version-specific Conda definition file is

found, the default Conda environment for the module is used. If Conda execution wrappers are not used in a workflow or for a particular executor, the Conda environment definition will simply be ignored for the whole workflow or the tasks run by the executor, respectively. Thus, previously developed workflows will not be affected by these changes.

The Docker execution wrapper allows tasks to be run within containers built from Docker images using Docker, Podman, or Singularity. Furthermore, it provides an option for automatically mounting files and directories on the host machine that are used in parameters of tasks. This option is enabled by default but can be disabled. Thus, adding container virtualization to an executor does not require changes to corresponding tasks. Similar to the Conda execution manager, module- and module-version-specific Docker images can be enabled by adding 1 or more files to the module folder specifying the image name to be used for the corresponding tasks. An example for using Docker and Conda in combination is provided in the workflow for RNA-seq mapping and differential gene expression analysis available from the workflow repository and with the Watchdog distribution.

New execution modes

In the original Watchdog version, the Watchdog scheduler had to run continuously on the computer on which workflow execution is started. If workflow execution was interrupted, e.g., by a computer crash or reboot, only a manual restart option was available. This required the last task finished successfully to be identified or some analyses to be rerun in case only some subtasks of a task finished successfully. To avoid this problem, Watchdog 2.0 now supports 2 additional execution modes (see Fig. 5). The first one allows workflow execution to be resumed at any point and rerunning only the tasks or subtasks in a workflow that did not finish successfully, were modified, or depended on modified tasks. The second execution mode allows detachment from workflow execution by shutting down the Watchdog scheduler on the current computer while tasks distributed to a computer cluster continue running. The scheduler can then reattach to the workflow execution at a later time either from the same or a different computer. This can be used for instance to reboot the machine running the scheduler or to switch from a desktop computer to a laptop without interrupting execution of tasks running on a computer cluster.

Resume mode

As described above, Watchdog 2.0 creates a detailed log file during execution of a workflow containing successfully finished (sub)tasks, as well as their input parameters and return values. In resume mode, Watchdog 2.0 uses the log file of a previous workflow run to determine which (sub)tasks have to be (re-)executed. Individual (sub)tasks are identified by their input parameter combinations. (Sub)tasks not listed in the log file with exactly the same input parameter values will be scheduled to be executed. Furthermore, (sub)tasks that previously finished successfully with the same parameters are re-executed if they depend on other (sub)tasks that are (re-)run.

This allows the resumption of not only workflows that were interrupted unexpectedly (e.g., by hardware failure or power outage) but also workflows that were modified, i.e., by changing parameters for some tasks, without unnecessarily rerunning tasks. Here, Watchdog 2.0 guarantees that all results are updated that may be affected by the modification. Furthermore, additional samples, e.g., for other conditions or more replicates, can be easily included without rerunning analyses for samples that have already been processed. Importantly, identification of (sub)tasks that require (re-)execution is performed automatically without manual user input. This both reduces the overhead for the user and eliminates the risk that they may forget some steps that need to be repeated.

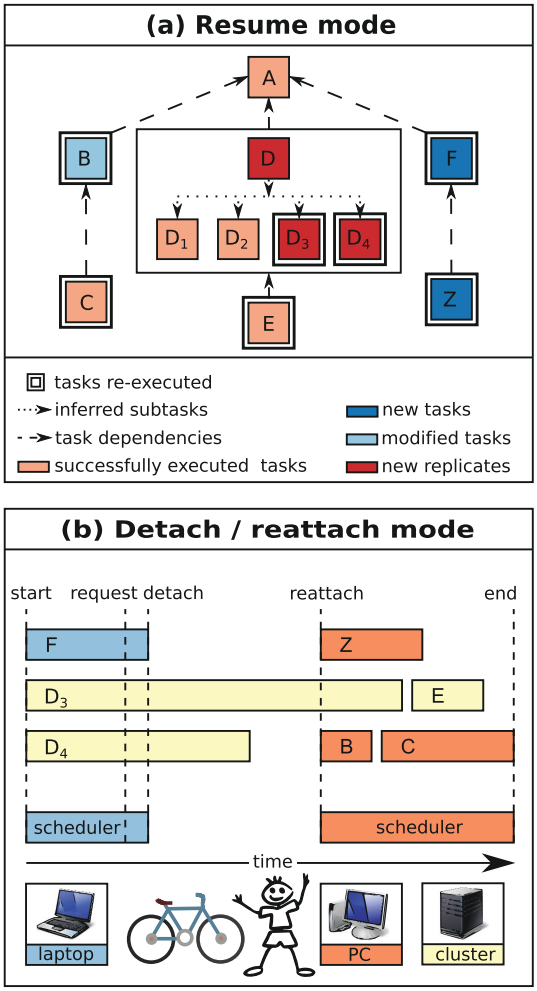


Figure 5: New execution modes in Watchdog 2.0. **(a)** Resume mode: From the log file of a previous workflow execution and the workflow XML file, the Watchdog scheduler automatically detects (sub)tasks that either have not yet run successfully, are new or modified, or require processing of additional samples. Consequently, only these (sub)tasks are executed, as well as all (sub)tasks depending on them (dependencies indicated as dashed lines). Light red indicates (sub)tasks that were previously executed successfully; light blue, tasks that were since modified; dark blue, new tasks that were added; and dark red, additional subtasks that have to be executed because additional samples were added. Double lines around tasks indicate which (sub)tasks have to be (re-)executed after resuming this workflow. **(b)** Visualization of the detach/reattach mode after resuming the workflow shown in (a). In this case, subtasks of D and task E are executed on a computer cluster, while a local executor is used for all other tasks. In this example, the Watchdog scheduler is originally started on a laptop and some tasks are scheduled and executed. After a while, a detach request is sent and no more new tasks are scheduled on the local host. Once the tasks on the local computer (blue) have finished, the detach file is written and the scheduler terminates. Subtasks D_3 and D_4 submitted to the computer cluster (yellow) continue to be executed. When the user reattaches to workflow execution, this time on a desktop computer (orange), new tasks are again scheduled.

The Watchdog 2.0 resume mode is illustrated in Fig. 5a for an example workflow. In this case, a task was modified (task B); additional samples were added to task D (marked red), requiring additional subtasks to be run; and additional tasks were added (F and Z). In resume mode, task B will be rerun because of modified parameters and task C because it depends on task B. For task D, only the new subtasks will be executed, but task E will be repeated because it depends on D. In addition, the newly added tasks will be run.

It should be noted here that after a workflow has been run at least once, changes to the workflow should be limited to adding new (sub)tasks (e.g., for new samples) or dependencies. Removing (sub)tasks or dependencies between tasks may lead to inconsistencies with old versions of input data being accidentally used for a task. Thus, this should only be done with utmost care.

Detach / reattach mode

In most cases, the Watchdog scheduler will run on a laptop or desktop computer and outsource all resource-intensive tasks to a distributed computer system, e.g., a computer cluster. Because execution of long, resource-intensive workflows may take hours or even days to complete, it may not always be possible for the Watchdog scheduler to be running continuously on the host computer. For instance, the host running Watchdog might require a reboot to install software updates or dedicated computer cluster submission hosts may not allow long-running programs. If the Watchdog scheduler is run on a laptop, the user may want to change locations with their laptop. To support these use cases, Watchdog 2.0 now provides the option to detach the scheduler from a running workflow and reattach at a later time. Notably, the user does not have to decide before execution whether to use this mode but can decide to detach at any time after starting execution in either normal or resume mode.

In Watchdog 2.0, the user can request to detach using a keystroke combination (Ctrl-C) or a link in the email notification. After the request is sent, Watchdog will wait for tasks to complete that are running either on the local host or a remote host via SSH, but schedule no further tasks on these executors. In contrast, Watchdog will continue to submit tasks on cluster executors with workload managers working independently of Watchdog (currently SGE and SLURM are supported). Once all tasks on the local and remote hosts are finished, Watchdog will save the information on tasks running on cluster executors to a file and then terminate itself. From this moment, tasks already running on or submitted to computing clusters will continue running or be scheduled to run by the corresponding workload managers, but no new tasks can be submitted to these clusters.

The detach file can then be used at a later time to reattach to workflow execution at the point where it was stopped previously. Watchdog will then obtain information on the execution status of tasks that were still running on or submitted to computer clusters before detaching, i.e., whether they are still running or finished successfully or with errors, and continue scheduling tasks on all executors accordingly. Notably, the Watchdog scheduler can also be reattached on another computer using the detach file, allowing for instance switching from a laptop at home to a desktop computer at work as illustrated in Fig. 5b. Moreover, Watchdog 2.0 also provides a command-line tool to periodically start the scheduler in auto-detach mode. In this mode, the scheduler checks whether tasks were finished successfully, submits new tasks if possible, and then terminates itself automatically.

Comparison to other WMSs

In this article, we present a number of new developments in our WMS Watchdog. The previously published version of Watchdog [12] was already extensively compared against the most popular WMSs for biological analyses, i.e., Galaxy (Galaxy, RRID:SCR_006281) [8], KNIME (KNIME, RRID:SCR_006164) [9], Snakemake (Snakemake, RRID:SCR_003475) [10], and Nextflow [11] (see Fig. 12 in [12] for this comparison). Compared features included, e.g., availability of GUIs/web interfaces for workflow design, execution and monitoring, support for parallel, distributed, and cloud computing, dependency definition, and many more. This showed that Watchdog combined features of existing WMSs and provided novel useful features for execution and monitoring of workflows for users both with and without programming skills.

Because these features are essentially unchanged, we do not repeat this comparison here but refer the reader to our original publication [12]. In the following, we discuss how Watchdog 2.0 compares to these other WMSs regarding the new features we present in this article because these were not previously analyzed. First, we provide a brief description of Galaxy, KNIME, Snakemake, and Nextflow. For more details, see our original publication [12].

Galaxy is targeted at experimentalists without programming experience and allows data analyses to be performed in the web browser. Workflows can be constructed on public or private Galaxy servers in a web-based user interface from a set of available tools and can then be executed. New tools for use in a Galaxy workflow are defined in an XML format specifying the input parameters for this tool, as well as the program to execute.

KNIME is an open-source data analysis platform based on the Eclipse integrated development environment (IDE). It provides a powerful GUI for workflow construction, execution, and visualization of results, which can also be used without programming experience. Java programming skills are required for making a new tool available in a so-called node because multiple Java classes have to be extended.

Snakemake uses a Python-based language to define workflows in a so-called Snakefile as a set of rules that describe how output files are created from input files. Dependencies between rules are determined automatically on the basis of input and output files, and the order of rule execution is determined upon invocation based on these dependencies. Encapsulation of reusable components can be performed using so-called wrappers. Writing workflows and wrappers requires knowledge of the Snakemake syntax and some degree of programming skills.

Nextflow extends the Unix pipes model to transfer complex data between consecutive processes as shared data streams. It provides its own scripting language based on the Groovy programming language to define workflows. Individual analysis steps are defined as processes in the Nextflow workflow itself; thus, no actual encapsulation of tools into reusable components is supported. Similar to Snakemake, programming experience is required to define workflows and no GUI is provided.

For the following comparison, features were grouped broadly into categories reusability, reproducibility, and workflow execution. A summary of the comparison is presented in Table 1.

Reusability

For this part of the comparison, we focused on features that support development and sharing of tools (modules in Watchdog, tools in Galaxy, nodes in KNIME, rules in Snakemake, processes in Nextflow) for (re-)use in multiple analysis workflows as well as sharing and repurposing of existing workflows (F1–F7 in Ta-

Table 1: Comparison of Watchdog with 4 other commonly used WMSs.

Category	No.	Feature	Watchdog	Galaxy	KNIME	Snakemake	Nextflow
Reusability	F1	Support for tool creation	Command-line/GUI	Command-line[1]	Eclipse Wizard	No	NA
	F2	Tool documentation	XML based	XML based	XML based	YAML based[2]	NA
	F3	Tool reference book	Web page generator	Part of GUI	Part of GUI	Web page generator	NA
	F4	Tool versioning	Yes	Yes	Yes	Yes	NA
	F5	Sharing of tools	Repository[3]	ToolShed[4]	KNIME Hub[5]/NodePit[6,*]	Repository[7]	NA
	F6	Sharing of workflows	Repository[3]	ToolShed[4]	KNIME Hub[5]/NodePit[6,*]	Repository[8]	nf-core[9,*]
	F7	Repurposing of workflows	XML edit/GUI	GUI	GUI	Copy Snakefile	Command-line
Reproducibility	F8	Software version logging	Yes	No	No	Yes	No
	F9	Software deployment	Execution wrappers, Conda, Docker	Conda, Docker	No	Conda, Docker	Conda, Docker
	F10	Creation of workflow report	Yes	List via history	Static description[10]	HTML report	HTML report
	F11	Citation export	Yes	Yes	No	No	No
Execution	F12	Resume workflow	Yes	No	Yes	Yes	Yes
	F13	Process only updated tasks	Yes	No	Yes	Yes[11]	Yes
	F14	Process only new replicates	Yes	No	No	Yes[11]	Yes
	F15	Detach/reattach	Yes	Yes[12]	Non-free feature[13]	Yes[14]	No

The selected WMSs are compared against Watchdog based on features grouped broadly into the categories reusability, reproducibility, and execution. NA: not applicable. [1]Python-based command-line program (Planemo). [2]No explicit documentation of parameters but example Snakefile and wrapper source code is part of the documentation. [3][18]. [4][28]. [5][29]. [6][30]. [7][31]. [8][32]. [9][33.] [10]A description that was manually created for a specific workflow can be displayed but is not dynamically created. [11]Flag "–list-params-changes" or "–list-input-changes" in combination with the "–forcerun" flag. [12]Client: anytime/server: if jobs are not executed locally on the server. [13]Non-free SGE extension or KNIME server required. [14]Sending of a TERM signal stops scheduling of new jobs and waits for all running jobs to finish; Ctrl+C kills all jobs running on the local computer, while jobs running on a computing cluster continue to run. *Community project.

ble 1). Because there is no real encapsulation of tools in Nextflow, most of these features are not applicable to it.

Support for tool creation (F1) is provided in Galaxy by the command-line program Planemo, which is similar to the helper script originally provided by Watchdog for module creation. Notably, Planemo also requires all parameters for a new tool to be added manually. For KNIME, an Eclipse extension (KNIME Node Wizard) is available, which generates the project structure, the plug-in manifest, and all required Java classes. However, the Java classes only contain the basic backbone (in particular, no parameters or flags) and have to be massively extended by the developer. Snakemake does not provide any software or script for defining wrappers.

All 3 WMSs allow documenting (F2) tools and their parameters in XML or YAML format. In the case of Snakemake, the specification does not require explicit documentation of parameters and input and output. Instead, an example Snakefile showing the use of the wrapper has to be provided. A reference book containing information on all available tools (F3) can be generated for Snakemake wrappers as a separate web page. This contains the example Snakefile, the code of the wrapper, author information, and software dependencies. In contrast, the documentation of KNIME nodes and Galaxy tools, respectively, is displayed on their respective GUI/web interface during workflow creation. Furthermore, all 3 WMSs perform tool versioning (F4).

For sharing tools (F5) or complete workflows (F6) with other users, Galaxy and KNIME operate dedicated sharing platforms [28, 29], while Snakemake provides source code repositories similar to Watchdog 2.0 [31, 32]. Furthermore, dedicated sharing platforms are operated by the KNIME and Nextflow community [30, 33].

Repurposing an existing workflow for new data (F7) requires different steps in the different WMSs. In Galaxy and KNIME, existing workflows can be imported and subsequently input files or values have to be selected/modified in the web interface and GUI, respectively. For Nextflow, input is provided via command-line parameters. For Snakemake, relative paths to input files are hard-coded in the Snakefile. Thus, repurposing a Snakemake workflow only requires the Snakefile to be copied to a directory in which input files are stored or linked in the subdirectory structure used in the Snakefile. In Watchdog workflows, input files and parameters are also hard-coded but absolute paths are used. In a well-designed workflow, global constants are defined for input values and files in the settings section and used throughout the workflow. Thus, repurposing only requires these constants to be edited either in a text or XML editor or the GUI. This is not more effort than required by other WMSs, with the exception of Snakemake. However, it provides more flexibility than Snakemake regarding how input data are distributed in the file system, and workflows can be stored anywhere, e.g., in a directory containing all previously developed workflows.

Reproducibility

Here, we focus on features (F8–F11) related to reproducibility of analysis results carried out at an earlier time, on different computer systems, and/or by other scientists. Most of the other WMSs do not support explicit logging of external software during workflow execution similar to Watchdog 2.0 (F8). However, Galaxy, Snakemake, and Nextflow support controlling external software dependencies and versions with the Conda package manager or using Docker containers (F9). Furthermore, Snakemake reports on executed workflows (see next paragraph) display the Conda environment for each task, including software versions.

A description of all performed analysis steps (F10) can be obtained in Snakemake and Nextflow through generation of HTML reports, in which individual steps are listed in a table format and in the case of Snakemake also visualized as a graph. Galaxy displays all executed tasks as a list in its analysis history. In contrast, KNIME supports only static workflow descriptions that have to be prepared by the workflow developer. The dynamic report created by Watchdog 2.0 from the execution log does not only list performed steps but includes short descriptions of each step prepared by module developers with citation information and (optionally) PubMed IDs (F11). The only other WMS allowing the declaration of citations for tools is Galaxy. In this case, a list containing citations for all tools used can be exported after executing a workflow in Galaxy. None of the other WMSs support creation of a step-by-step report for inclusion in a manuscript draft similar to Watchdog 2.0.

Execution

All WMSs except Galaxy can resume execution of partly executed workflows (F12) and are able to detect new tasks, modified tasks, or tasks with altered dependencies and consequently execute only these tasks (F13). With Snakemake and Nextflow, new samples (e.g., additional replicates) can be included in an analysis workflow without having to reprocess all samples (F14), but this option has to be forcibly triggered in Snakemake. This is not possible for KNIME workflows. One possibility to avoid unnecessary reprocessing in KNIME is to implement KNIME nodes that can detect whether the corresponding task was already executed successfully on a sample as done by Hastreiter et al. [34]. However, this adds additional overhead for node development.

Finally, similar execution modes to the detach/reattach mode of Watchdog 2.0 (F15) are at least partly supported by all compared WMSs apart from Nextflow. Because Galaxy is a web-based system, the user can log off (detach) and log in (reattach) at any time and from different client systems. Furthermore, the Galaxy server can also be restarted while tasks continue running on a computer cluster if no tasks are executed locally on the server. In KNIME, remote execution is only possible with non-free extensions like the KNIME Server or a cluster extension. If tasks are executed remotely using such an extension, the local KNIME instance can be detached and reattached to workflow execution. Finally, Snakemake provides the option to stop scheduling by sending the TERM signal and waiting for all jobs to be finished before terminating. Later, workflow execution can then simply be resumed. However, this mode also stops scheduling of jobs on computer clusters and waits for jobs running on computer clusters to be finished. Alternatively, Ctrl+C kills the main Snakemake process and all jobs running on the local computer, but jobs already running on a computing cluster keep running. With the correct use of profiles, it is then possible for the workflow to check the status of those jobs after a restart.

Conclusion

In this article, we present the new developments in Watchdog 2.0, which focus on improving reusability of modules and workflows, reproducibility of analysis results, and convenience of workflow execution.

To simplify module development, we developed the moduleMaker GUI for semi-automatically creating a module for a software package by parsing its help page. Manual overhead for the module creator is then mostly limited to choosing the best regular expression set, validating and correcting automatically identified parameters, and adding additional parameters or re-

turn values considered necessary. Furthermore, we established public sharing repositories to support and encourage exchange of developed modules and workflows between scientists. Modules are now documented in a standardized documentation format, from which an HTML-based module reference book can automatically be created. The reference book provides an overview and details on available modules and can be easily regenerated to integrate new modules, e.g., modules created by other developers.

To guarantee reproducibility of workflow results, we introduced module versions and extensive logging of successfully executed steps including parameter values and third-party software versions. From the log file of a workflow execution, a report can then be automatically generated that serves both as a documentation of the analysis steps and as a starting point for drafting the corresponding methods section of a manuscript. This not only reduces the effort in creating a description of the analysis, it also prevents accidental omission of individual steps. In addition, Watchdog 2.0 now provides integrated support for automatic deployment of software, in particular with Conda or Docker, in the form of execution wrappers.

Finally, with the new resume and detach/reattach execution mode, convenience and flexibility of workflow execution is greatly enhanced in Watchdog 2.0. The resume mode not only implements the state of the art for WMSs that allows resumption of interrupted workflow execution, but automatically identifies and re-executes tasks with modified parameters or additional input samples as well as downstream tasks that depend on them. The detach/reattach mode allows shutting down the Watchdog scheduler on a local computer while jobs continue to be executed on computer clusters. The user can then reattach to workflow execution and resume scheduling of tasks at a later time and even from a different computer.

While many of the new features in Watchdog 2.0 are also present in other popular WMSs, none are implemented in all of them. Furthermore, even if these features are available in other WMSs, the implementations in Watchdog 2.0 often add additional capabilities, such as, e.g., the possibility to automatically generate a step-by-step report. Combined with the existing advantages of Watchdog highlighted in our original publication, we thus believe that Watchdog 2.0 will be of great benefit to users with a wide range of computer skills for performing large-scale bioinformatics analyses in a flexible and reproducible manner.

Availability of Source Code and Requirements

- Project name: Watchdog 2.0
- Project home page: https://www.bio.ifi.lmu.de/watchdog
- Source code: https://github.com/klugem/watchdog, https://github.com/watchdog-wms
- Operating system(s): Platform independent
- Programming language: Java
- Other requirements: Java 11 or higher, JavaFX 11 or higher for the GUIs, individual requirements for modules
- License: GNU General Public License v3.0
- DOI: https://doi.org/10.5281/zenodo.3764538
- RRID:SCR_018355
- biotoolsID: biotools:watchdog

Availability of Supporting Data and Materials

Snapshots of the Watchdog 2.0 code and the module and workfow repository used for this article are available in the GigaDB data repository [35].

Abbreviations

ATAC-seq: assay for transposase-accessible chromatin using sequencing; ChIP-seq: chromatin immunoprecipitation sequencing; GUI: graphical user interface; IDE: integrated development environment; JSON: Javascript Object Notation; NCBI: National Center for Biotechnology Information; NGS: next-generation sequencing; RNA-seq: RNA sequencing; SSH: Secure Shell; SRA: Sequence Read Archive; WMS: workflow management system.

Competing Interests

The authors declare that they have no competing interests.

Funding

This work was supported by grants FR2938/7-1, FR2938/10-1, and CRC 1123 (Z2) from the Deutsche Forschungsgemeinschaft (DFG) to C.C.F.

Authors' Contributions

M.K. developed the software and wrote the manuscript. M.-S.F. tested Watchdog 2.0 and implemented modules and the workflow for the analysis of circular RNAs in high-throughput sequencing data. A.L.M. implemented the moduleMaker GUI under supervision of C.C.F. and M.K. C.C.F. tested the software, helped in revising the manuscript, and supervised the project. All authors read and approved the final manuscript.

References

1. Hayden EC. Technology: The $1,000 genome. Nature 2014;**507**:294–5.
2. International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. Nature 2004;**431**:931–45.
3. Wang Z, Gerstein M, Snyder M. RNA-Seq: A revolutionary tool for transcriptomics. Nat Rev Genet 2009;**10**:57–63.
4. Furey TS. ChIP-seq and beyond: New and improved methodologies to detect and characterize protein-DNA interactions. Nat Rev Genet 2012;**13**:840–52.
5. Buenrostro JD, Wu B, Chang HY, et al. ATAC-seq: A method for assaying chromatin accessibility genome-wide. Curr Protoc Mol Biol 2015;**109**:21.29.1–9.
6. Guo W, Tzioutziou N, Stephen G, et al. 3D RNA-seq - a powerful and flexible tool for rapid and accurate differential expression and alternative splicing analysis of RNA-seq data for biologists. bioRxiv 2019, doi.org/10.1101/656686.
7. Sundararajan Z, Knoll R, Hombach P, et al. Shiny-Seq: Advanced guided transcriptome analysis. BMC Res Notes 2019;**12**:432.
8. Taylor J, Schenck I, Blankenberg D, et al. Using Galaxy to perform large-scale interactive data analyses. Curr Protoc Bioinformatics 2007;**Chapter 10**:Unit 10.5.
9. Berthold MR, Cebron N, Dill F, et al. KNIME: The Konstanz Information Miner. In: Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007). Heidelberg-Berlin: Springer; 2007:319–26.
10. Köster J, Rahmann S. Snakemake–a scalable bioinformatics workflow engine. Bioinformatics 2012;**28**:2520–2.
11. Di Tommaso P, Chatzou M, Floden EW, et al. Nextflow enables reproducible computational workflows. Nat Biotechnol 2017;**35**:316–9.

12. Kluge M, Friedel CC, Watchdog – a workflow management system for the distributed analysis of large-scale experimental data. BMC Bioinformatics 2018;**19**:97.
13. Conda. https://conda.io. Accessed 11 November 2019.
14. Docker. https://www.docker.com/. Accessed 1 April 2020.
15. McAffer J, Lemieux JM, Aniszczyk C. Eclipse Rich Client Platform. 2nd ed. Boston, MA: Addison-Wesley Professional; 2010.
16. moduleMaker. https://github.com/watchdog-wms/moduleMaker. Accessed 23 April 2020.
17. Ulkit. https://getuikit.com. Accessed 11 November 2019.
18. Watchdog WMS Community, https://github.com/watchdog-wms/. Accessed 23 April 2020.
19. Watchdog's module repository, https://github.com/watchdog-wms/watchdog-wms-modules. Accessed 23 April 2020.
20. Watchdog's workflow repository, https://github.com/watchdog-wms/watchdog-wms-workflows. Accessed 23 April 2020.
21. Watchdog's module reference book, https://watchdog-wms.github.io/watchdog-wms-modules. Accessed 23 April 2020.
22. Gao Y, Zhang J, Zhao F. Circular RNA identification based on multiple seed matching. Brief Bioinform 2018;**19**:803–10.
23. Westholm JO, Miura P, Olson S, et al. Genome-wide analysis of *Drosophila* circular RNAs reveals their structural and sequence properties and age-dependent neural accumulation. Cell Rep 2014;**9**:1966–80.
24. Guo Y, Mahony S, Gifford DK. High resolution genome wide binding event finding and motif discovery reveals transcription factor spatial binding constraints. PLoS Comput Biol 2012;**8**(8):e1002638.
25. Yu G, Wang LG, He QY. ChIPseeker: An R/Bioconductor package for ChIP peak annotation, comparison and visualization. Bioinformatics 2015;**31**(14):2382–3.
26. Leinonen R, Sugawara H, Shumway M, et al. The Sequence Read Archive. Nucleic Acids Res 2011;**39**:D19–21.
27. Kim D, Paggi JM, Park C, et al. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat Biotechnol 2019;**37**:907–15.
28. Galaxy Tool Shed. https://toolshed.g2.bx.psu.edu. Accessed 11 November 2019.
29. KNIME Hub. https://hub.knime.com. Accessed 11 November 2019.
30. NodePit. https://nodepit.com. Accessed 11 November 2019.
31. SnakeMake Wrappers repository. https://bitbucket.org/snakemake/snakemake-wrappers. Accessed 11 November 2019.
32. SnakeMake Workflows repository. https://github.com/snakemake-workflows. Accessed 11 November 2019.
33. nf-core. https://nf-co.re. Accessed 11 November 2019.
34. Hastreiter M, Jeske T, Hoser J, et al. KNIME4NGS: A comprehensive toolbox for next generation sequencing analysis. Bioinformatics 2017;**33**:1565–7.
35. Kluge M, Friedl MS, Menzel AL, et al. Supporting data for "Watchdog 2.0: New developments for reusability, reproducibility, and workflow execution." GigaScience Database 2020; https://dx.doi.org/10.5524/100758.

700–715 *Nucleic Acids Research, 2019, Vol. 47, No. 2* *Published online 22 November 2018*
doi: 10.1093/nar/gky1148

MIR sequences recruit zinc finger protein ZNF768 to expressed genes

Michaela Rohrmoser[1], Michael Kluge[2], Yousra Yahia[3], Anita Gruber-Eber[1], Muhammad Ahmad Maqbool[3], Ignasi Forné[4], Stefan Krebs[5], Helmut Blum[5], Ann Katrin Greifenberg[6], Matthias Geyer[6], Nicolas Descostes[7,8], Axel Imhof[4], Jean-Christophe Andrau[3], Caroline C. Friedel[2] and Dirk Eick[1,*]

[1]Department of Molecular Epigenetics, Helmholtz Center Munich and Center for Integrated Protein Science Munich (CIPSM), Marchioninistrasse 25, 81377 Munich, Germany, [2]Institute for Informatics, Ludwig-Maximilians-Universität München, Amalienstrasse 17, 80333 Munich, Germany, [3]Institut de Génétique Moléculaire de Montpellier (IGMM), Univ Montpellier, CNRS-UMR5535, Montpellier, France, [4]Biomedical Center Munich, ZFP, Großhadener Strasse 9, 82152 Planegg-Martinsried, Germany, [5]Laboratory for Functional Genome Analysis (LAFUGA) at the Gene Center, Ludwig-Maximilians-Universität München, Feodor-Lynen-Strasse 25, 81377 Munich, Germany, [6]Institute of Structural Biology, University of Bonn, Sigmund-Freud-Str. 25, 53127 Bonn, Germany, [7]Department of Biochemistry and Molecular Pharmacology, New York University Langone School of Medicine, New York, NY 10016, USA and [8]Howard Hughes Medical Institute, New York University Langone School of Medicine, New York, NY 10016, USA

Received July 06, 2018; Revised October 25, 2018; Editorial Decision October 29, 2018; Accepted October 29, 2018

ABSTRACT

Mammalian-wide interspersed repeats (MIRs) are retrotransposed elements of mammalian genomes. Here, we report the specific binding of zinc finger protein ZNF768 to the sequence motif GCTGTGTG (N_{20}) CCTCTCTG in the core region of MIRs. ZNF768 binding is preferentially associated with euchromatin and promoter regions of genes. Binding was observed for genes expressed in a cell type-specific manner in human B cell line Raji and osteosarcoma U2OS cells. Mass spectrometric analysis revealed binding of ZNF768 to Elongator components Elp1, Elp2 and Elp3 and other nuclear factors. The N-terminus of ZNF768 contains a heptad repeat array structurally related to the C-terminal domain (CTD) of RNA polymerase II. This array evolved in placental animals but not marsupials and monotreme species, displays species-specific length variations, and possibly fulfills CTD related functions in gene regulation. We propose that the evolution of MIRs and ZNF768 has extended the repertoire of gene regulatory mechanisms in mammals and that ZNF768 binding is associated with cell type-specific gene expression.

INTRODUCTION

Approximately half of mammalian genomes is of repetitive nature and composed of long (LINE) and short interspersed sequences (SINE) (1,2). Mammalian-wide interspersed repeats (MIRs) are an ancient family of retrotransposed SINEs that spread genome-wide before and during mammalian radiation (3,4). MIRs are ∼240 bp long and consist of tRNA-derived sequences, a 70 bp MIR-specific core region, and sequences similar to the 3′ ends of LINEs. MIRs are enriched at gene loci in euchromatin, harbor putative transcription-factor binding sites, provide insulator and enhancer function (5–8), encode microRNAs, are transcribed by RNA polymerase III (9,10), are associated with tissue-specific gene expression (5,11), and sometimes provide splicing signals and contribute to exonization (12). MIRs constitute 5–16% of the genome in marsupials and monotremes and 0.5-3% in placentalia (13). Like other transposable elements, MIRs have shaped gene regulatory networks in vertebrates (14–17), but our understanding how MIRs regulate gene activity is still elusive.

Similarly to MIRs, the family of zinc finger proteins (ZNFs) strongly expanded in mammals (18,19). Widespread binding of ZNFs to regulatory regions indicates that mammalian genomes contain an extensive ZNF regulatory network that targets a diverse range of genes and pathways (20,21). Zinc finger protein 768 (ZNF768) evolved in mammals and is defined by a domain of ten zinc fingers with >96% (Figure 1) identity in placentals and marsupials, but is less conserved in monotremes (Supplementary Figure S1). Placentalia additionally evolved an array of 10–20 heptad repeats in the amino-terminus of ZNF768, which is absent in marsupials and monotremes. This array has a striking similarity to the carboxy-terminal domain (CTD)

*To whom correspondence should be addressed. Tel: +49 89 3187 1512; Email: eick@helmholtz-muenchen.de

Nucleic Acids Research, 2019, Vol. 47, No. 2 701

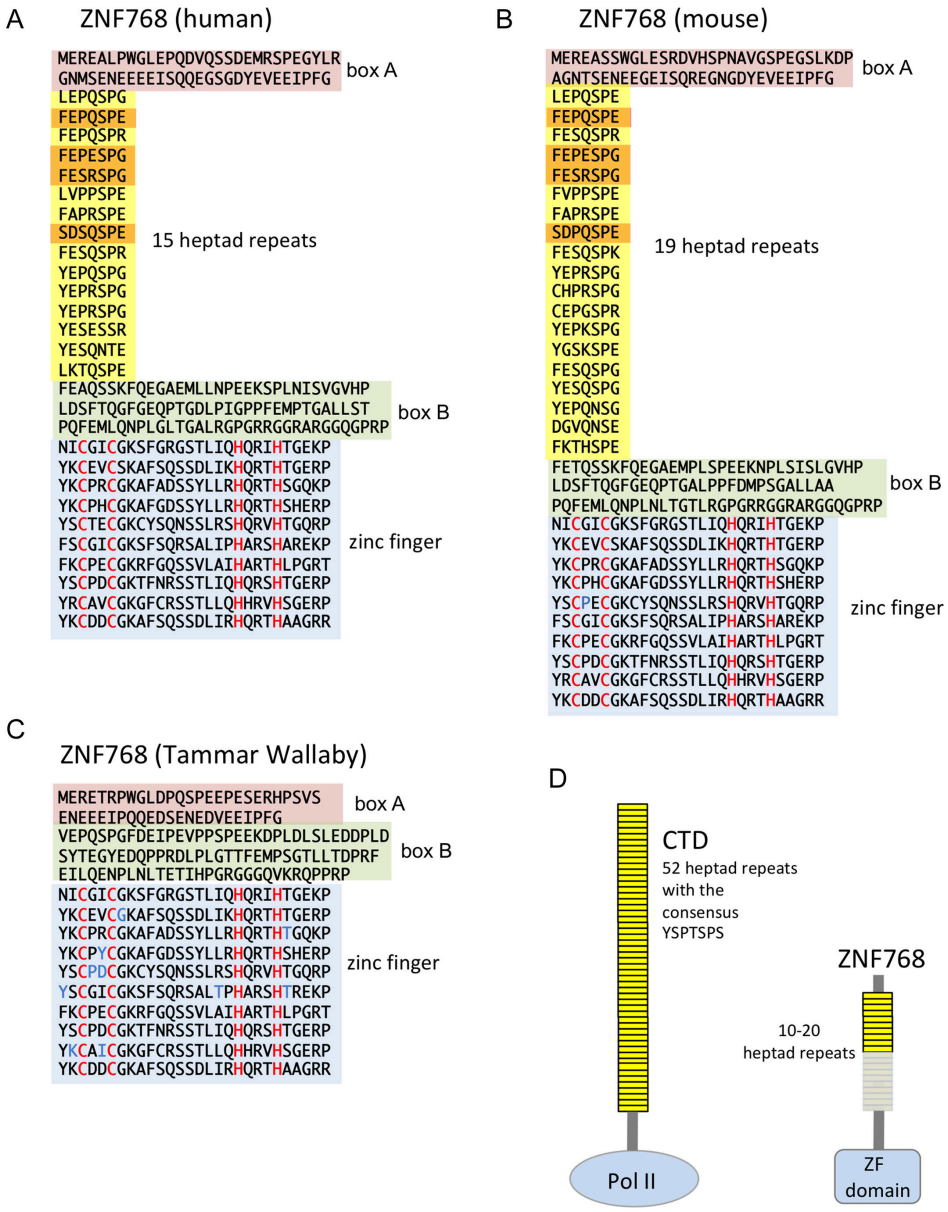


Figure 1. Domain structure of ZNF768 in placentalia and marsupials and comparison with the CTD of RNA polymerase II. (**A**) Human ZNF768 is composed of domains box A (red box) and box B (green box) at the N-terminus interrupted by an array of 15 heptad repeats (yellow box) and a domain of 10 zinc fingers at the C-terminus (blue box). (**B**) Mouse ZNF768 evolved an array of 19 heptad repeats. (**C**) ZNF768 of the marsupial Tammar Wallaby contains conserved A, B, and zinc finger domains, while the array of heptad repeats is absent. (**D**) Number of heptad repeats in RNA polymerase II in vertebrates and ZNF768 in placentalia (see also Supplementary Figure S1).

of the large subunit (Rpb1) of RNA polymerase II (Pol II), which is composed of 52 heptad repeats with the consensus sequence $Y_1S_2P_3T_4S_5P_6S_7$.

The CTD functions as a platform for recruitment and dissociation of cellular factors to the transcription machinery and is mainly regulated during the transcription cycle by phosphorylation of heptad repeats by various kinases (22–26). It is required for initiation, elongation, and termination of transcription, but also for capping, splicing, and 3′processing of the nascent transcript. Interestingly, CTD can function as transcriptional activator after fusion to a GAL4 DNA binding domain (27). Furthermore, transition of Pol II through the transcription cycle is also observed if CTD is fused to other subunits of Pol II (28). Recent reports further provide evidence that CTD of Pol II can aggregate reversibly alone, or with low complexity domains of other transcription factors, like FUS, and that the ability for

702 Nucleic Acids Research, 2019, Vol. 47, No. 2

phase separation in liquid droplets is an important feature for the regulation of transcriptional activity (29–32).

Due to the striking similarity of the heptad repeat array in ZNF768 with the array of heptad repeats in CTD of Pol II we investigated if ZNF768 can act as a transcription factor and fulfill gene regulatory functions in cells.

MATERIALS AND METHODS

Tissue culture and recombinant gene expression

U2OS osteosarcoma cells were cultured in Dulbecco's modified Eagle's medium (DMEM, Gibco) and Raji B-cells in RPMI 1640 medium (Gibco) supplemented with 10% fetal calf serum (FCS, Bio&Sell), 2 mM L-glutamine (Gibco), 100 U/ml penicillin (Gibco), and 100 μg/ml streptomycin (Gibco) at 37°C at 8% or 5% CO_2, respectively. Stably transfected U2OS cell lines were generated with the expression vector pRTS-1 (33) using Polyfect (QIAGEN) followed by hygromycin B (200 μg/ml) selection. Conditional gene expression was induced with 1 μg/ml doxycycline. Recombinant ZNF768 and mutants are tagged C-terminally by a hemagglutinin (HA) tag and synthesized with an optimized codon usage (Gene Art, Regensburg). Details for cloning in pRTS has been described elsewhere (34). All plasmids were confirmed by DNA sequencing prior to expression.

Monoclonal antibody

The generation of monoclonal antibodies has been described previously (34). The ZNF768-specific peptide RSPESDSQSPEFESQSPRYEPQSPGYEPRSPG (synthesized by PSL GmbH, Heidelberg) was coupled to ovalbumin for immunization. The rat monoclonal antibody 7D6 used in this study (IgG2c) specifically recognizes human ZNF768.

Immunoprecipitation (IP) and SDS-PAGE

Cells were washed twice with cold phosphate-buffered saline (PBS) and lysis was performed in 100 μl lysis buffer per 2×10^6 cells (50 mM Tris–HCl, pH 8.0, 150 mM NaCl, 1% NP-40 (Roche), 1x PhosSTOP (Roche), 1× protease inhibitor cocktail (Roche)) at 4°C for 30 min, followed by sonication on ice using a BRANSON Sonifier 250 (15 s on, 15 s off, 50% duty) and centrifugation at 16 400 rpm (FA-45-24-11 rotor) for 10 min at 4°C. Immunoprecipitation was performed using Dynabeads® Protein A und G (1:1) (Invitrogen). Lysates were incubated with antibody-coupled beads (2.5 μg of antibodies for 4 h at 4°C, followed by three washes with 1 ml lysis buffer) overnight. Beads were washed three times with 1 ml lysis buffer and boiled in laemmli buffer (2% SDS, 10% glycerol, 60 mM Tris–HCl, pH 6.8, 10 mM EDTA, 1 mM PMSF, 100 mM DTT, 0.01% bromophenol blue) for SDS-PAGE. Whole cell lysates or IP samples were resolved by SDS-PAGE (10% or 15%) and transferred onto nitrocellulose transfer membranes (GE Healthcare). The membrane was blocked with 5% milk/TBS-T for 1 h. Incubation with primary antibodies was performed over night at 4°C, followed by incubation with HRP-conjugated secondary antibodies for 1 h and chemiluminescence detection with ECL (GE Healthcare).

Immunofluorescence microscopy

U2OS cells were seeded on a coverslip and grown for 24 h. Cells were washed with PBS and fixed with 2% paraformaldehyde (PFA) at RT for 5 min. After permeabilization with 0.15% TritonX-100, samples were blocked with 1% BSA and incubated with 7D6 or HA-specific mAbs over night at 4°C. Samples were washed with PBS for 5 min at RT, 0.15% Triton X-100 for 10 min at RT, blocked with 1% BSA for 7 min and incubated with Cy5-conjugated donkey anti-rat immunoglobulin (Dianova) in the dark for 45 min. Cells were washed again, stained with 4′,6-diamidino-2-phenylindole (DAPI) (Sigma) and mounted on slides using fluorescent mounting medium (Dako). Confocal microscopy was performed on a Leica LSCM SP2 fluorescence microscope using the objective HCX PL APO 63× 1.4. Images were processed using ImageJ 1.37 V and Fuji software and the plug-in RGB profiler. Scale bars were calculated as follows:

$B \times 5\ \mu m / P$ (B = picture length in μm, P = (512 pixel × voxel size) in μm)

siRNA transfection

siRNA transfection was performed according to the manufacturer's protocol using HS_ZNF768_1 FlexiTube siRNA (Qiagen) and the HiPerFect Transfection Reagent (Qiagen) with the exception that transfection was repeated after 24 h. Negative (non-silencing) siRNA (Qiagen) was used as control.

Cell proliferation assay

Cell proliferation was determined using the Real-time xCELLigence System (Roche). U2OS cells were seeded at a density of 3.000 cells per 100 μl in equilibrated 96-well microtiter xCELLigence assay plates (E-plates). Conditional gene expression was induced with 1 μg/ml doxycycline at the indicated time points. Alternatively, siRNA transfection was performed according to the manufacturer's protocol.

Purification of ZNF768

Expression plasmids of human ZNF768 (UniProt accession number Q9H5H4) were cloned from a synthetic gene that was codon optimized for expression in *Escherichia coli* cells (Gene Art, Regensburg). A full length ZNF768 (1–540) construct and a construct consisting of the N-terminal heptad-repeats only (1–197) were cloned by PCR with restrictions sites NcoI/EcoRI and ligated into a pGEX-4T1 vector modified with a TEV protease cleavage site. All plasmids were confirmed by DNA sequencing prior to expression.

Plasmids were transformed into *E. coli* BL21(DE3) cells and induced at an OD_{600} of 0.6 to 1.0 with 0.3 mM IPTG for 4 h growth. Cells were harvested in lysis buffer (20 mM Tris–HCl pH 8.0, 500 mM NaCl, 10% glycerol, 1 mM DTE) and lysed by ultrasound. Fusion proteins were isolated with GSH Sepharose FastFlow (GE Healthcare) affinity chromatography methods. Cleavage of the GST-tag was achieved by adding TEV protease in a 1/50 ratio and was

Nucleic Acids Research, 2019, Vol. 47, No. 2 703

performed for 20 h at 4°C. Protein solution was concentrated and loaded on a preparative HiLoad 16/60 Superdex 200 prep grade gel filtration column (GE Healthcare) for full length ZNF768 or on a HiLoad 16/60 Superdex 75 column for the truncated version of ZNF768 (1–197), respectively, and equilibrated in gel filtration buffer (50 mM HEPES pH 7.5, 150 mM NaCl and 1 mM TCEP). Fractions of the peak containing ZNF768 proteins determined by SDS PAGE analysis were pooled and concentrated to 5 mg/ml. The protein was aliquoted, snap frozen in liquid nitrogen and stored at −80°C.

Gel shift assay

Gel shift assay was performed according to the manufacturer's protocol using the DIG Gel Shift Kit, second generation (Sigma-Aldrich). Briefly, oligonucleotides were annealed to equimolar amounts of their complementary strands (M1: 5′- CAGTGCTGTGTGAC CTTGGGCAAGTCACTTAACCTCTCTGCAGT-3′, M2: 5′- CAGTGCTGTGTGCAGTCAGTCAGT CAGTCAGTCCTCTCTGCAGT-3′ and M3: 5′- CAGTCAGTTGTGACCTTGGGCAAGTCACTT AACCTCCAGTCAGT-3′) by heating to 95°C for 5 min and cooling slowly to room temperature. Double-stranded oligonucleotide probes were labelled at the 3′ end using DIG-11-dUTP and terminal transferase. Binding reactions were performed in 20 μl volumes containing binding buffer [20 mM HEPES, pH 7.6, 1 mM EDTA, 10 mM $(NH4)_2SO_4$, 5 mM DTT, 0.2% (w/v) Tween 20, 30 mM KCl], 50 ng/μl Poly [d(I-C)] and 5 ng/μl Poly L-lysine at room temperature for 15 min. 0.6 ng of DIG-labelled DNA and extract of 1.5–15 μg purified ZNF768-WT or ZNF768 1–197 was used. For competition experiments, unlabeled competitor DNA was added in excess. Protein-DNA-complexes were separated by a native 6% (w/v) polyacrylamide 0.5× TBE gel, transferred onto a positively charged Nylon membrane (GE Healthcare), fixed by Stratagene cross-linker and detected by chemiluminescent substrate CSPD (Roche).

Chromatin immunoprecipitation for ChIP-seq

Cells were crosslinked using a formaldehyde containing solution (10 mM NaCl, 0.1 mM EDTA pH 8.0, 0.05 mM EGTA pH 8.0, 5 mM HEPES pH 7.8 and 1% formaldehyde) for 10 min at 20°C, the reaction was quenched by the addition of glycin to a final concentration of 250 μM for 5 min. Crosslinked cells were collected and washed twice with PBS before snap freezing in liquid nitrogen and storage at −80°C until subsequent use.

Prior to sonication, the crosslinked cells were resuspended in lysis buffer (50 mM HEPES pH 7.5, 140 mM NaCl, 1 mM EDTA pH 8.0, 10% glycerol, 0.75% NP-40, 0.25% Triton X-100, 1× protease inhibitor cocktail) at 4°C for 20 min. Nuclei were collected by centrifugation and washed in a second buffer (200 mM NaCl, 1 mM EDTA pH 8.0, 0.5 mM EGTA pH 8.0, 10 mM Tris pH 8.0, 1× protease inhibitor cocktail) for 10 min at 4°C then collected by centrifugation and resuspended in the shearing buffer (1 mM EDTA pH 8.0, 0.5 mM EGTA pH 8.0, 10 mM Tris pH 8.0, 100 mM NaCl, 0.1% Na-Deoxycholate, 0.5% *N*-lauroylsarcosine, 1× protease inhibitor cocktail). Sonication was carried out in a Bioruptor Pico ultrasounds water bath (Diagenode B01060001) for 30 cycles of 30 s ON and 30 s OFF pulses in 4°C water. Sonicated extracts were centrifuged at high speed in the presence of 0.1% of Triton X-100 and snap frozen in liquid nitrogen and then stored at −80°C until subsequent use.

Prior to ChIP, the ZNF768 mAb was coupled to protein-G coated magnetic beads (Dynabeads, life technologies) by incubation in 0.5% BSA PBS overnight at 4°C. Pre-coated beads were then washed and incubated with the sonicated chromatin extracts. ChIP was carried out overnight at 4°C on a rotating wheel. The equivalent of 10×10^7 cells sonicated extract was used for each ChIP experiment for both cell lines. After incubation, the beads were washed 7× with wash buffer (50 mM HEPES pH 7.6, 500 mM LiCl, 1 mM EDTA pH 8.0, 1% NP-40, 0.7% Na-Deoxycholate, 1× protease inhibitor cocktail) followed by one wash with TE-NaCl buffer (10 mM Tris pH 8.0, 1 mM EDTA pH 8.0, 50 mM NaCl). Immunoprecipitated chromatin was eluted by two sequential incubations with 100 μl elution buffer (50 mM Tris pH 8.0, 10 mM EDTA pH 8.0, 1% SDS) at 65°C for 15 min. The two eluates were pooled and incubated at 65°C for 12 h to reverse-crosslink of chromatin, followed by treatment with RNase A (0.2 μg/ml) at 37°C for 2 h and proteinase K (0.2 μg/ml) at 55°C for 2 h. The DNA was isolated by phenol:chloroform:isoamylalcohol (25:24:1 pH 8.0) extraction followed by Qiaquick PCR Purification (Qiagen, Germany) and quantified with Qubit DS DNA HS Assay (ThermoFisher Scientific, USA).

At least 1 ng of ChIP DNA was used to prepare sequencing library with Illumina ChIP Sample Library Prep Kit (Illumina, USA) with a few optimizations to the protocol. The ChIP DNA was size selected using Ampure beads (Life technologies) to enrich for fragments <400 bp prior to end-repair, 3′end adenylation and adapter ligation. Library fragments were then directly amplified by 10 cycles of PCR. Barcoded libraries from different samples were pooled together and sequenced on Illumina HiSeq2000 platform in paired-end sequencing runs.

RNA-seq libraries

For preparation of total RNA cells (0.9 Mio/ml) were harvested and resuspended in TRIzol reagent (Life Technologies) and snap-frozen in liquid nitrogen. After thawing RNA was extracted from 0.4 ml of TriZol lysate using the direct-zol RNA Miniprep (Zymo Research, Irvine CA, USA) as described in the manufacturer's protocol. RNA was assessed for purity by UV–vis spectrometry (Nanodrop) and for integrity by Bioanalyzer (Agilent Bioanalyzer 2100, Agilent, Santa Clara USA). RNA was of high purity (abs. 260/280 > 1.9, abs. 269/239 > 2.1) and integrity (Bioanalyzer RIN > 9) and thus used for further processing. For production of RNA-seq libraries total RNA was DNAse treated (dsDNAse, Fermentas) and 100 ng of this RNA was processed with a strand-specific protocol (RNA-seq complete kit, NuGEN, San Carlos, USA). In brief the RNA was reverse transcribed to cDNA with a reduced set of hexamer primers, avoiding excessive representation of rRNA in the cDNA. Second strand cDNA synthesis

was done in presence of dUTP. After ultrasonic fragmentation of the cDNA and end repair, Illumina-compatible adapter were ligated. Adapters contained uracil in one strand, allowing complete digestion of the second-strand derived DNA. After strand selection the libraries were amplified, assessed for correct insert size on the Agilent Bioanalyser and diluted to 10 nM. Barcoded libraries were mixed in equimolar amounts and sequenced on an Illumina HiSeq1500 in single-read mode with a read length of 100 bp.

Deep sequencing

ChIP-seq and RNA-seq analysis was performed as previously described (35). Four biological replicates of U2OS and Raji cells were used for RNA-seq library construction.

ChIP-seq data processing

Raw sequencing reads were aligned to the human genome (hg38) using BWA (36). Sequence reads with an alignment score <30 for paired-end reads and <20 for single-end reads were discarded as well as all reads that aligned equally well to different positions in the genome. Peak calling was performed using GEM (37) in GPS mode and with a *q*-value cutoff of 0.01. Overlapping peaks (peak centers: ±100 bp) were merged both within samples and across all four samples to obtain the final list of unique peaks. Motif discovery was performed using MEME-ChIP (38) and sequence logos of the binding motif and surrounding regions were created using weblogo (39). Annotation of peaks relative to gene features was performed using the ChipSeeker package in R (40). Gene annotations were taken from GENCODE version 25 (41). Repeat annotations by RepeatMasker and phyloP100 conservation scores (42) for hg38 were downloaded from the UCSC genome browser. Visualization of ZNF768 binding on the genome and corresponding peaks was performed using Gviz (43). For the analysis of binding frequencies of ZNF768 in promoter, UTR, exon and intron regions, the same number of 200 bp regions were randomly selected using BEDtools (44).

RNA-seq data processing

Quality check of sequencing reads was performed using FastQC (available at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc). Sequencing reads were mapped against the human genome (hg38) and human rRNA sequences using ContextMap version 2.7.9 (45) (using BWA as short read aligner and default parameters). Number of read fragments per gene were determined from mapped RNA-seq reads in a strand-specific manner using featureCounts (46) and GENCODE version 25 gene annotations. RPKM values were calculated using edgeR and averaged between replicates (47). Differential gene expression analysis was performed using limma (48). Functional enrichment analysis for UniProt keywords and Gene Ontology terms was performed with the DAVID webserver (49). Significantly enriched terms were determined using a cutoff of 0.05 on the *P*-value adjusted for multiple testing using the method by Benjamini and Hochberg (50). Analysis workflows were implemented and run using the workflow management system Watchdog (51).

Purification of ZNF768 for mass spectrometric analysis

For purification of ZNF768, Raji or U2OS cells (3×10^8) were collected and IP was performed in 3 biological replicates as described in the respective paragraph of immunoprecipitation. Simultaneously, αZNF768 antibody (7D6) and αPes1 (8E9) antibody, respectively, was coupled to Sepharose A and G beads for 4 h at 4°C. αZNF768 antibody (7D6) was used to identify the interactome of ZNF768 whereas αPes1 (8E9) antibody served as a subclass control (52,53).

On beads digestion

After the last washing step with lysis buffer, beads were washed three times by adding 100 μl of 50 mM NH_4HCO_3. For trypsin digest, beads were transferred to a clean tube and incubated with 100 μl of 10 ng/μl trypsin-solution in 1M urea and 50 mM NH_4HCO_3 for 30 min at 25°C. Samples were centrifuged at 800 rpm and supernatant was transferred into a fresh tube. Beads were washed twice with 50 μl of 50 mM NH_4HCO_3. The supernatants were pooled into the corresponding tube and incubated overnight at 25°C after addition of 1mM DTT. Iodoacetamide (IAA) 10 μl (5mg/ml) was added and incubated for 30 min in the dark at 25°C. To quench the IAA, 1 μl of 1M DTT was added and samples were incubated for 10 min at 25°C, followed by addition of 2.5 μl of trifluoroacetic acid (TFA) and desalting using 2× C18 Stagetips (54). Stagetips were washed three times with 20 μl of 100% ACN (1000 rpm, 1 min) and three times by adding 20 μl of 0.1% TFA (1800 rpm, 1 min). Subsequently, samples were added (800 rpm, 30 min) and washed 3 times with 20 μl of 0.1% TFA, followed by elution into a clean tube by washing three times with 20 μl of 80% ACN/25% TFA solution. Finally, samples were evaporated to dryness, resuspended in 20 μl formic acid solution and stored at −20°C until LC–MS analysis.

Protein quantification by liquid chromatography coupled to tandem mass spectrometry (LC–MS/MS)

Purified peptides (5μl) were automatically injected into in an Ultimate 3000 RSLC HPLC system (Dionex Thermo), separated on an analytical column C18 micro column (75 μm i.d. × 15 cm, packed in-house with Reprosil Pur C18 AQ 2.4 μm, Doctor Maisch) using a 50-min gradient from 5 to 60% acetonitrile in 0.1% formic acid. The effluent from the HPLC was subsequently electrosprayed into a LTQ Orbitrap XL mass spectrometer (Thermo). The MS instrument was operated in a data dependent mode to automatically switch between full scan MS and MS/MS acquisition. Survey full scan MS spectra (from m/z 300 to 1800) were acquired in the Orbitrap with a resolution of $R = 60\,000$ at m/z 400 (after accumulation to a 'target value' of 500,000 in the linear ion trap). The six most intense peptide ions with charge state between 2 and 4 were sequentially isolated to a target value of 10,000 and fragmented in the linear ion trap by collision induced dissociation (CID). For all measurements with the Orbitrap mass analyzer, three lock-mass ions from ambient air ($m/z = 371.10123$, 445.12002, 519.13882) were used for internal. Usual MS conditions were: spray voltage, 1.5 kV; no sheath and

Nucleic Acids Research, 2019, Vol. 47, No. 2 705

auxiliary gas flow; heated capillary temperature, 200°C; normalized collision energy 35% for CID in LTQ. The threshold for ion selection was 10 000 counts for MS2. The used activation was 0.25 and activation time 30 ms. MaxQuant 1.5.2.8 was used to identify proteins and quantify by iBAQ with the following parameters: Database, Uniprot_Hsapiens_3AUP000005640_170526; MS tol, 10ppm; MS/MS tol, 0.5 Da; Peptide FDR, 0.1; protein FDR, 0.01 Min. peptide length, 5; variable modifications, oxidation (M); fixed modifications, carbamidomethyl (C); peptides for protein quantitation, razor and unique; min. peptides, 1; min. ratio count, 2. Identified proteins were considered as interaction partners if their MaxQuant iBAQ values displayed a greater value than $\log_2$ 5-fold enrichment (FC) and *P*-value 0.05 (*t*-test adjusted for multiple comparisons) when compared to the control.

RESULTS

ZNF768 domain structure and conservation

An array of ten zinc fingers at the C-terminus of ZNF768 shows high conservation in placentalia and marsupials (>96%) but is less conserved in monotremes (blue boxes in Figure 1 and Supplementary Figure S1). At the N-terminus, two sequence blocks (box A and box B, red and green boxes in Figure 1) are conserved in placentalia and marsupials, but replaced by unrelated sequences in monotremes. In addition, ZNF768 of placentalia has evolved an array of heptad repeats that is positioned between box A and box B. The number of repeats varies between 20 repeats in mouse lemur and 10 repeats in pika and malayan pangolin. Mouse ZNF768 contains 19 repeats, chimpanzee 16 and human 15 repeats (Supplementary Figure S1). Similar to heptad repeats in CTD of Pol II, the heptad repeats in ZNF768 show no length variation (except a single extended repeat in pika). However, the composition of amino acids in the heptad repeats shows higher variation in ZNF768, both, within and between species (Supplementary Figures S1 and S2B). Serine-5 and proline-6 residues show the highest conservation between ZNF768 and Pol II, followed by the residues corresponding to tyrosine-1 and proline-3 in the CTD, the position of threonine-4 and serine-7 show only little or almost no conservation. The position corresponding to serine-2 in the CTD is particularly remarkable in ZNF768. It is replaced in almost all repeats by an acidic amino acid (mostly glutamic acid). The phosphorylation of serine-2 residues in CTD by P-TEFb is a hallmark in RNA elongation control (55) and a replacement of serine-2 may mimic its phoyphorylation. It is thus tempting to speculate that the array of heptad repeats in ZNF768 potentially can mimic a CTD phosphorylated at serine-2 residues.

ZNF768 is associated with euchromatin and required for growth and cell viability

To study the cellular function of ZNF768 we first raised a monoclonal antibody (7D6) towards human ZNF768 using a peptide containing heptad repeats 8–12 as epitope (Figure 1A, materials and methods). This antibody preferentially stains euchromatic regions in the nucleus of fixed U2OS cells (Figure 2A and Supplementary Figure S3A). Expression of the zinc finger-containing C-terminal domain of ZNF768 (Figure 2E) caused a similar staining pattern, while expression of the N-terminus containing the array of heptad repeats resulted in a more diffuse staining of the nucleus (Supplementary Figure S3B), suggesting that the zinc finger domain is responsible for the association of ZNF768 with euchromatin. Antibody 7D6 immunoprecipitated endogenous ZNF768 protein quantitatively from extracts of osteosarcoma cell line U2OS and B-lymphoid cell line Raji (Figure 2B), proving its high specificity and suitability to study the binding of ZNF768 to DNA in chromatin immunoprecipitation (ChIP) experiments. Knockdown experiments of ZNF768 confirmed the specificity of mAb 7D6 (Figure 2C) and showed further that ZNF768 is required for viability and proliferation of U2OS cells (Figure 2D). In line with its essential function, expression of mutants with deletions of either the N- or C-terminal domain of ZNF768 have a dominant-negative phenotype and inhibit cell proliferation (Figure 2E–G). Finally, ZNF768 is a phosphoprotein and can be phosphorylated at almost all heptad repeat serine-5 residues (www.cellsignal.com, Supplementary Figure S2A). Treatment of cellular extracts of U2OS cells with alkaline phosphatase causes a shift of the hyperphosphorylated form of ZNF768 (Supplementary Figure S2C) and reveals that a large fraction of ZNF768 is hyperphosphorylated in U2OS cells.

Identification of the ZNF768 binding motif in cellular DNA

To investigate if ZNF768 can bind to specific DNA sequences, we performed ChIP experiments with mAb 7D6 using extracts of U2OS and Raji cells. DNA libraries of two biological replicates were prepared for each cell line and analyzed by next generation sequencing. Peak calling identified a total of 21 012 unique peaks and 13.1% of these peaks were consistently identified in all four samples and an additional 28.8% at least in both replicates for the same cell type (Supplementary Figure S4). Generally, ZNF768 binding sites distributed over all chromosomes (Supplementary Figure S5). Motif discovery identified several potential binding motifs for ZNF768 (Supplementary Figure S6A). The top two identified motifs were found in 46% and 37% of peaks, respectively, and for both motifs the other motif was often found as a secondary motif at a distance of ∼20 bp. We thus hypothesized that the ZNF768 binding motif consists of anchor regions connected by a linker region of ∼20 bp. In fact, 58.1% of identified peaks contained this consensus motif with at most three mismatches in the anchor regions and a linker region of 20 ± 3 bp (Figure 3A, Supplementary Figures S4 and S6B). For peaks identified in all 4 samples, this number was as high as 98.3% and the vast majority of peaks with motif hits (83.5%) had a linker length of 20 bp (Supplementary Figure S6B). Gelshift experiments with recombinant ZNF768 protein confirmed the motif, GCTGTGTG (N_{20}) CCTCTCTG, and revealed that the nucleotide sequence of the spacer between the two anchor regions is likely not critical for binding (Supplementary Figure S7).

706 Nucleic Acids Research, 2019, Vol. 47, No. 2

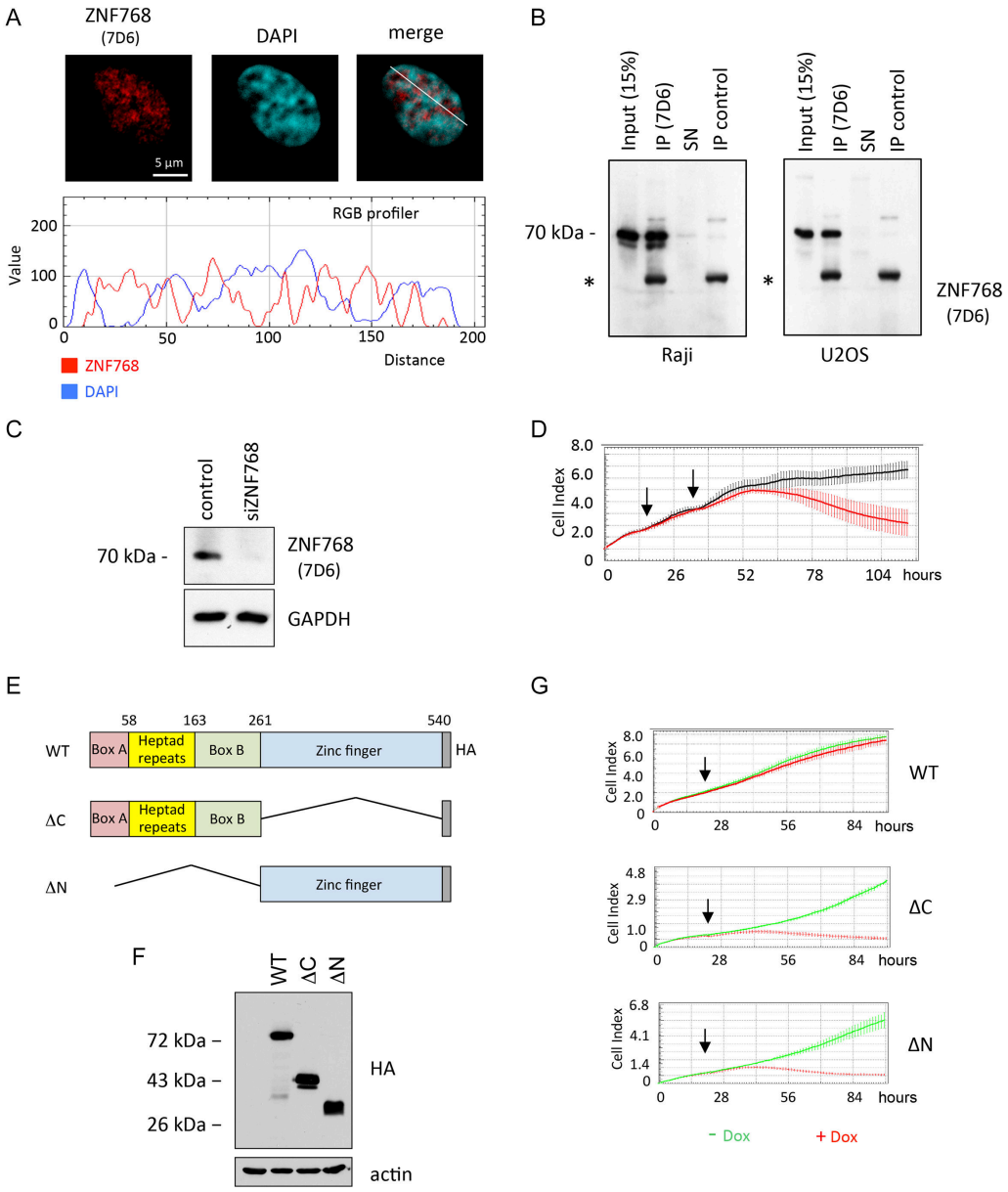


Figure 2. ZNF768 is associated with euchromatin and essential for cell viability and proliferation. (**A**) Confocal image of U2OS cells stained with DAPI and with the ZNF768-specific mAb 7D6; merge of images on the right hand site. White line marks the area of the RGB profiler, profiles of DAPI and ZNF768 at the bottom. (**B**) mAb 7D6 immunoprecipitates a 70 kD protein from extracts of Raji and U2OS cells. SN: supernatant, *: Ig heavy chain. (**C**) siRNA mediated knockdown of ZNF768 in U2OS cells. (**D**) Growth kinetics of U2OS cells after knockdown of ZNF768 measured by xCelligence (Roche). Arrows indicate consecutive addition of siRNA. (**E**) Expression constructs of HA-tagged ZNF768 wild-type and deletion mutants and (**F**) expression control in U2OS cells. (**G**) Growth kinetics of U2OS cells after expression of ZNF768-WT and ZNF768 mutants measured by xCelligence (Roche). Arrows indicate addition of doxycycline.

ZNF768 binds to MIR sequences

A systematic comparison of ZNF768 binding sites to repeats in the human genome showed an enrichment of binding sites within all four types of MIRs (Figure 3B). 12,488/21,012 peaks overlapped with MIRs. Furthermore, almost all peaks (92%) with the binding motif were contained in a MIR sequence and the consensus sequence for all MIR types actually contains the ZNF768 binding motif (Supplementary Figure S8A). Despite this fact, only a small fraction (12.2%) of MIRs in the human genome contains the ZNF768 binding motif, which is not surprising giving a per-base identity <80% for human MIR sequences. Although only 15.8% of MIRs containing the binding mo-

Nucleic Acids Research, 2019, Vol. 47, No. 2 707

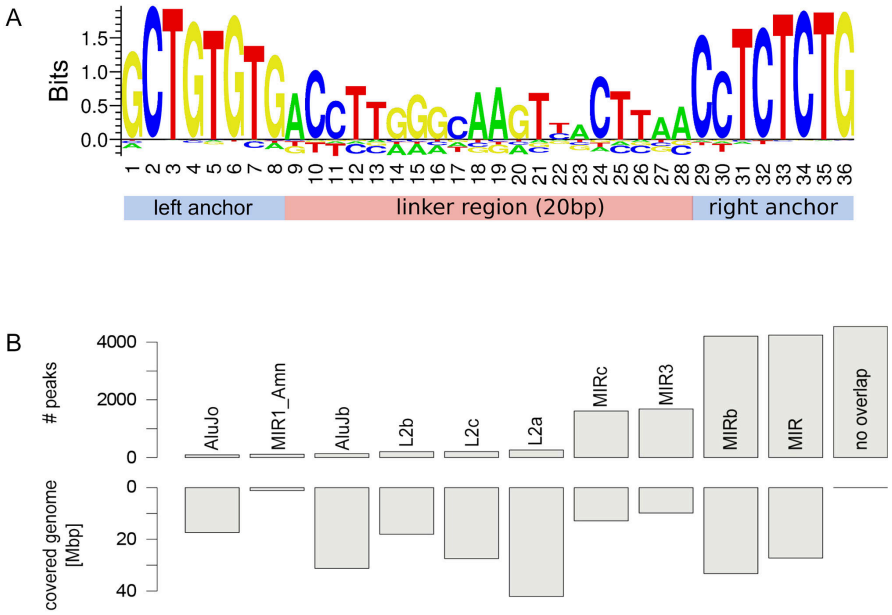


Figure 3. DNA binding motif and genomic binding of ZNF768 to MIRs. (**A**) Consensus ZNF768 binding motif identified by ChIP-seq experiments in Raji and U2OS cells and determined from peaks containing the motif (±3 mismatches in the anchors and linker length of 20 ± 3 bp). (**B**) Number of ZNF768 binding sites overlapping with particular type of repetitive sequences (top; in case of multiple overlaps the largest overlap is used) compared to the genomic length covered by the corresponding type of repetitive sequence (bottom). The right-most bar shows the number of ZNF768 binding sites with no overlap to repetitive sequences.

tif were found to be bound by ZNF768, this fraction increased to 54.2% when considering MIRs with a more stringent and strict version of the motif (linker length: 19/20bp, 1 mismatch in anchors). Thus, most MIRs diverged so far from the consensus that the binding motif was lost and no general conservation of the binding motif in human MIRs was observed (Supplementary Figure S8B). This divergence also allowed reliably aligning reads to MIR sequences despite their repetitive origin. Only reads that could be aligned uniquely to the genome were used for peak calling. Although 8524 detected peaks (40.6%) were not within MIRs, 13.5% of these peaks contained the binding motif. The remaining peaks may contain a weaker version of the binding motif, recruit ZNF768 to chromatin by other mechanisms (e.g. looping), or represent spurious binding.

Interestingly, MIRs with ZNF768 binding show a clear conservation of the two anchor motifs in the human genome. Sequences of the linker in the binding motif and outside of the binding motif were not particularly conserved, similar to MIRs without binding of ZNF768 (Supplementary Figure S8B). We further investigated whether ZNF768 binding sites in MIRs were also conserved across species by analyzing phyloP100 conservation scores determined from a multiple alignment of 99 vertebrate genomes against the human genome. Positive PhyloP scores indicate slower than expected evolution. The analysis of phyloP100 scores within and around the binding motif (±25 bp) in MIR sequences bound by ZNF768 showed increased conservation for most positions within the anchor regions (Figure 4A). Sequences outside of the binding motif or within the linker region, however, were mostly not conserved. Unbound MIR sequences showed no particular conservation (Figure 4B) indicating that ZNF768 binding represents a conserved function of a subset of MIR sequences in mammals.

ZNF768 binding is associated with transcribed genes

We next asked if ZNF768 binding sites were enriched in regulatory elements of genes. We found a strong enrichment of ZNF768 binding in promoters and a slight enrichment of binding in exons and introns, while the binding frequency in intergenic sequences was reduced (Figure 5A). Interestingly, the 1061 ZNF768 binding sites outside of MIRs that contained the binding motif showed an even higher enrichment at promoters. To investigate whether genes with ZNF768 binding tended to be more highly expressed, we analyzed RNA-seq data of four replicates of total RNA of Raji and U2OS cells (Supplementary Table S1). In both cell lines, protein-coding genes with ZNF768 binding in the promoter or 5′UTR were more highly expressed on average than the remaining protein-coding genes (Figure 5B). In contrast, binding in intronic regions showed only a small but significant effect (Figure 5B). This provides evidence that ZNF768 regulates transcription by binding in or near promoter regions of active genes.

ZNF768 binds to genes with cell type-specific expression

Raji and U2OS cells revealed common and cell type-specific binding sites of ZNF768. Common sites were for instance associated with genes for RNA polymerase II subunit E

708 Nucleic Acids Research, 2019, Vol. 47, No. 2

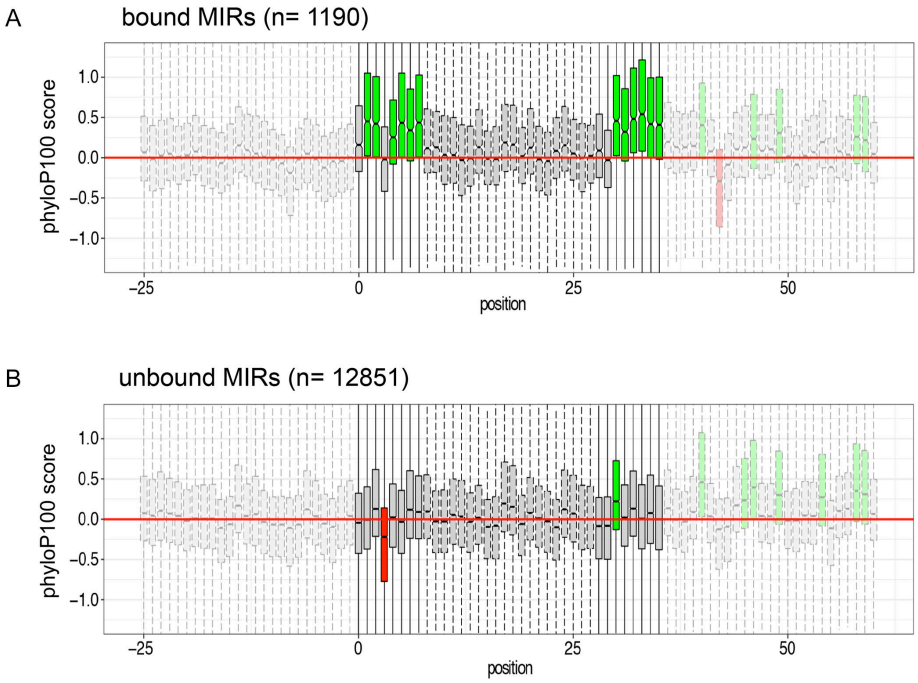


Figure 4. Conservation of the ZNF768 motif (+25 bp on either side) in MIRs either (**A**) bound or (**B**) not bound by ZNF768. Only MIRs were considered that align without gaps to the MIR consensus sequence in the region of the ZNF768 binding motif +25 bp on either side. Distribution of PhyloP100 scores are indicated by boxplots for each position (green = median PhyloP100 score > 0.2) and regions within the motif region are indicated in more intensive colors.

(POLR2E) (Figure 6A) or Solute Carrier Family 1 Member 5 (SLC1A5) and Mitochondrial Ribosomal Protein S5 (MRPS5) (Supplementary Figure S9A,B). These genes are expressed in Raji and U2OS cells and show similar peaks in both cell lines. Thus, many of the 2747 identified common binding sites in Raji and U2OS may be associated with commonly expressed genes. We also identified a large number of peaks that were present either in Raji or U2OS cells. In U2OS cells, strong peaks for ZNF768 were associated with the promoter region of the GAS2L1 gene (Figure 6B), the ID1 and SNPH genes (Supplementary Figure S9C,D) and the gene body of the ANXA2 and ALDH7A1 genes (Supplementary Figure S9E, F), but were absent or only faintly detectable in Raji cells. These genes are expressed in U2OS but not in Raji cells. Inversely, Raji cells showed a strong ZNF768 binding site in the promoter region of the B-Lymphocyte Surface Antigene (CD19) gene (Figure 6C), which is a B cell-specific non-receptor tyrosine kinase required for B cell receptor signaling. Strong Raji-specific peaks were further detected for the genes CD86, ATP2A3, RHOH, PLCG2, LYN, and ARHGDIB (Supplementary Figure S9G–L). These genes are expressed in Raji but not U2OS cells.

A global analysis of differential gene expression between U2OS and Raji cells showed significant differences in fold-changes for genes with peaks specific to either cell line (Figure 6D and Supplementary Table S1). In particular, genes with U2OS-specific peaks were on average 30-fold higher expressed in U2OS cells. For genes with Raji-specific peaks, the fold-changes in gene expression were lower. This may be due to the higher number of peaks identified in Raji cells, indicating a higher sensitivity but lower specificity compared to peaks in U2OS cells. Thus, a significant fraction of seemingly Raji-specific peaks may simply have been missed in U2OS. We conclude that binding of ZNF768 occurs preferentially at expressed genes and at least in part in a cell type-specific manner. The underlying mechanisms regulating the cell type-specific binding of ZNF768 in Raji and U2OS cells are currently unclear, but may involve, e.g. DNA methylation or other epigenetic marks.

ZNF768-regulated genes in U2OS cells

To study the gene regulatory potential of ZNF768, we induced expression of the dominant-negative mutant ZNF768-ΔN (Figure 2E) in U2OS cells and analyzed changes in the transcriptome after 12 h. A >2-fold change in RNA levels was detected for 500 downregulated and 155 upregulated genes (Supplementary Table S2). Functional enrichment analysis of repressed genes revealed several significantly enriched gene sets including two gene sets containing DNA binding proteins (105 genes) and zinc finger proteins (103 genes) (Figure 7A and Supplementary Table S3). Repressed genes in both gene sets show a large overlap (63 genes) with repressed transcription-associated genes (Figure 7B). We conclude that ZNF768 can act as transcriptional regulator and is required particularly for the expression of other transcription factors.

Nucleic Acids Research, 2019, Vol. 47, No. 2 709

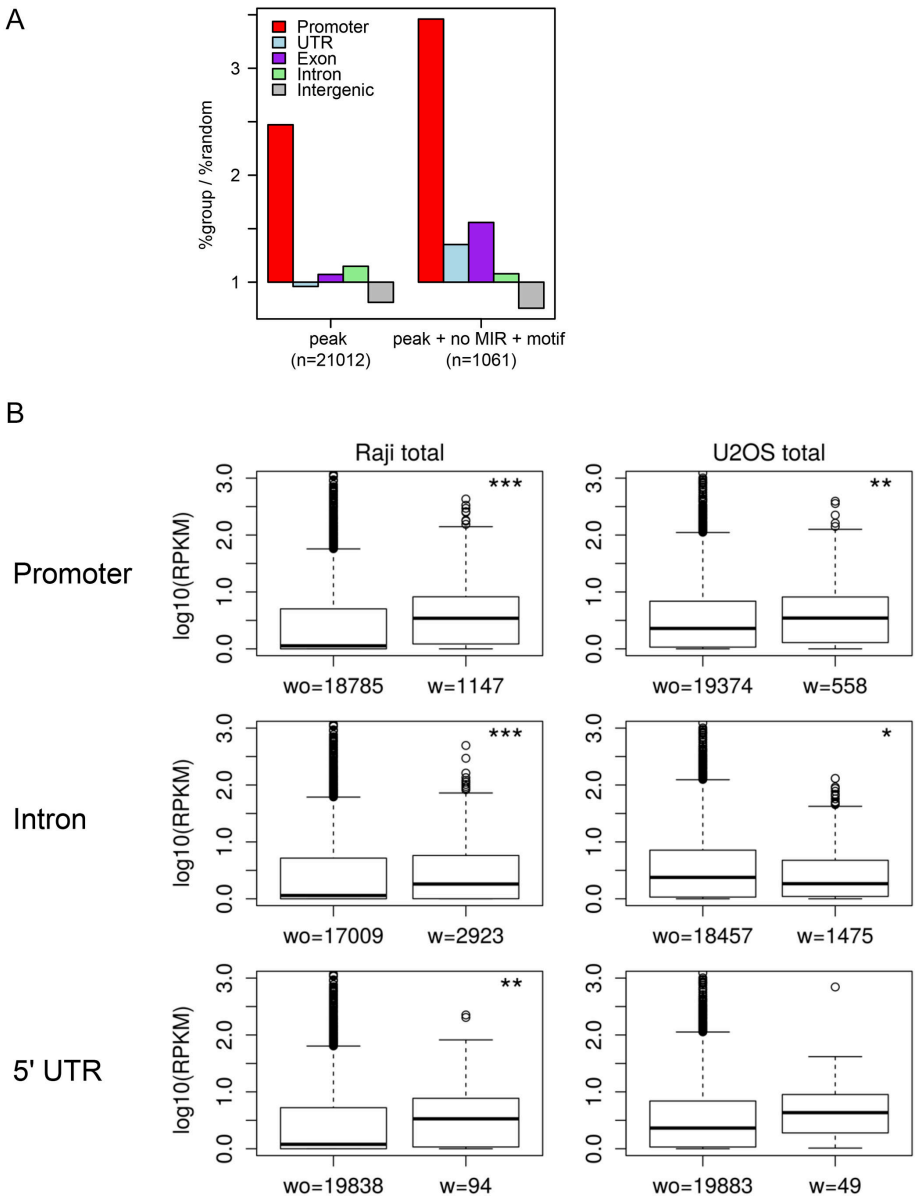


Figure 5. Genomic distribution of the ZNF768 binding motif in Raji and U2OS cells. (**A**) Frequency of ZNF768 binding sites in promoters (–1 kb to transcription start site) and other genomic regions compared to randomly selected binding sites with the same peak length distribution. This shows an enrichment of genomic binding of ZNF768 in promoters, in particular for motif-containing peaks outside of MIRs. (**B**) Boxplots illustrating the distribution of expression levels in total RNA (quantified as RPKM = reads per kilobase per Million mapped reads) in Raji or U2OS cells for genes without (wo) and with (w) peaks in the respective cells. A pseudocount of 1 was added to all RPKM values before plotting. *P*-values for a Wilcoxon rank sum test comparing RPKM levels between the two groups are indicated as: $*P < 10^{-3}$, $**P < 10^{-5}$, $***P < 10^{-10}$

Mass spectrometric analysis of ZNF768 associated factors

We used the mAb 7D6 for a combined immunoprecipitation (IP) and mass spectrometric (MS) assay to identify ZNF768 associated factors. The ZNF768 interactome of Raji and U2OS cells showed a large overlap and twenty of the best thirty interactors were found in both cell lines (Figure 8 A,B, Supplementary Figure S10). Among the common factors we identified three subunits of the Elongator complex (Elp1, Elp2 and Elp3), SR rich splicing factor (SUGP2), centromere protein E (CENPE), several E3 ligases (USP13, Trim33, and HERC2), proteins with centrosomal functions (CEP170-1, Cep170-2 and NIN), and other factors. The binding of Elongator subunit Elp3 to ZNF768 was confirmed in IP experiments with an Elp3-specific antibody (Figure 8C). mAb 7D6 could immunoprecipitate a significant fraction of Elp3 protein of cellular extracts of Raji cells.

710 Nucleic Acids Research, 2019, Vol. 47, No. 2

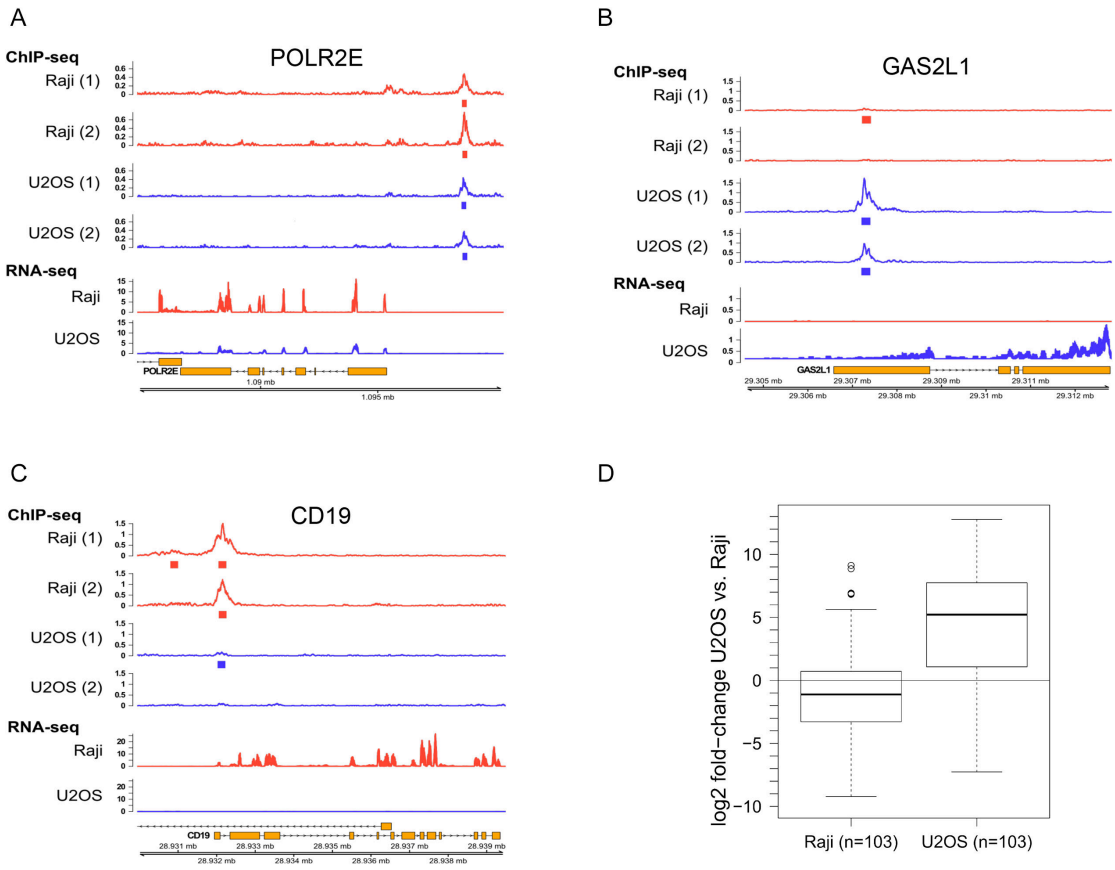


Figure 6. Common and cell type-specific peaks of ZNF768 in Raji and U2OS cells. ChIP-seq (replicates shown separately) and RNA-seq (mean of four replicates) read coverage (in counts per million) for example genes. Identified peaks are shown as rectangles below the corresponding ChIP-seq sample. Genomic coordinates and gene annotation (boxes = exons, lines = introns, strand indicated by arrowheads) are shown in the bottom row. (**A**) ZNF768 binding in the promoter upstream region of the RNA Polymerase II Subunit E (POLR2E) gene, which is expressed in both Raji and U2OS cells. (**B**) Binding of ZNF768 in the promoter region of the Growth Arrest Specific 2-Like (GAS2L1) gene, which is expressed in U2OS but not Raji cells. (**C**) Binding of ZNF768 to the promoter region of the B-Lymphocyte Surface Antigene (CD19) gene, which is expressed in Raji but not U2OS cells. (**D**) Genes in Raji and U2OS cells with cell-specific peaks differ in gene expression. Boxplots illustrate the distribution of fold-changes in gene expression between both cell lines (determined with limma) for genes with cell-specific peaks (= peaks identified in gene body or 1 kb upstream in both replicates of the corresponding cell line but not for the other cell line; to account for the differences in sensitivity between Raji and U2OS ChIP-seq, for Raji only the 103 genes with the top-scoring Raji-specific peaks were evaluated, i.e. the same number of genes as with U2OS-specific peaks). Significance of the difference in median values was determined using the Wilcoxon rank sum test ($***P \leq 10^{-10}$).

The results suggests that ZNF768 can recruit Elongator and other factors to expressed genes in Raji and U2OS cells.

DISCUSSION

ZNF768 binds to MIR sequences

ZNF768 proteins in mammals contain an array of ten zinc fingers that allow the specific binding to DNA. ChIP-seq experiments revealed approximately ten to twenty thousand ZNF768 binding sites in the genome of Raji and U2OS cells. The majority of these sites is contained within MIR sequences and shares a common binding motif that is part of the MIR consensus sequence. The motif of the binding site is 36 bp long and consists of two anchor sequences of 8 bp separated by a linker of 20 bp, which probably does not contribute to the binding specificity of ZNF768 as revealed by gel shift experiments. ZNF768 binds preferentially at or near promoters, suggesting that binding of ZNF768 is associated with gene expression. In agreement with this assumption we observed ZNF768 binding preferentially in euchromatic regions of the nucleus. Likewise, MIR sequences have been reported to be associated with transcriptional active euchromatin but not heterochromatin (5,6). Strikingly, the number of MIR sequences in mammals varies considerably from about 20% of the total genome in monotremes to 1% or 3% of the genome in mice and humans, respectively. Furthermore, the ZNF768 binding motif, although part of the MIR consensus sequence, is not conserved in all MIRs, but only in those displaying a peak in ZNF768 ChIP-seq experiments. Notably, we also detected ∼1000 peaks containing the ZNF768 binding motif outside of MIRs. This category of peaks showed the highest association with promoters.

Given the length of the detected DNA binding motif and the position of the two anchor sequences at its flanks it is

Nucleic Acids Research, 2019, Vol. 47, No. 2 711

A

description	adj. p–value	# genes	% of downregulated genes
DNA–binding	3.9e–09	105	21%
Zinc–finger	7.2e–09	103	21%
Coiled coil	1.2e–08	144	29%
Phosphoprotein	1.5e–08	325	65%
Zinc	5.4e–08	117	23%
Polymorphism	1.2e–07	343	69%
Transcription regulation	5.2e–07	111	22%
Nucleus	5.8e–07	216	43%
Transcription	8.0e–07	113	23%
Metal–binding	4.4e–06	148	30%
Cell division	4.5e–06	37	7.4%
Cell cycle	4.6e–06	51	10%
Mitosis	7.3e–06	29	5.8%
Chromosome	1.1e–05	33	6.6%
ATP–binding	1.5e–05	76	15%
Nucleotide–binding	1.5e–04	87	17%
Helicase	2.2e–03	16	3.2%
Cytoskeleton	4.8e–03	58	12%
DNA repair	5.6e–03	24	4.8%
DNA damage	5.9e–03	27	5.4%
Ubl conjugation	9.2e–03	81	16%
Microtubule	9.8e–03	21	4.2%
TPR repeat	1.7e–02	15	3%

B

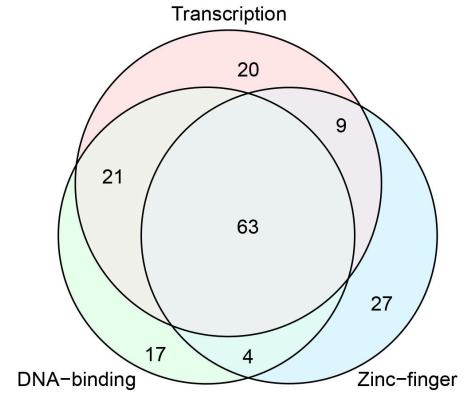


Figure 7. Functional enrichment analysis for UniProt keywords for genes downregulated upon expression of the dominant-negative mutant ZNF768-ΔN in U2OS cells. (**A**) Significantly enriched UniProt keywords (identified with DAVID at an adjusted *P*-value < 0.05) for downregulated genes (>2-fold down-regulated, adjusted *P*-value <0.01, for full details see also Supplementary Table S3). (**B**) Venn diagram of downregulated genes annotated with the keywords Transcription, DNA-binding, and Zinc finger. The data indicates that inhibition of ZNF768 downregulates other transcription factors with zinc finger domains.

likely that the proximal and distal, but not the central zinc fingers, of the array of 10 zinc fingers contribute to DNA binding. The potential function of the central zinc fingers is currently unknown. We have currently no evidence for other conserved motifs upstream or downstream of the ZNF768 binding motif. Notably, we also observed ZNF768 peaks at gene loci that do not contain the DNA binding motif. It is currently unclear if binding to these loci requires the zinc finger domain and/or other parts of the protein.

ZNF768 is an essential gene for cell proliferation

Knockdown experiments as well as the expression of dominant-negative mutants revealed the functional requirement of ZNF768 for cell viability and proliferation. Expression of a mutated form of ZNF768 containing only the C-terminal or N-terminal domain, respectively, led to a decline of the cell index in cell proliferation assays. A decline of this index was also seen after siRNA-mediated knockdown of ZNF768 expression. This indicates that the functional loss of ZNF768 cannot be compensated by other cellular factors. Our results suggest that ZNF768, despite being an evolutionary young gene, gained essential function(s) for the expression of growth related genes. A detailed genetic analysis combined with mass spectrometry experiments will be required in the future to analyze the function of ZNF768 in the context of growth control in more detail.

Cell type-specific binding of ZNF768 to gene loci

ChIP-seq analysis of ZNF768 revealed common but also a large number of differential binding sites in Raji and U2OS cells. Furthermore, many putative binding sites containing the binding motif were not occupied by ZNF768 in either Raji or U2OS cells. This observation suggests that binding of ZNF768 to DNA is regulated and that not all binding sites are equally accessible in Raji and U2OS. The mechanism(s) regulating the different accessibility is currently unknown but may include DNA methylation, histone composition at binding motifs or specific histone marks. Additionally, other cellular factors may block or permit binding of ZNF768 to the binding motif. In this context it will be important to determine at which stage of cell differentiation the access of ZNF768 to its binding motif is regulated.

The observed differential binding of ZNF768 in Raji and U2OS cells further prompted us to ask whether binding of ZNF768 can mark differentially expressed genes in both cell lines. In fact, we found a general correlation between ZNF768 binding and the activity of adjacent genes. In particular, we found a correlation between ZNF768 binding and gene expression for those genes that are active only in Raji or U2OS cells. From these data we conclude that binding of ZNF768 can mark commonly as well as cell type-specifically expressed genes in Raji and U2OS cells.

712 Nucleic Acids Research, 2019, Vol. 47, No. 2

A

Raji

gene name	enrichment factor (log2)	p-value (-log10)
NEURL4	11.50	5.68
ZNF768	11.18	3.27
MSANTD4	10.95	2.10
CXorf38	10.68	5.11
NIPSNAP2	10.50	6.40
MRPS28	10.23	1.67
ELP3	10.05	6.41
CEP170-1	10.00	2.90
ARL1	9.97	5.35
USP13	9.68	2.36
CCDC92	9.50	2.23
CEP170-2	9.50	7.20
MRPS23	8.85	1.47
MCCC1	8.55	3.21
VARS	8.49	5.70
ELP2	8.44	3.13
PRRC1	8.40	1.45
CENPE	8.36	5.04
MRPS31	8.35	1.34
MRPS34	8.34	1.49
TWNK	8.22	4.86
TRIM33	8.16	4.68
MRPS35	7.72	1.39
SUGP2	7.64	1.57
ELP1	6.76	3.66
HERC2	6.26	2.17
MCCC2	6.19	3.06
BMS1	5.82	5.52
KIF2A	5.73	4.20
NIN	4.72	1.57

B

U2OS

gene name	enrichment factor (log2)	p-value (-log10)
USP13	17.24	13.37
ZNF768	17.12	3.87
GAGE2D	16.91	5.71
ELP3	15.92	3.68
ZNF837	13.97	5.98
AEBP1	13.83	2.62
ELP2	13.67	2.37
SRSF5	12.80	6.61
CXorf38	11.78	9.08
MRPS34	11.20	5.91
MCCC2	11.18	2.68
CEP170-2	10.72	5.71
CEP170-1	10.35	3.50
CCDC92	10.32	5.21
HLA-DRB1	10.03	7.30
MRPS14	9.69	1.31
SUGP2	9.44	2.32
ELP1	9.35	3.68
TRIM33	9.06	4.54
MRPS23	8.73	1.37
CCDC47	8.27	1.58
NIN	7.99	3.49
GRPEL1	7.76	1.63
CENPE	7.74	3.25
HERC2	7.44	3.14
MCCC1	7.11	1.32
MRPS26	6.01	1.33
KIF2A	5.53	3.67
MSANTD4	5.39	3.71
AHNAK	5.15	4.03

C

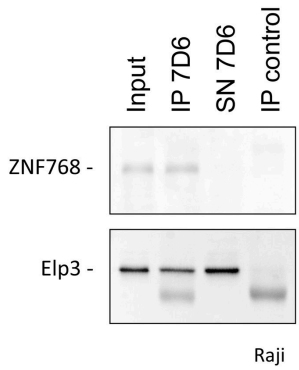


Figure 8. ZNF768 interactome. ZNF768 was immunoprecipitated from cellular extracts of (**A**) Raji and (**B**) U2OS cells. Thirty interaction factors with the highest enrichment are shown. Common factors in both cell lines are depicted in red. The list of all interaction factors is shown in Supplementary Table S4. (**C**) ZNF768 mAb 7D6 specifically co-immunoprecipitates Elp3 from cellular extracts of Raji cells.

ZNF768 functions as transcription factor

Finally, we asked if binding of ZNF768 is required for expression of specific genes. To demonstrate this we studied the transcriptome of U2OS cells 12 h after overexpression of a ZNF768 mutant lacking the N-terminal domain. We found several hundred genes that were significantly repressed after expression of this dominant-negative mutant. We also found a few induced genes, which may be upregulated indirectly. The gene ontology analysis of repressed genes revealed several gene classes related to transcriptional regulation suggesting that ZNF768 is hierarchically located upstream of a network of transcription factor genes and may function as a regulatory master gene for this network.

The notion that ZNF68 may act as a transcription factor was further supported by mass spectrometric analysis of the ZNF768 interactome in Raji and U2OS cells. In both cell

Nucleic Acids Research, 2019, Vol. 47, No. 2 713

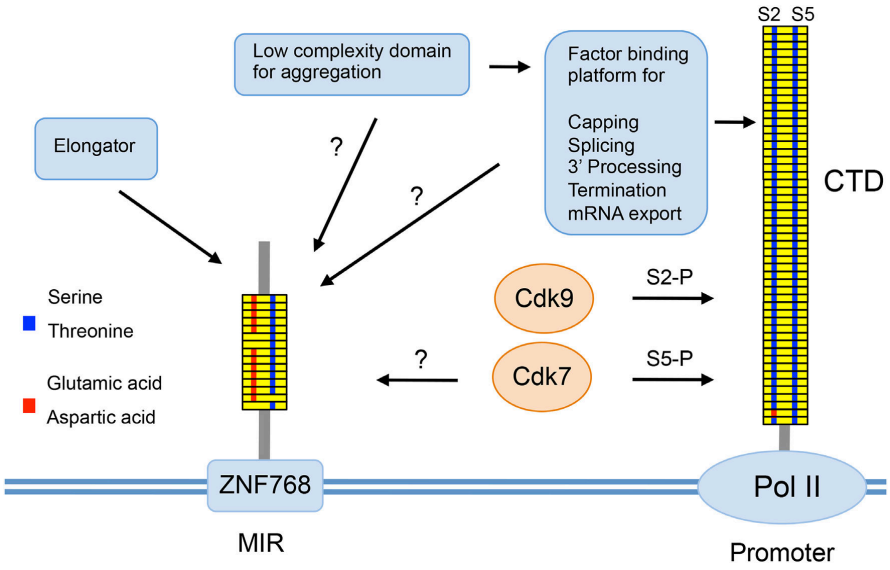


Figure 9. Known functions and regulation of Pol II CTD via its heptad repeat array and the implication with regards to possible functions and regulation of the heptad array in ZNF768 of placentalia (represented by human ZNF768).

lines ZNF768 interacts with subunits Elp1, Elp2 and Elp3 of the Elongator complex. The complex is conserved from yeast to mammals, consists of six subunits, Elp1-6, and has been proposed to function in the control of RNA elongation (56). The Elongator was found associated to the hyperphoshorylated form of Pol II, but the mode of interaction and the involved Elongator subunits are still elusive. Our data suggest that recruitment of Elongator to active genes may also occur by ZNF768. ZNF768 binds first a subcomplex of Elongator consisting of Elp1-3 that subsequently may assemble with subunits Elp4-6. In the future it will be interesting to study if heptad repeats of ZNF768 are involved in the recruitment of Elongator, as suggested for the CTD of Pol II, and if ZNF768 of marsupials lacks the ability of Elongator recruitment.

Originally, the array of heptad repeats in ZNF768 attracted our attention to study the function of ZNF768 as transcriptional activator due to its similarity to the array of heptad repeats in CTD of Pol II. This raises a couple of intriguing questions. First, can this array fulfill similar or related functions as the array of heptad repeats in CTD? If so, can the acidic amino acids that are present at many positions in heptad repeats of ZNF768 mimic a hyperphosphorylated form of Pol ll? Such a mimicry is most likely for position 2 of heptad repeats in ZNF768, which contains glutamic acid in almost all repeats across all species. It is tempting to speculate that binding of ZNF768 can recruit cellular factors to genomic loci that otherwise are recruited only if serine-2 of CTD is phosphorylated, e.g. by Cdk9, or other kinases (see model in Figure 9). In contrast, serine-5 residues are conserved between ZNF768 and the CTD and may depend on phosphorylation in ZNF768, similar as in the CTD, to allow interaction with other factors. Future work will address these and other questions and illuminate if and how the new regulatory network of ZNF768 and MIR sequences has contributed to speciation of placentalia.

DATA AVAILABILITY

GEO submissions: ChIP-Seq (GSE111879), RNA-seq Raji cells (GSE111880), RNA-seq U2OS cells (GSE111881). The mass spectrometry proteomics data have been deposited to the ProteomeXchange Consortium via the PRIDE (57) partner repository with the dataset identifier PXD010831.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank Elisabeth Kremmer for help with the generation of ZNF768 mAb.

FUNDING

D.E. and A.I. were supported by the Deutsche Forschungsgemeinschaft (DFG), SFB1064, Chromatin Dynamics and DFG excellence cluster CIPSM. In D.E. and J.C.A. labs, the work was supported by a German-French BMBF-ANR grant 'EpiGlyco'. CNRS; 'Agence Nationale de la Recherche' (ANR); 'amorcage jeunes équipes' Fondation pour la Recherche Medicale FRM [AJE20130728183 to J.C.A.]; Deutsche Forschungsgemeinschaft (DFG) [FR2938/7-1 and CRC 1123 (Z2) to C.C.F. and M.K.]; Deutsche Forschungsgemeinschaft (DFG) [GE 976/9-2 to M.G.] and is a member of the DFG excellence cluster ImmunoSensation. Funding for open access charge: Helmholtz Center Munich.

Conflict of interest statement. None declared.

REFERENCES

1. Kramerov,D.A. and Vassetzky,N.S. (2011) Origin and evolution of SINEs in eukaryotic genomes. *Heredity (Edinb.)*, **107**, 487–495.
2. Redi,C.A. and Capanna,E. (2012) Genome size evolution: sizing mammalian genomes. *Cytogenet Genome Res.*, **137**, 97–112.
3. Jurka,J., Zietkiewicz,E. and Labuda,D. (1995) Ubiquitous mammalian-wide interspersed repeats (MIRs) are molecular fossils from the mesozoic era. *Nucleic Acids Res.*, **23**, 170–175.
4. Smit,A.F. and Riggs,A.D. (1995) MIRs are classic, tRNA-derived SINEs that amplified before the mammalian radiation. *Nucleic Acids Res.*, **23**, 98–102.
5. Jjingo,D., Conley,A.B., Wang,J., Marino-Ramirez,L., Lunyak,V.V. and Jordan,I.K. (2014) Mammalian-wide interspersed repeat (MIR)-derived enhancers and the regulation of human gene expression. *Mob. DNA*, **5**, 14.
6. Jjingo,D., Huda,A., Gundapuneni,M., Marino-Ramirez,L. and Jordan,I.K. (2011) Effect of the transposable element environment of human genes on gene length and expression. *Genome Biol. Evol.*, **3**, 259–271.
7. Smith,A.M., Sanchez,M.J., Follows,G.A., Kinston,S., Donaldson,I.J., Green,A.R. and Gottgens,B. (2008) A novel mode of enhancer evolution: the Tal1 stem cell enhancer recruited a MIR element to specifically boost its activity. *Genome Res.*, **18**, 1422–1432.
8. Wang,J., Vicente-Garcia,C., Seruggia,D., Molto,E., Fernandez-Minan,A., Neto,A., Lee,E., Gomez-Skarmeta,J.L., Montoliu,L., Lunyak,V.V. *et al.* (2015) MIR retrotransposon sequences provide insulators to the human genome. *Proc. Natl. Acad. Sci. U.S.A.*, **112**, E4428–E4437.
9. Varshney,D., Vavrova-Anderson,J., Oler,A.J., Cowling,V.H., Cairns,B.R. and White,R.J. (2015) SINE transcription by RNA polymerase III is suppressed by histone methylation but not by DNA methylation. *Nat. Commun.*, **6**, 6569.
10. Yeganeh,M., Praz,V., Cousin,P. and Hernandez,N. (2017) Transcriptional interference by RNA polymerase III affects expression of the Polr3e gene. *Genes Dev.*, **31**, 413–421.
11. Carnevali,D., Conti,A., Pellegrini,M. and Dieci,G. (2017) Whole-genome expression analysis of mammalian-wide interspersed repeat elements in human cell lines. *DNA Res.*, **24**, 59–69.
12. Krull,M., Petrusma,M., Makalowski,W., Brosius,J. and Schmitz,J. (2007) Functional persistence of exonized mammalian-wide interspersed repeat elements (MIRs). *Genome Res.*, **17**, 1139–1145.
13. Smit,A.F., Hubley,R. and Green,P. (2015). *Repeatmasker Open 4.0*. http://www.repeatmasker.org.
14. Cournac,A., Koszul,R. and Mozziconacci,J. (2016) The 3D folding of metazoan genomes correlates with the association of similar repetitive elements. *Nucleic Acids Res.*, **44**, 245–255.
15. Feschotte,C. (2008) Transposable elements and the evolution of regulatory networks. *Nat. Rev. Genet.*, **9**, 397–405.
16. Medstrand,P., van de Lagemaat,L.N. and Mager,D.L. (2002) Retroelement distributions in the human genome: variations associated with age and proximity to genes. *Genome Res.*, **12**, 1483–1495.
17. Testori,A., Caizzi,L., Cutrupi,S., Friard,O., De Bortoli,M., Cora,D. and Caselle,M. (2012) The role of Transposable Elements in shaping the combinatorial interaction of Transcription Factors. *BMC Genomics*, **13**, 400.
18. Matthews,J.M. and Sunde,M. (2002) Zinc fingers–folds for many occasions. *IUBMB Life*, **54**, 351–355.
19. Yang,P., Wang,Y. and Macfarlan,T.S. (2017) The role of KRAB-ZFPs in transposable element repression and Mammalian evolution. *Trends Genet.*, **33**, 871–881.
20. Imbeault,M., Helleboid,P.Y. and Trono,D. (2017) KRAB zinc-finger proteins contribute to the evolution of gene regulatory networks. *Nature*, **543**, 550–554.
21. Najafabadi,H.S., Mnaimneh,S., Schmitges,F.W., Garton,M., Lam,K.N., Yang,A., Albu,M., Weirauch,M.T., Radovani,E., Kim,P.M. *et al.* (2015) C2H2 zinc finger proteins greatly expand the human regulatory lexicon. *Nat. Biotechnol.*, **33**, 555–562.
22. Bentley,D.L. (2014) Coupling mRNA processing with transcription in time and space. *Nat. Rev. Genet.*, **15**, 163–175.
23. Eick,D. and Geyer,M. (2013) The RNA polymerase II carboxy-terminal domain (CTD) code. *Chem. Rev.*, **113**, 8456–8490.
24. Harlen,K.M. and Churchman,L.S. (2017) The code and beyond: transcription regulation by the RNA polymerase II carboxy-terminal domain. *Nat. Rev. Mol. Cell Biol.*, **18**, 263–273.
25. Jeronimo,C., Collin,P. and Robert,F. (2016) The RNA polymerase II CTD: the increasing complexity of a low-complexity protein domain. *J. Mol. Biol.*, **428**, 2607–2622.
26. Zaborowska,J., Egloff,S. and Murphy,S. (2016) The pol II CTD: new twists in the tail. *Nat. Struct. Mol. Biol.*, **23**, 771–777.
27. Seipel,K., Georgiev,O., Gerber,H.P. and Schaffner,W. (1994) Basal components of the transcription apparatus (RNA polymerase II, TATA-binding protein) contain activation domains: is the repetitive C-terminal domain (CTD) of RNA polymerase II a "portable enhancer domain"? *Mol. Reprod. Dev.*, **39**, 215–225.
28. Suh,H., Hazelbaker,D.Z., Soares,L.M. and Buratowski,S. (2013) The C-terminal domain of Rpb1 functions on other RNA polymerase II subunits. *Mol. Cell*, **51**, 850–858.
29. Burke,K.A., Janke,A.M., Rhine,C.L. and Fawzi,N.L. (2015) Residue-by-residue view of in vitro FUS granules that bind the C-terminal domain of RNA polymerase II. *Mol. Cell*, **60**, 231–241.
30. Hnisz,D., Shrinivas,K., Young,R.A., Chakraborty,A.K. and Sharp,P.A. (2017) A phase separation model for transcriptional control. *Cell*, **169**, 13–23.
31. Kwon,I., Kato,M., Xiang,S., Wu,L., Theodoropoulos,P., Mirzaei,H., Han,T., Xie,S., Corden,J.L. and McKnight,S.L. (2013) Phosphorylation-regulated binding of RNA polymerase II to fibrous polymers of low-complexity domains. *Cell*, **155**, 1049–1060.
32. Schwartz,J.C., Ebmeier,C.C., Podell,E.R., Heimiller,J., Taatjes,D.J. and Cech,T.R. (2012) FUS binds the CTD of RNA polymerase II and regulates its phosphorylation at Ser2. *Genes Dev.*, **26**, 2690–2695.
33. Bornkamm,G.W., Berens,C., Kuklik-Roos,C., Bechet,J.M., Laux,G., Bachl,J., Korndoerfer,M., Schlee,M., Holzel,M., Malamoussi,A. *et al.* (2005) Stringent doxycycline-dependent control of gene activities using an episomal one-vector system. *Nucleic Acids Res.*, **33**, e137.
34. Rohrmoser,M., Holzel,M., Grimm,T., Malamoussi,A., Harasim,T., Orban,M., Pfisterer,I., Gruber-Eber,A., Kremmer,E. and Eick,D. (2007) Interdependence of Pes1, Bop1, and WDR12 controls nucleolar localization and assembly of the PeBoW complex required for maturation of the 60S ribosomal subunit. *Mol. Cell Biol.*, **27**, 3682–3694.
35. Shah,N., Maqbool,M.A., Yahia,Y., El Aabidine,A.Z., Esnault,C., Forne,I., Decker,T.M., Martin,D., Schuller,R., Krebs,S. *et al.* (2018) Tyrosine-1 of RNA polymerase II CTD controls global termination of gene transcription in mammals. *Mol. Cell*, **69**, 48–61.
36. Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics.*, **25**, 1754–1760.
37. Guo,Y., Mahony,S. and Gifford,D.K. (2012) High resolution genome wide binding event finding and motif discovery reveals transcription factor spatial binding constraints. *PLoS Comput. Biol.*, **8**, e1002638.
38. Machanick,P. and Bailey,T.L. (2011) MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics*, **27**, 1696–1697.
39. Crooks,G.E., Hon,G., Chandonia,J.M. and Brenner,S.E. (2004) WebLogo: a sequence logo generator. *Genome Res.*, **14**, 1188–1190.
40. Yu,G., Wang,L.G. and He,Q.Y. (2015) ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization. *Bioinformatics*, **31**, 2382–2383.
41. Harrow,J., Frankish,A., Gonzalez,J.M., Tapanari,E., Diekhans,M., Kokocinski,F., Aken,B.L., Barrell,D., Zadissa,A., Searle,S. *et al.* (2012) GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Res.*, **22**, 1760–1774.
42. Pollard,K.S., Hubisz,M.J., Rosenbloom,K.R. and Siepel,A. (2010) Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.*, **20**, 110–121.
43. Hahne,F. and Ivanek,R. (2016) Visualizing genomic data using gviz and bioconductor. *Methods Mol. Biol.*, **1418**, 335–351.
44. Quinlan,A.R. and Hall,I.M. (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**, 841–842.
45. Bonfert,T., Kirner,E., Csaba,G., Zimmer,R. and Friedel,C.C. (2015) ContextMap 2: fast and accurate context-based RNA-seq mapping. *BMC Bioinformatics*, **16**, 122.

46. Liao,Y., Smyth,G.K. and Shi,W. (2014) featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, **30**, 923–930.
47. Robinson,M.D., McCarthy,D.J. and Smyth,G.K. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.
48. Ritchie,M.E., Phipson,B., Wu,D., Hu,Y., Law,C.W., Shi,W. and Smyth,G.K. (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.*, **43**, e47.
49. Huang da,W., Sherman,B.T. and Lempicki,R.A. (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.*, **4**, 44–57.
50. Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B Nr.*, **57**, 289–300.
51. Kluge,M. and Friedel,C.C. (2018) Watchdog - a workflow management system for the distributed analysis of large-scale experimental data. *BMC Bioinformatics.*, **19**, 97.
52. Holzel,M., Rohrmoser,M., Schlee,M., Grimm,T., Harasim,T., Malamoussi,A., Gruber-Eber,A., Kremmer,E., Hiddemann,W., Bornkamm,G.W. *et al.* (2005) Mammalian WDR12 is a novel member of the Pes1-Bop1 complex and is required for ribosome biogenesis and cell proliferation. *J. Cell Biol.*, **170**, 367–378.
53. Kellner,M., Rohrmoser,M., Forne,I., Voss,K., Burger,K., Muhl,B., Gruber-Eber,A., Kremmer,E., Imhof,A. and Eick,D. (2015) DEAD-box helicase DDX27 regulates 3′ end formation of ribosomal 47S RNA and stably associates with the PeBoW-complex. *Exp. Cell Res.*, **334**, 146–159.
54. Ishihama,Y., Rappsilber,J. and Mann,M. (2006) Modular stop and go extraction tips with stacked disks for parallel and multidimensional Peptide fractionation in proteomics. *J. Proteome Res.*, **5**, 988–994.
55. Guo,J. and Price,D.H. (2013) RNA polymerase II transcription elongation control. *Chem. Rev.*, **113**, 8583–8603.
56. Otero,G., Fellows,J., Li,Y., de Bizemont,T., Dirac,A.M., Gustafsson,C.M., Erdjument-Bromage,H., Tempst,P. and Svejstrup,J.Q. (1999) Elongator, a multisubunit component of a novel RNA polymerase II holoenzyme for transcriptional elongation. *Mol. Cell*, **3**, 109–118.
57. Vizcaino,J.A., Csordas,A., del-Toro,N., Dianes,J.A., Griss,J., Lavidas,I., Mayer,G., Perez-Riverol,Y., Reisinger,F., Ternent,T. *et al.* (2016) 2016 update of the PRIDE database and its related tools. *Nucleic Acids Res.*, **44**, 447–456.

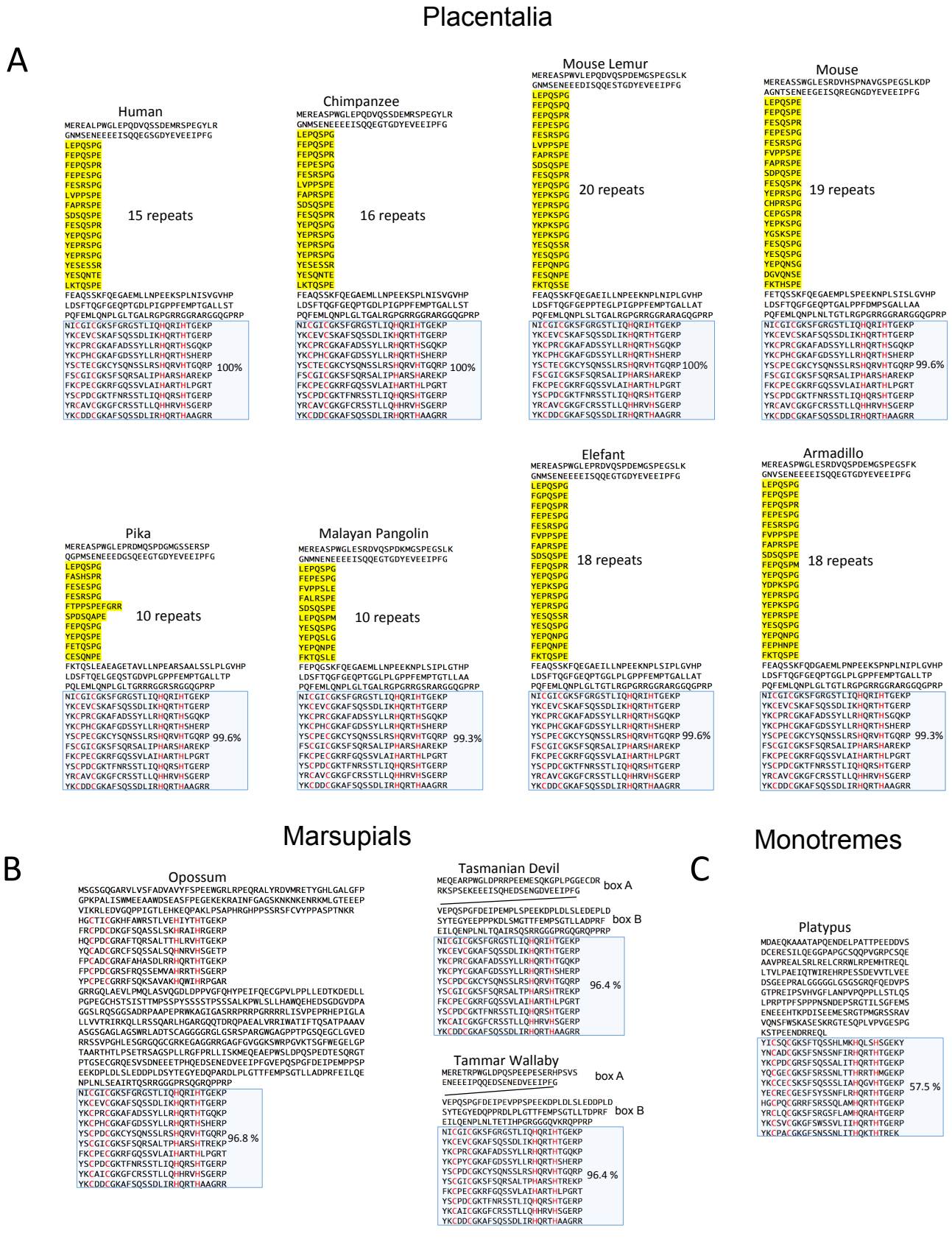

Placentalia
A
Human
MEREALPWGLEPQDVQSSDEMRSPEGYLR
GNMSENEEEEISQQEGSGDYEVEEIPFG
LEPQSPG
FEPQSPE
FEPQSPR
FEPESPG
FESRSPG
LVPPSPE
FAPRSPE
SDSQSPE
FESQSPR
YEPQSPG
YEPRSPG
YEPRSPG
YESESSR
YESQNTE
LKTQSPE
15 repeats
FEAQSSKFQEGAEMLLNPEEKSPLNISVGVHP
LDSFTQGFGEQPTGDLPIGPPFEMPTGALLST
PQFEMLQNPLGLTGALRGPGRRGGRARGGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCSKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCTECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
100%
Chimpanzee
MEREASPWGLEPQDVQSSDEMRSPEGYLR
GNMSENEEEEISQQEGTGDYEVEEIPFG
LEPQSPG
FEPQSPE
FEPQSPR
FEPESPG
FESRSPG
LVPPSPE
FAPRSPE
SDSQSPE
FESQSPR
YEPQSPG
YEPRSPG
YEPRSPG
YESESSR
YESQNTE
LKTQSPE
16 repeats
FEAQSSKFQEGAEMLLNPEEKSPLNISVGVHP
LDSFTQGFGEQPTGDLPIGPPFEMPTGALLST
PQFEMLQNPLGLTGALRGPGRRGGRARGGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCSKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCTECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
100%
Mouse Lemur
MEREASPWVLEPQDVQSPDEMGSPEGSLK
GNMSENEEEEDISQQESTGDYEVEEIPFG
LEPQSPG
FEPQSPQ
FEPQSPR
FEPESPG
FESRSPG
LVPPSPE
FAPRSPE
SDSQSPE
FESQSPR
YEPQSPG
YEPKSPG
YEPRSPG
YEPKSPG
YKPKSPG
YEPKSPG
YESQSSR
YESQSPG
FEPQNPG
FESQNPE
FKTQSSE
20 repeats
FEAQSSKFQEGAEILLNPEEKNPLNIPLGVHP
LDSFTQGFGEPPTEGLPIGPPFEMPTGALLAT
PQFEMLQNPLSLTGALRGPGRRGGRARAGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCSKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCTECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
100%
Mouse
MEREASSWGLESRDVHSPNAVGSPEGSLKDP
AGNTSENEEGEISQREGNGDYEVEEIPFG
LEPQSPE
FEPQSPE
FESQSPR
FEPESPG
FESRSPG
FVPPSPE
FAPRSPE
SDPQSPE
FESQSPK
YEPRSPG
CHPRSPG
CEPGSPR
YEPKSPG
YGSKSPE
FESQSPG
YESQSPG
YEPQNSG
DGVQNSE
FKTHSPE
19 repeats
FETQSSKFQEGAEMPLSPEEKNPLSISLGVHP
LDSFTQGFGEQPTGALPPFDMPSGALLAA
PQFEMLQNPLNLTGTLRGPGRRGGRARGGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCSKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCPECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
99.6%
Pika
MEREASPWGLEPRDMQSPDGMGSSERSP
QGPMSENEEEDGSQEEGTGDYEVEEIPFG
LEPQSPG
FASHSPR
FESESPG
FESRSPG
FTPPSPEFGRR
SPDSQAPE
FEPQSPG
YEPQSPE
FETQSPG
CESQNPE
10 repeats
FKTQSLEAEAGETAVLLNPEARSAALSSLPLGVHP
LDSFTQELGEQSTGDVPLGPPFEMPTGALLTP
PQLEMLQNPLGLTGRRRGGRSRGGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCSKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCPECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
99.6%
Malayan Pangolin
MEREASPWGLESRDVQSPDKMGSPEGSLK
GNMNENEEEEISQQEGTGDYEVEEIPFG
LEPQSPG
FEPESPG
FVPPSLE
FALRSPE
SDSQSPE
LEPQSPM
YESQSPG
YEPQSLG
YEPQNPE
FKTQSLE
10 repeats
FEPQGSKFQEGAEMLLNPEEKNPLSIPLGTHP
LDSFTQGFGEQPTGGLPLGPPFEMPTGTLLAA
PQFEMLQNPLGLTGALRGPGRRGSRARGGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCGKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCPECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
99.3%
Elefant
MEREASPWGLEPRDVQSPDEMGSPEGSLK
GNMSENEEEEISQQEGTGDYEVEEIPFG
LEPQSPG
FGPQSPE
FEPQSPR
FEPESPG
FESRSPG
FVPPSPE
FAPRSPE
SDSQSPE
FEPQSPR
YEPQSPG
YEPKSPG
YEPRSPG
YEPRSPG
YESQSSR
YESQSPG
YEPQNPG
FEPQNPE
FKTQSPE
18 repeats
FEAQSSKFQEGAEILLNPEEKNPLSIPLGVHP
LDSFTQGFGEQPTGGLPLGPPFEMPTGALLAT
PQFEMLQNPLGLTGTLRGPGRRGGRARGGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCSKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCPECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
99.6%
Armadillo
MEREASPWGLESRDVQSPDEMGSPEGSFK
GNVSENEEEEISQQEGTGDYEVEEIPFG
LEPQSPG
FEPQSPE
FEPQSPR
FEPESPG
FESRSPG
FVPPSPE
FAPRSPE
SDSQSPE
FEPQSPM
YEPQSPG
YDPKSPG
YEPRSPG
YEPKSPG
YEPRSPE
YESQSPG
YEPQNPG
FEPHNPE
FKTQSPE
18 repeats
FEAQSSKFQDGAEMLPNPEEKSPNPLNIPLGVHP
LDSFTQGFGEQPTGGLPLGPPFEMPTGALLTP
PQFEMLQNPLGLTGTLRGPGRRGGRARGGQGPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCSKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHSGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCPECGKCYSQNSSLRSHQRVHTGQRP
FSCGICGKSFSQRSALIPHARSHAREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YRCAVCGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
99.3%
Marsupials
B
Opossum
MSGSGQGARVLVSFADVAVYFSPEEWGRLRPEQRALYRDVMRETYGHLGALGFP
GPKPALISWMEEAAWDSEASFPEGEKEKRAINFGAGSKNNKNENRKMLGTEEEP
VIKRLEDVGQPPIGTLEHKEQPAKLPSAPHRGHPPSSRSFCVYPPASPTNKR
HGCTICGKHFAWRSTLVEHIYTHTGEKP
FRCPDCDKGFSQASSLSKHRAIHRGERP
HQCPDCGRAFTQRSALTTHLRVHTGEKP
YQCADCGRCFSQSSALSQHNRVHSGETP
FPCADCGRAFAHASDLRRHQRTHTGEKP
FPCPDCGRSFRQSSEMVAHRRTHSGERP
YPCPECGRRFSQKSAVAKHQWIHRPGAR
GRRGQLAEVLPMQLASVQGDLDPPVGFQHYPEIFQECGPVLPPLLEDTKDEDLL
PGPEGCHSTSISTTMPSSPYSSSSTPSSSALKPWLSLLHAWQEHEDSGDGVDPA
GGSLRQSGGSADRPAAPEPRWKAGIGASRRPRRPGRRRRLISVPEPRHEPIGLA
LLVVTRIRKQLLRSSQARLHGARGQQTDRQPAEALVRRIWATIFTQSATPAAAV
ASGSGAGLAGSWRLADTSCAGGGGRGLGSRSPARGWGAGPPTPGSQEGCLGVED
RRSSVPGHLESGRGQGCGRKEGAGGRRGAGFGVGGKSWRPGVKTSGFWEGELGP
TAARTHTLPSETRSAGSPLLRGFPRLLISKMEQEAEPWSLDPQSPEDTESQRGT
PTGSECGRQESVSDNEEETPHQEDSENEDVEEIPFGVEPQSPGFDEIPEMPPSP
EEKDPLDLSLEDDPLDSYTEGYEDQPARDLPLGTTFEMPSGTLLADPRFEILQE
NPLNLSEAIRTQSRRGGGPRSQGRQPPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCGKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHTGQKP
YKCPHCGKAFGDSSYLLRHQRTHSHERP
YSCPDCGKCYSQNSSLRSHQRVHTGQRP
YSCGICGKSFSQRSALTPHARSHTREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YKCAICGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
96.8 %
Tasmanian Devil
MEQEARPWGLDPRRPEEMESQKGPLPGGECDR
RKSPSEKEEEISQHEDSENGDVEEIPFG
box A
VEPQSPGFDEIPEMPLSPEEKDPLDLSLEDEPLD
SYTEGYEEPPPKDLSMGTTFEMPSGTLLADPRF
box B
EILQENPLNLTQAIRSQSRRGGGPRGQGRQPPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCGKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHTGQKP
YKCPYCGKAFGDSSYLLRHQRTHSHERP
YSCPDCGKCYSQNSSLRSHQRVHTGQRP
YSCGICGKSFSQRSALTPHARSHTREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YKCAICGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
96.4 %
Tammar Wallaby
MERETRPWGLDPQSPEEPESERHPSVS
ENEEEIPQQEDSENEDVEEIPFG
box A
VEPQSPGFDEIPEVPPSPEEKDPLDLSLEDDPLD
SYTEGYEDQPPRDLPLGTTFEMPSGTLLTDPRF
box B
EILQENPLNLTETIHPGRGGGQVKRQPPRP
NICGICGKSFGRGSTLIQHQRIHTGEKP
YKCEVCGKAFSQSSDLIKHQRTHTGERP
YKCPRCGKAFADSSYLLRHQRTHTGQKP
YKCPYCGKAFGDSSYLLRHQRTHSHERP
YSCPDCGKCYSQNSSLRSHQRVHTGQRP
YSCGICGKSFSQRSALTPHARSHTREKP
FKCPECGKRFGQSSVLAIHARTHLPGRT
YSCPDCGKTFNRSSTLIQHQRSHTGERP
YKCAICGKGFCRSSTLLQHHRVHSGERP
YKCDDCGKAFSQSSDLIRHQRTHAAGRR
96.4 %
C
Monotremes
Platypus
MDAEQKAAATAPQENDELPATTPEEDDVS
DCERESILQEGGPAPGCSQQPVGRPCSQE
AAVPREALSRLRELCRRWLRPEMHTREQL
LTVLPAEIQTWIREHRPESSDEVVTLVEE
DSGEEPRALGGGGGLGSGSGRQFQEDVPS
GTPREIPSVHHVGFLANPVPQPPLLSTLQS
LPRPTPFSPPPNSNDEPSRGTILSGFEMS
ENEEEHTKPDISEEMESRGTPMGRSSRAV
VQNSFWSKASESKRGTESQPLVPVGESPG
KSTPEENDRREQL
YICSQCGKSFTQSSHLMKHQLSHSGEKY
YNCADCGKSFSNSSNFIRHQRTHTGEKP
YKCPDCGRSFSQSSALITHQRTHTGEKP
YQCGECGKSFSRSSNLTTHRRTHMGEKP
YKCCECSKSFSQSSSLIAHQGVHTGEKP
YECRECGESFSYSSNFLRHQRTHTGERP
HGCPQCGRRFSRSSQLAMHQRTHTGEKP
YRCLQCGKSFSRGSFLAMHQRAHTGERP
YKCSVCGKGFSWSSVLIIHQRTHTGERP
YKCPACGKGFSNSSNLITHQKTHTREK
57.5 %

Figure S1

Figure S1

Protein sequence of ZNF768 in various species of mammals. A domain of 10 zinc fingers is highly conserved in all mammals (blue box).
(**A**) An array containing 10 - 20 heptad repeats (yellow box) is a characteristic of placental animals. (**B**) Marsupials Opossum, Tammar Wallaby and Tasmanian devil lack the array of heptad-repeats. ZNF768 Tammar Wallaby and Tasmanian devil contain box A and box B. (**C**) Monotreme Platypus lacks the array of heptad repeats, box A and box B, and shows further a reduced conservation of the zinc finger domain (blue box). Sorting was performed according the phylogenetic relationship.

Reference sequence of mammalian ZNF768 proteins:

Human, *Homo sapiens*, NP_078947.3
Chimpanzee, *Pan troglodytes*, XP_016785186.1
Mouse lemur, *Microcebus murinus*, XP_012619672.1
Mouse, *Mus musculus*, NP_666314.1
Pika, *Ochotona princeps*, XP_012782987
Malayan pangolin, *Manis javanica*, XP_017519428.1
Elefant, *Loxodonta africana*, XP_010596820.1
Armadillo, *Dasypus novemcinctus*, XP_012375444.1
Opossum, *Monodelphis domestica*, XP_007498557.2
Tasmanian devil, *Sarcophilus harrisii*, XP_012398319.1
Tammar wallaby, *Macropus eugenii*, ENSMEUP00000000462
Platypus, *Ornithorhynchus anatinus,* ENSOANP00000018579

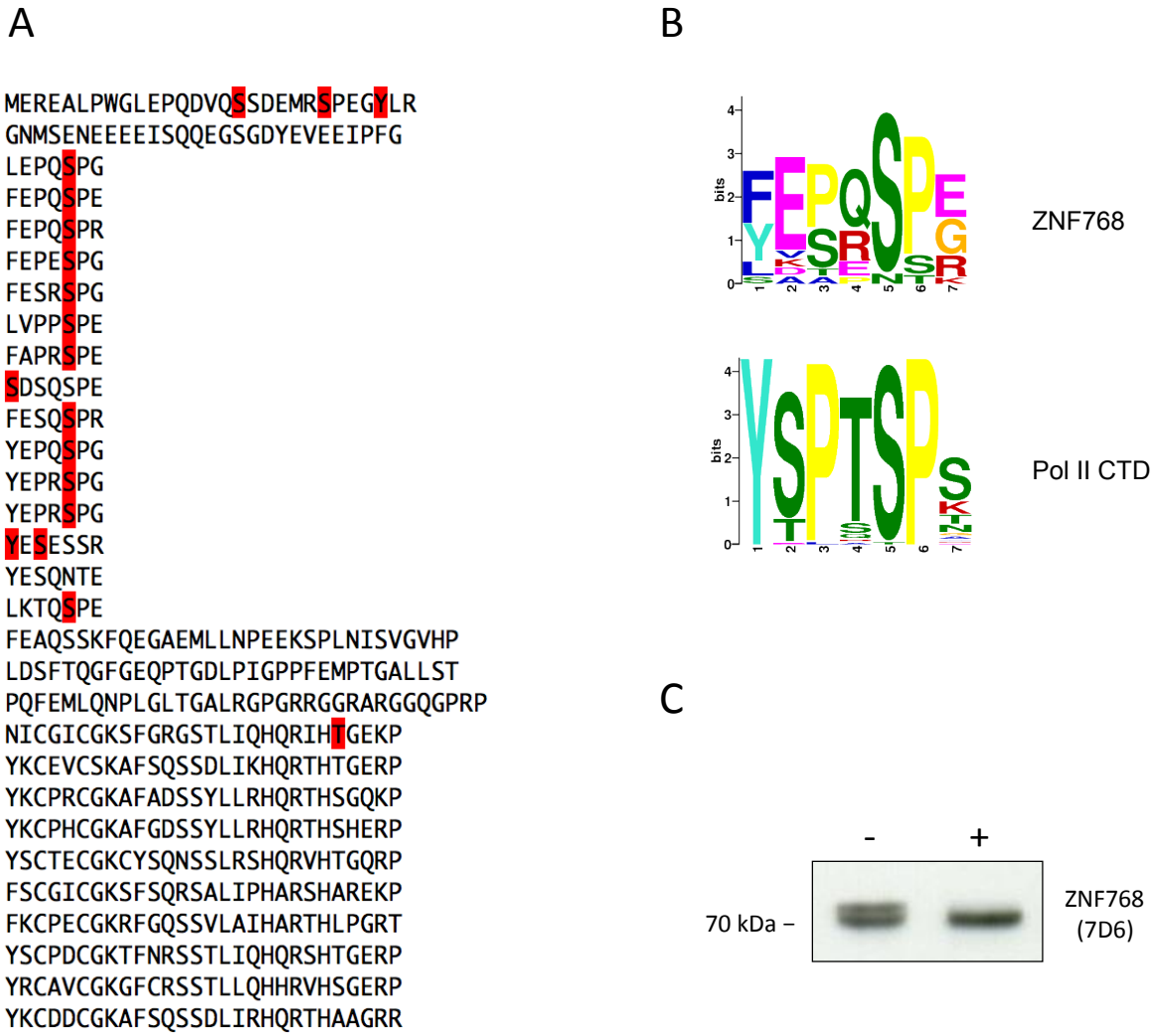


Figure S2

Phosphorylation of ZNF768. (**A**) Confirmed phosphorylation sites in human ZNF768 protein (www.cellsignal.com). (**B**) Heptad repeat consensus motif for ZNF768 determined from the 15 heptad repeats in human ZNF768 and Pol II CTD (Fig. 1B) using MEME. (**C**) Western blot of cellular extracts of U2OS cells before (-) and after (+) treatment with alkaline phosphatase.

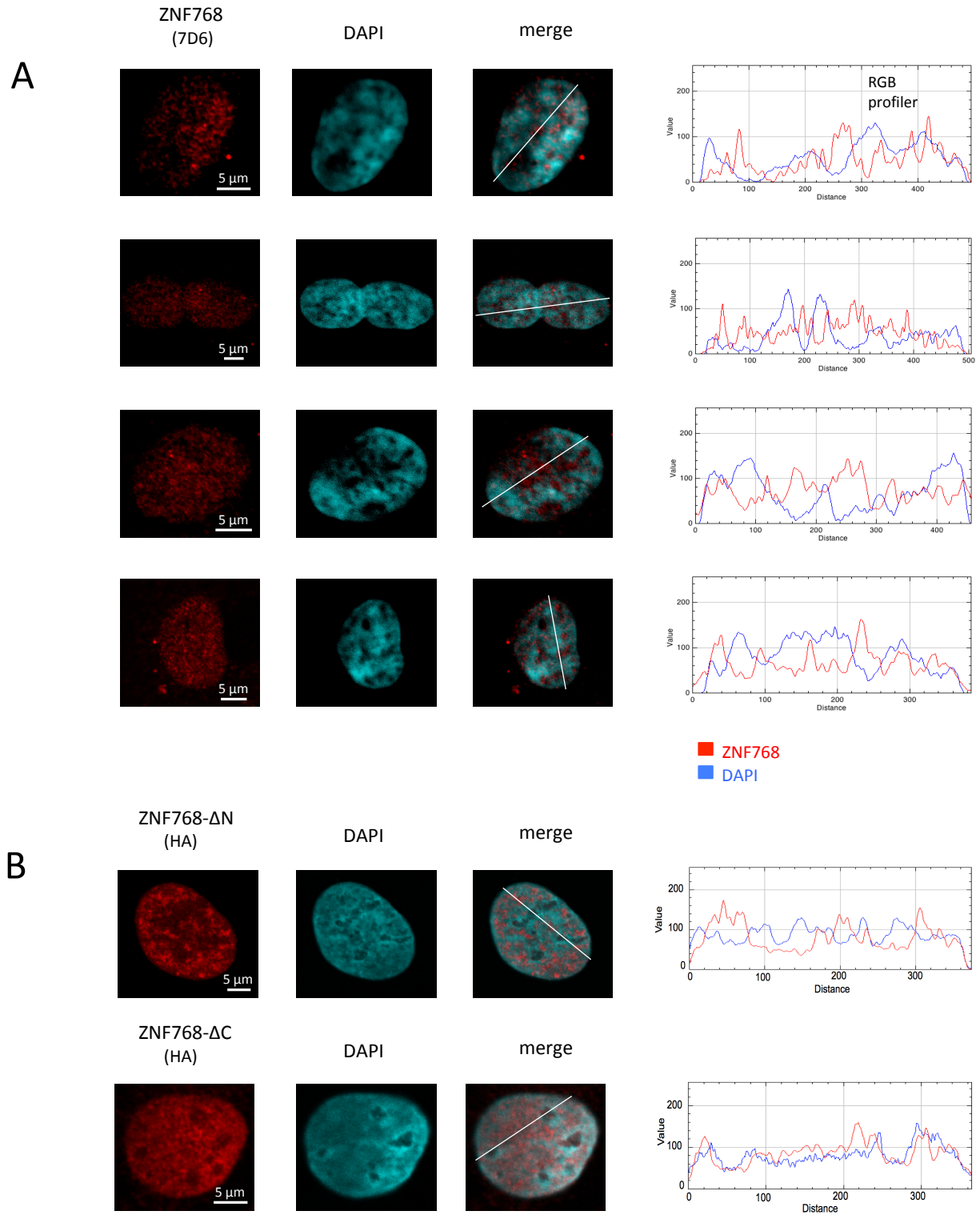


Figure S3

Confocal images of ZNF768 in U2OS cells. (**A**) Endogenous ZNF768 was stained with mAb 7D6 (red), chromatin with DAPI, and merged. RGB profiler is shown on the right hand site, white line marks the scanned area. (**B**) Confocal images of ZNF768 mutants (Structure of mutants is described in Figure 2E).

A

	# merged regions	unique (1/4)	diff cell (2/4)	consistent cell (2/4)	consistent cell+ (3/4)	all (4/4)
Raji (rep. 1)	15854 (56.1%)	7324 (25.2%)	631 (80.82%)	3048 (62.11%)	336 (89.88%)	2747 (98.33%)
Raji (rep. 2)	9242 (80.08%)	1169 (55.77%)	305 (96.72%)	3048 (62.11%)	216 (99.07%)	2747 (98.33%)
U2OS (rep. 1)	8983 (84.31%)	2421 (60.43%)	876 (86.53%)	716 (79.75%)	1654 (95.16%)	2747 (98.33%)
U2OS (rep. 2)	4527 (88.67%)	358 (29.33%)	60 (78.33%)	716 (79.75%)	87 (90.8%)	2747 (98.33%)
unique regions	21012 (58.09%)	11272 (36.07%)	936 (86%)	3764 (65.46%)	2293 (94.59%)	2747 (98.33%)

B

	MIRs	MIRs with motif	MIRs with stringent motif	motifs genome-wide	stringent motifs genome-wide
total	579294	70760	13812	423846	14852
bound by ZNF768	13168	11187	7481	12875	7706
Percentage bound	2.3	15.8	54.2	3.0	51.9

Figure S4

ZNF768 peaks in Raji and U2OS cells. (**A**) Number of peaks and percentage of peaks (in brackets) containing the binding motif in replicate experiment 1 and 2 in Raji and U2OS cells. Explanation of column headings: # merged regions = total number of peaks in each sample, overlapping peaks (peak centers ±100 bp) were merged within samples and across all 4 samples to obtain unique peak regions; unique (1/4) = number of peaks identified only in one sample; diff cell (2/4) = number of peaks identified in one replicate for each cell type; consistent cell (2/4) = number of peaks identified in both replicates for one cell type but not in any replicate for the other cell type; consistent cell+ (3/4) = number of peaks identified in both replicates for one cell type and one replicate for the other cell type; all (4/4) = number of peaks identified in all four samples. (**B**) ZNF768 peaks in MIR sequences and in the whole genome.

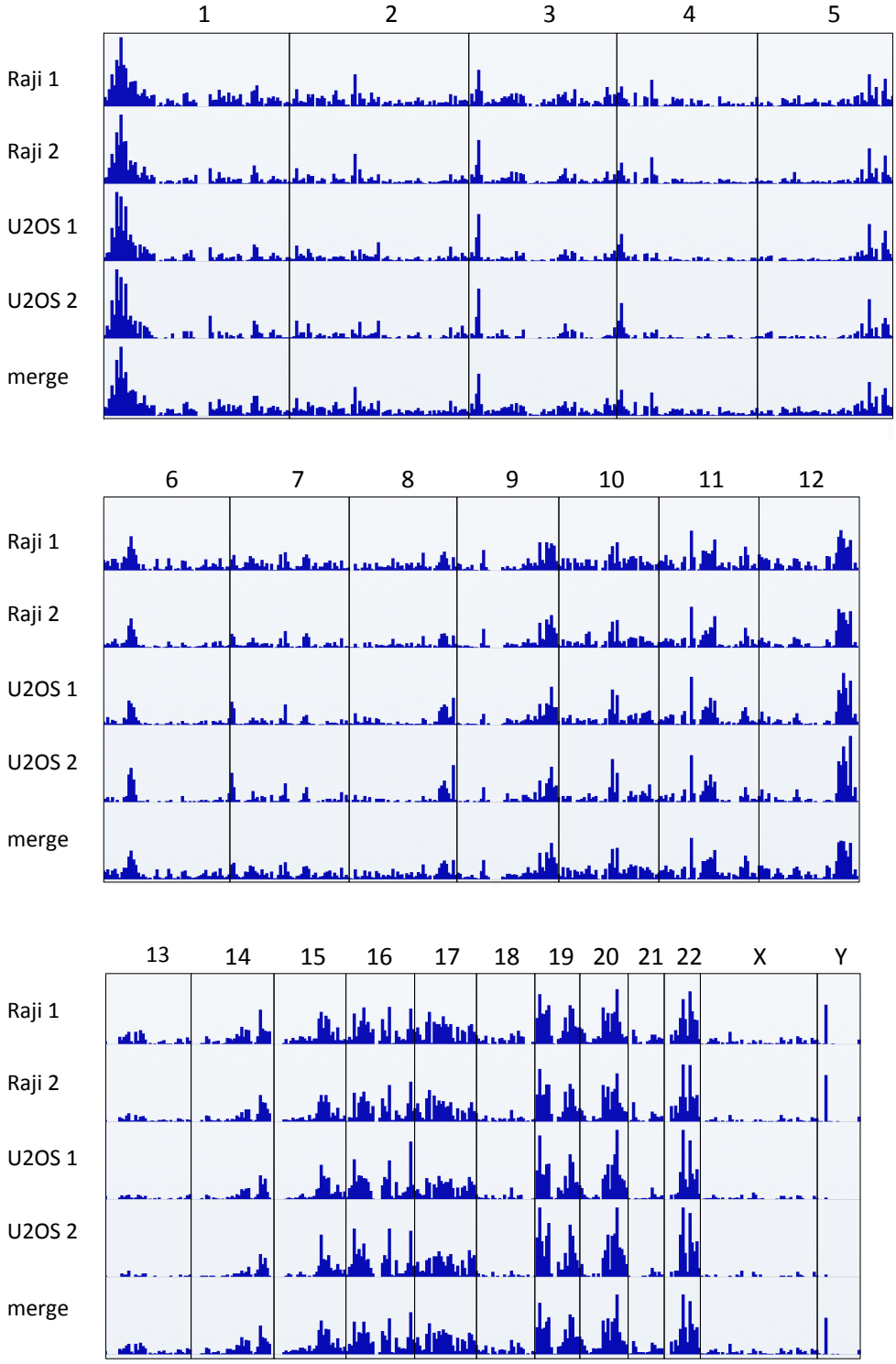
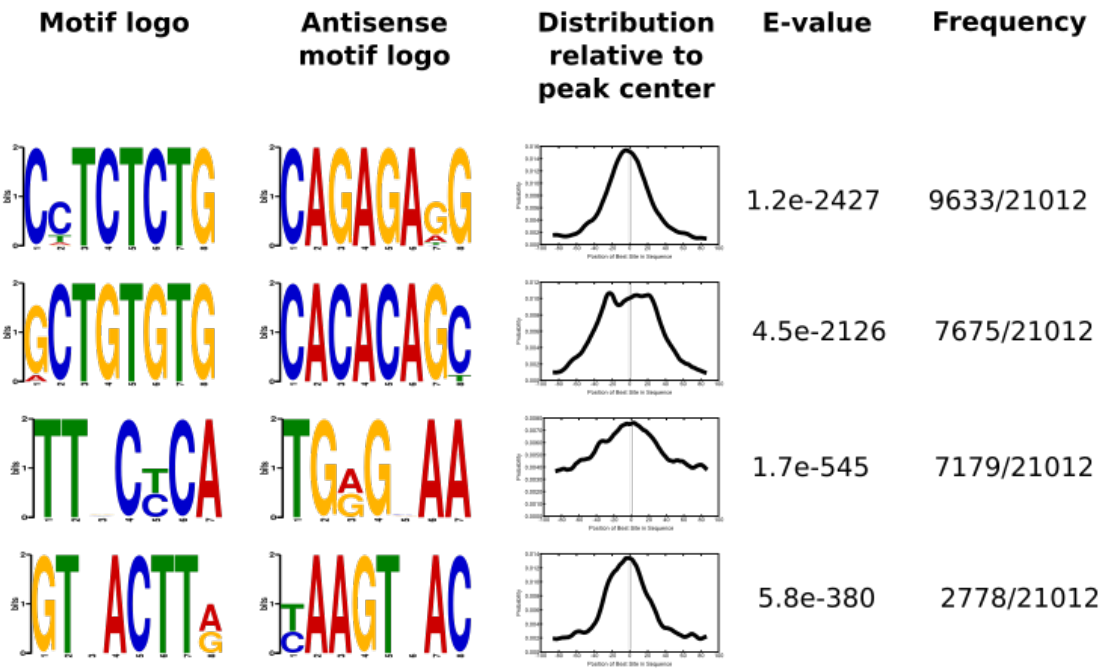


Figure S5
Distribution of ZNF768 peaks in ChIP experiments in Raji and U2OS cells over chromosomes 1 – 22, and X and Y chromosome.

A



B

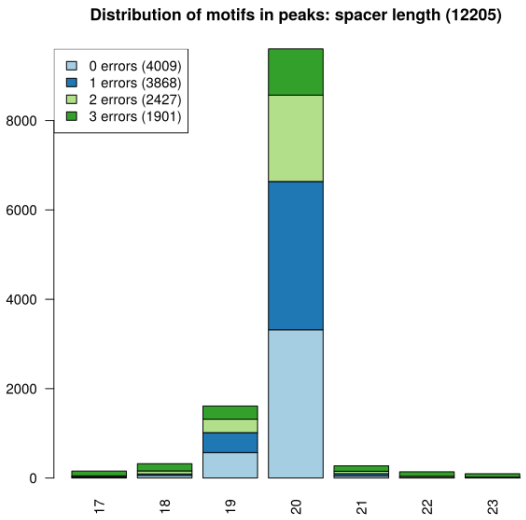


Figure S6

(**A**) Top four motif logos identified using MEME-ChIP in merged unique ZNF768 peaks in Raji and U2OS cells. Frequency indicates the number of merged unique peaks containing each motif. (**B**) Number of peaks matching the ZNF768 binding motif in Fig. 2A with a spacer length of 20±3bp. For each spacer length, the number of peaks containing the anchor regions with a certain number of mismatches is indicated.

A

M1: 5′- *CAGT***GCTGTGTGACCTTGGGCAAGTCACTTAACCTCTCTG***CAGT* -3′

M2: 5′- *CAGT***GCTGTGTG**CAGTCAGTCAGTCAGTCAGT**CCTCTCTG***CAGT* -3′

M3: 5′- *CAGT*CAGT**TGTGACCTTGGGCAAGTCACTTAACCTC**CAGT*CAGT* -3′

B

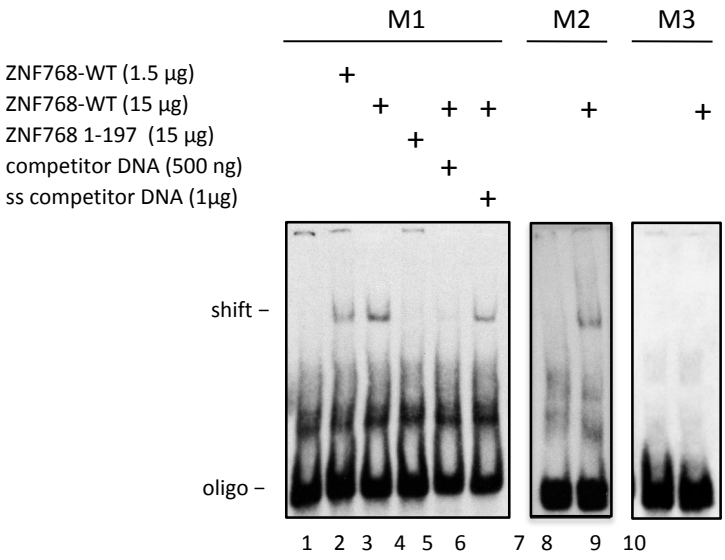
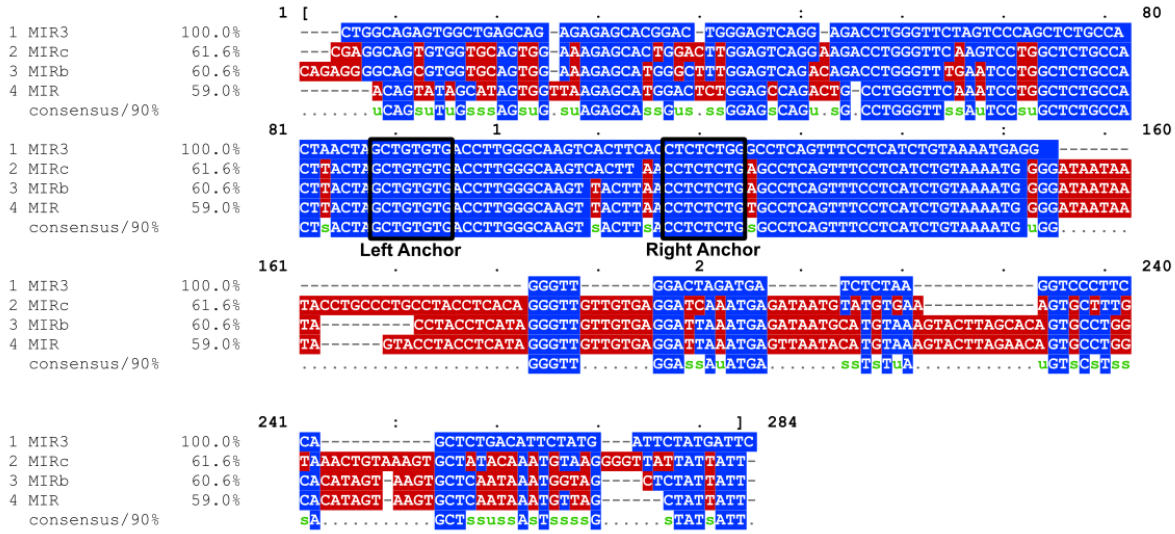


Figure S7

Electrophoretic mobility shift assay (EMSA) with ZNF768 protein. (**A**) Oligonucleotide with the sequence of the ZNF768 binding motif (M1, black sequence), with replacement of the spacer sequence (M2, red sequences), and with partial replacement of the anchor sequences (M3, red sequences). (**B**) Double-stranded fragments M1 – M3 were end-labelled with DIG-11-dUTP and analyzed in extracts with recombinant ZNF768 protein in EMSA.

A



B

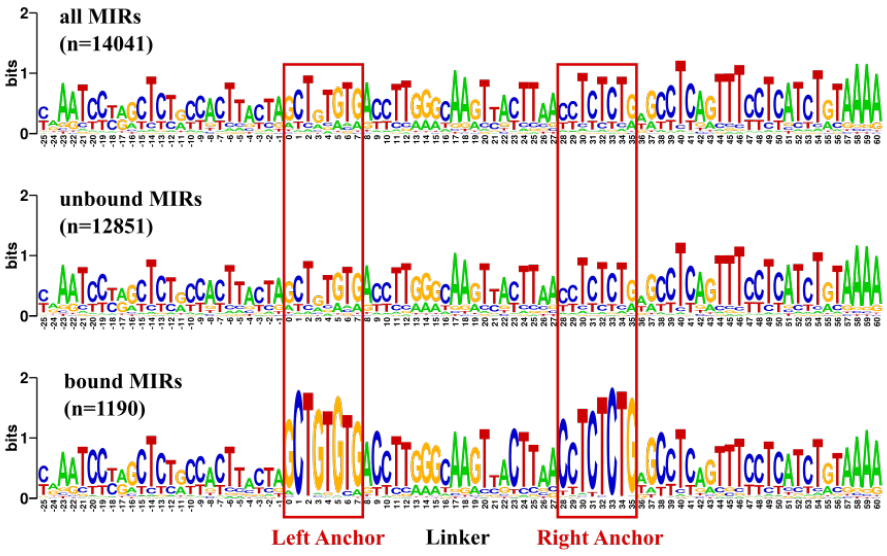


Figure S8

(**A**) Alignment of human MIR3, MIR3c, MIR3b, and MIR consensus sequences. The left and right anchor sequences of the ZNF768 binding motif from Fig. 3A are highlighted by black boxes. (**B**) Motif logos for human MIR sequences that align without gaps to the MIR consensus sequence in the region of the ZNF768 binding motif +25bp on either side (=14,041 MIR sequences). Logos are shown separately for all of these MIRs (top row), MIRs not bound by ZNF768 (middle row) and MIRs bound by ZNF768 (bottom row). MIRs not bound by ZNF768 show no particular conservation in human for the binding motif. The MIR core sequence covers the sequence from 93 nt to 159 nt (Smit and Riggs, Nucleic Acids Res. 23, 98-102).

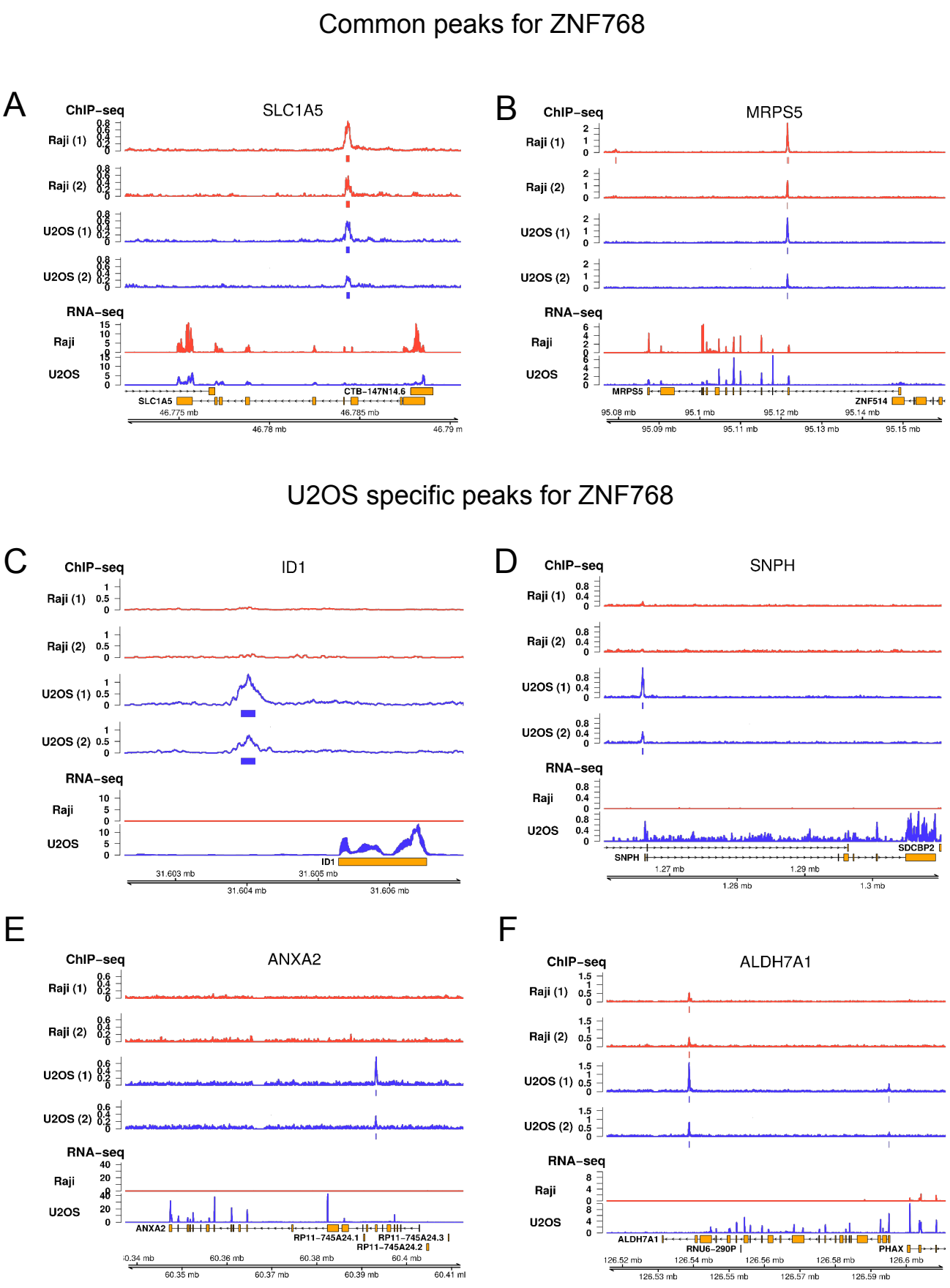

Common peaks for ZNF768
A
ChIP-seq
SLC1A5
Raji (1)
Raji (2)
U2OS (1)
U2OS (2)
RNA-seq
Raji
U2OS
SLC1A5
CTB-147N14.6
46.775 mb
46.785 mb
46.78 mb
46.79 m
B
ChIP-seq
MRPS5
Raji (1)
Raji (2)
U2OS (1)
U2OS (2)
RNA-seq
Raji
U2OS
MRPS5
ZNF514
95.08 mb
95.1 mb
95.12 mb
95.14 mb
95.09 mb
95.11 mb
95.13 mb
95.15 mb
U2OS specific peaks for ZNF768
C
ChIP-seq
ID1
Raji (1)
Raji (2)
U2OS (1)
U2OS (2)
RNA-seq
Raji
U2OS
ID1
31.603 mb
31.605 mb
31.604 mb
31.606 mb
D
ChIP-seq
SNPH
Raji (1)
Raji (2)
U2OS (1)
U2OS (2)
RNA-seq
Raji
U2OS
SNPH
SDCBP2
1.27 mb
1.29 mb
1.28 mb
1.3 mb
E
ChIP-seq
ANXA2
Raji (1)
Raji (2)
U2OS (1)
U2OS (2)
RNA-seq
Raji
U2OS
ANXA2
RP11-745A24.1
RP11-745A24.3
RP11-745A24.2
60.34 mb
60.36 mb
60.38 mb
60.4 mb
60.35 mb
60.37 mb
60.39 mb
60.41 mb
F
ChIP-seq
ALDH7A1
Raji (1)
Raji (2)
U2OS (1)
U2OS (2)
RNA-seq
Raji
U2OS
ALDH7A1
RNU6-290P
PHAX
126.52 mb
126.54 mb
126.56 mb
126.58 mb
126.6 mb
126.53 mb
126.55 mb
126.57 mb
126.59 mb

Figure S9

Raji specific peaks for ZNF768

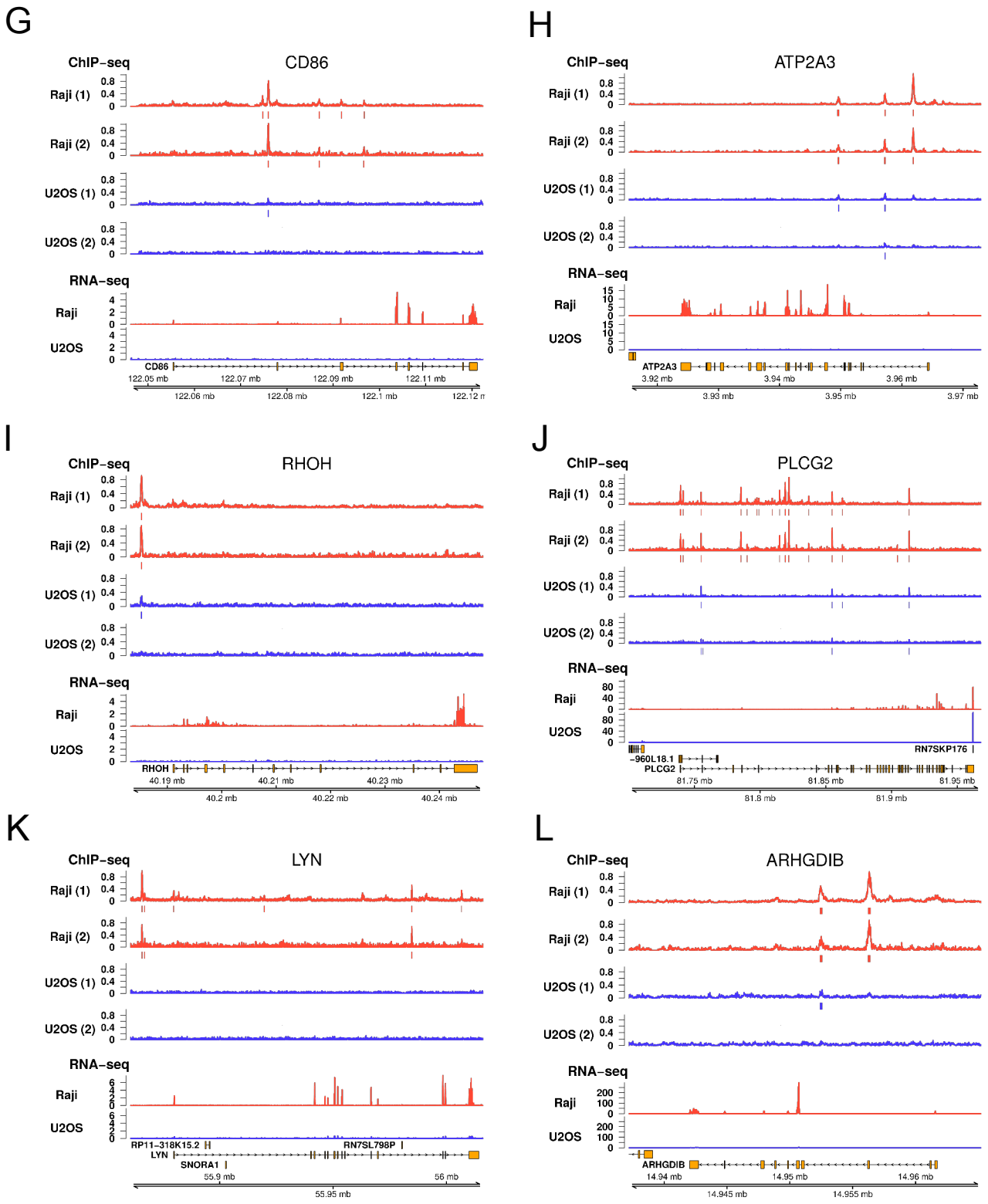


Figure S9

Figure S9

ChIP-seq (replicates shown separately) and RNA-seq (mean of 4 replicates) read coverage (in counts per million) for example genes. Identified peaks are shown as rectangles below the corresponding ChIP-seq sample. Genomic coordinates and gene annotation (boxes=exons, lines=introns, strand indicated by arrowheads) are shown in the bottom row. (**A,B**) Consistent ZNF768 binding in both cell types for the genes Solute Carrier Family 1 Member 5 (SLC1A5) and Mitochondrial Ribosomal Protein S5 (MRPS5). (**C**-**F**) Strong peaks for ZNF768 were associated with the promoter region of the ID1 and SNPH genes and the gene body of the ANXA2 and ALDH7A1 genes in U2OS cells, but are or only faintly visible in Raji cells. (**G**-**L**) Peaks were detected in Raji cells for the genes CD86, ATP2A3, RHOH, PLCG2, LYN, and ARHGDIB. No or weak peaks were identified in U2OS.

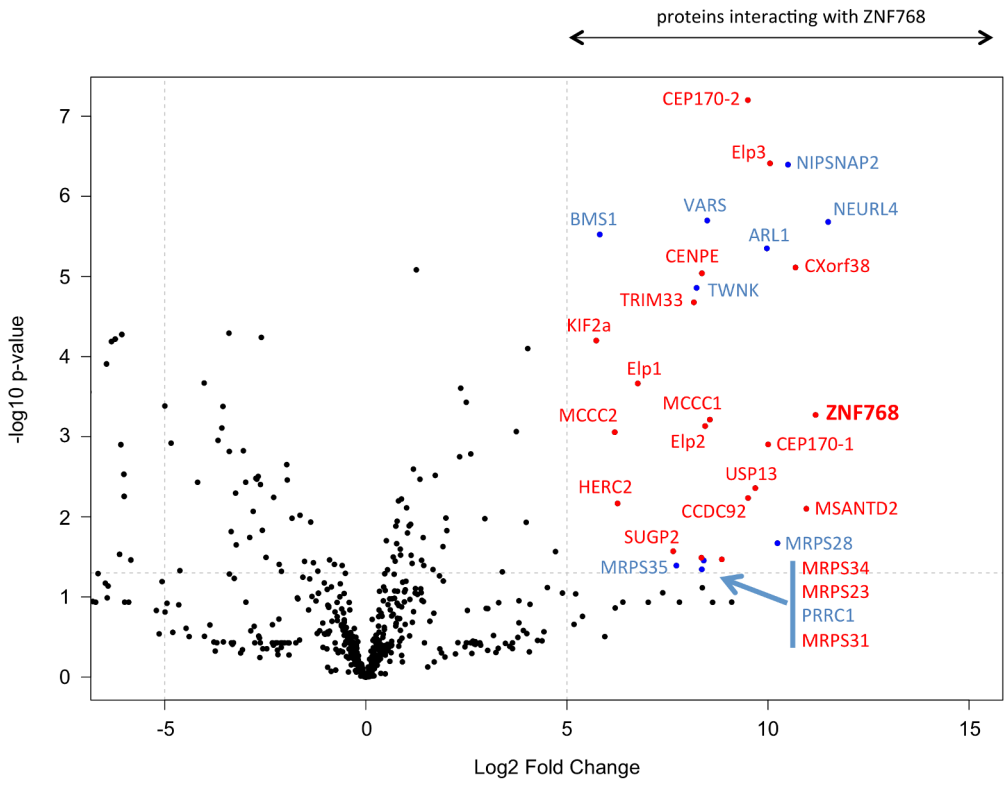


Figure S10

Volcano plot of the ZNF768-specific interactome of Raji cells. Interactions of proteins with log fold change higher than 5 and p-value >0.05 are indicated. Factors depicted in red were detected also among the 30 best interactors in U2OS cells. Dots on the left hand site indicate interactors of a control antibody directed against the nucleolar protein Pes1. Complete list of interactors is shown in Supplementary Table 4.

Article

SOURCE DATA | TRANSPARENT PROCESS | OPEN ACCESS

EMBO reports

CDK12 controls G1/S progression by regulating RNAPII processivity at core DNA replication genes

Anil Paul Chirackal Manavalan[1], Kveta Pilarova[1], Michael Kluge[2], Koen Bartholomeeusen[1,†], Michal Rajecky[1], Jan Oppelt[1], Prashant Khirsariya[3,4], Kamil Paruch[3,4], Lumir Krejci[4,5,6], Caroline C Friedel[2] & Dalibor Blazek[1,*]

Abstract

CDK12 is a kinase associated with elongating RNA polymerase II (RNAPII) and is frequently mutated in cancer. CDK12 depletion reduces the expression of homologous recombination (HR) DNA repair genes, but comprehensive insight into its target genes and cellular processes is lacking. We use a chemical genetic approach to inhibit analog-sensitive CDK12, and find that CDK12 kinase activity is required for transcription of core DNA replication genes and thus for G1/S progression. RNA-seq and ChIP-seq reveal that CDK12 inhibition triggers an RNAPII processivity defect characterized by a loss of mapped reads from 3′ends of predominantly long, poly(A)-signal-rich genes. CDK12 inhibition does not globally reduce levels of RNAPII-Ser2 phosphorylation. However, individual CDK12-dependent genes show a shift of P-Ser2 peaks into the gene body approximately to the positions where RNAPII occupancy and transcription were lost. Thus, CDK12 catalytic activity represents a novel link between regulation of transcription and cell cycle progression. We propose that DNA replication and HR DNA repair defects as a consequence of CDK12 inactivation underlie the genome instability phenotype observed in many cancers.

Keywords CDK12; G1/S; CTD Ser2 phosphorylation; premature termination and polyadenylation; tandem duplications
Subject Categories Cell Cycle; Chromatin, Transcription, & Genomics
DOI 10.15252/embr.201847592 | Received 15 December 2018 | Revised 9 June 2019 | Accepted 24 June 2019
EMBO Reports (2019) e47592

Introduction

Transcription of protein-coding genes is mediated by RNA polymerase II (RNAPII) and represents an important regulatory step of many cellular processes. RNAPII directs gene transcription in several phases, including initiation, elongation, and termination [1–3]. The C-terminal domain (CTD) of RNAPII contains repeats of the heptapeptide YSPTSPS, and phosphorylation of the individual serines within these repeats is necessary for individual steps of the transcription cycle [4,5]. Phosphorylation of RNAPII Ser2 is a hallmark of transcription elongation, whereas phosphorylation of Ser5 correlates with initiating RNAPII [1,6]. Various kinases have been implicated in CTD phosphorylation [7–10], and the kinase CDK12 is thought to phosphorylate predominantly Ser2 [11–18]. These findings were based on the use of phospho-CTD specific antibodies combined with various experimental approaches including *in vitro* kinase assays, long-term siRNA-mediated depletion of CDK12 from cells or application of the CDK12 inhibitor THZ531. However, each of these experiments has caveats with respect to the physiological relevance. The specific impact of a short-term CDK12-selective inhibition on CTD phosphorylation and genome-wide transcription in cells remains an important question to be addressed.

CDK12 and cyclin K (CCNK) are RNAPII- and transcription elongation-associated proteins [11,12,19]. CDK12 and its homolog CDK13 (containing a virtually identical kinase domain) associate with CCNK to form two functionally distinct complexes CCNK/CDK12 and CCNK/CDK13 [11,12,16,20]. Transcription of several core homologous recombination (HR) DNA repair genes, including *BRCA1, FANCD2, FANCI,* and *ATR,* is CDK12-dependent [11,16,21–23]. In agreement, treatment with low concentrations of THZ531 resulted in down-regulation of a subset of DNA repair pathway genes. Higher concentrations led to a much wider transcriptional defect [17]. Mechanistically, it has been suggested that CCNK is recruited to the promoters of DNA damage response genes such as *FANCD2* [24]. Other studies using siRNA-mediated CDK12 depletion showed diminished 3′end processing of *C-MYC* and *C-FOS* genes [18,25]. Roles for CDK12 in other co-transcriptionally regulated processes such as alternative or last exon splicing have also been

1 Central European Institute of Technology (CEITEC), Masaryk University, Brno, Czech Republic
2 Institut für Informatik, Ludwig-Maximilians-Universität München, München, Germany
3 Department of Chemistry, CZ Openscreen, Faculty of Science, Masaryk University, Brno, Czech Republic
4 Center of Biomolecular and Cellular Engineering, International Clinical Research Center, St. Anne's University Hospital, Brno, Czech Republic
5 Department of Biology, Masaryk University, Brno, Czech Republic
6 National Centre for Biomolecular Research, Masaryk University, Brno, Czech Republic
*Corresponding author. Tel: +420 730 588 450; E-mail: dalibor.blazek@ceitec.muni.cz
†Present address: Department of Biomedical Sciences, Institute of Tropical Medicine, Antwerp, Belgium

reported [26–28]. Nevertheless, comprehensive insights into CDK12 target genes and how CDK12 kinase activity regulates their transcription are lacking.

CDK12 is frequently mutated in cancer. Inactivation of CDK12 kinase activity was recently associated with unique genome instability phenotypes in ovarian, breast, and prostate cancers [29–31]. They consist of large (up to 2–10 Mb in size) tandem duplications, which are completely different from other genome alteration patterns, including those observed in *BRCA1*- and other HR-inactivated tumors. Furthermore, they are characterized by an increased sensitivity to cisplatin and thus represent potential biomarker for treatment response [29–33]. Although inactivation of CDK12 kinase activity clearly leads to HR defects and sensitivity to PARP inhibitors in cells [21,34–37], the discovery of the CDK12 inactivation-specific tandem duplication phenotype indicated a distinct function of CDK12 in maintenance of genome stability. The size and distribution of the tandem duplications suggested that DNA replication stress-mediated defect(s) are a possible driving force for their formation [30,31].

Proper transcriptional regulation is essential for all metabolic processes including cell cycle progression [38]. Transition between G1 and S phase is essential for orderly DNA replication and cellular division, and its deregulation leads to tumorigenesis [39]. G1/S progression is transcriptionally controlled by the well-characterized E2F/RB pathway. E2F factors activate transcription of several hundred genes involved in regulation of DNA replication, S phase progression, and also DNA repair by binding to their promoters [40]. Expression of many DNA replication genes (including *CDC6, CDT1, TOPBP1, MCM10, CDC45, ORC1, CDC7, CCNE1/2*), like many other E2F-dependent genes, is highly deregulated in various cancers [41–44]. However, it is not known whether or how their transcription is controlled downstream of the E2F pathway, for instance during elongation.

To answer the above questions, we used a chemical genetic approach to specifically and acutely inhibit endogenous CDK12 kinase activity. CDK12 inhibition led to a G1/S cell cycle progression defect caused by a deficient RNAPII processivity on a subset of core DNA replication genes. Loss of RNAPII occupancy and transcription from gene 3′ends coincided with a shift of the broad peaks of RNAPII phosphorylated at Ser2 from gene 3′ends into the gene body. Our results show that CDK12-regulated RNAPII processivity of core DNA replication genes is a key rate-limiting step of DNA replication and cell cycle progression and shed light into the mechanism of genomic instability associated with frequent aberrations of CDK12 kinase activity reported in many cancers.

Results

Preparation and characterization of AS CDK12 HCT116 cell line

The role of the CDK12 catalytic activity in the regulation of transcription and other cellular processes is poorly characterized. Most of the previous studies of CDK12 involved long-term depletion, which is prone to indirect and compensatory effects [11,12,14,23]. The recent discovery of the covalent CDK12 inhibitor THZ531 made it possible to study CDK12 kinase activity; however, THZ531 also inhibits its functionally specialized homolog CDK13 and transcriptionally related JNK kinases [17].

To overcome these limitations and determine the consequences of specific inhibition of CDK12, we modified both endogenous alleles of *CDK12* in the HCT116 cell line to express an analog-sensitive (AS) version that is rapidly and specifically inhibited by the ATP analog 3-MB-PP1 [45] (Fig 1A). This chemical genetic approach has been used to study other kinases [9,46,47] and was also attempted for CDK12 by engineering HeLa cells carrying a single copy of AS *CDK12* (with the other *CDK12* allele deleted) [48].

We applied CRISPR-Cas technology to mutate the gatekeeper phenylalanine (F) 813 to glycine (G) in both *CDK12* alleles in HCT116 cells (Figs 1A and EV1A). The single-strand oligo donor used as a template for CRISPR-Cas editing introduced a silent GTA>GTT mutation to prevent alternative splicing [48], and a TTT>GGG mutation to implement the desired F813G amino acid change and created a novel *BslI* restriction site used for screening (Fig EV1A). We validated our intact homozygous AS CDK12 HCT116 cell line by several approaches, including allele-specific PCR (Fig EV1B), *BslI* screening (Fig 1B; for expected restriction patterns see Fig EV1A), and Sanger sequencing (Fig 1C and Appendix Fig S1A and B). Immunoprecipitation (IP) of CDK12 from the WT and AS CDK12 HCT116 cells followed by Western blotting showed that equal amounts of CCNK associated with CDK12, and that comparable levels of CDK12 were expressed in both cell lines, confirming the functionality of the AS variant (Fig EV1C). To

Figure 1. Preparation and characterization of AS CDK12 HCT116 cell line.

A Scheme depicting preparation of AS CDK12 HCT116 cell line. Gate keeper phenylalanine (F) and glycine (G) are indicated in red, and adjacent amino acids in CDK12 active site are shown in black letters (left). ATP and ATP analog 3-MB-PP1 are shown as black objects in wild-type (WT) and AS CDK12 (blue ovals), respectively (right).

B Genotyping of AS and WT CDK12 clones. Ethidium bromide-stained agarose gel visualizing PCR products from genomic DNA of AS (AS-PCR) and WT (WT-PCR) CDK12 HCT116 cells and their digest with *BslI* enzyme (indicated as AS- *BslI* and WT- *BslI*). Primer positions and *BslI* restriction sites are depicted at Fig EV1A. Numbers on the left and right indicate DNA marker and DNA fragment sizes, respectively.

C Detailed insight into sequencing of genomic DNA from WT and AS CDK12 HCT116 cell lines. The genomic region in WT and AS CDK12 subjected to genome editing is shown in red rectangle; gate keeper amino acids F and G are in red. The full ~ 500 kb sequence surrounding the edited genomic region is in the Appendix Fig S1A and B.

D Effect of CDK12 inhibition on phosphorylation of the CTD of RNAPII. Western blot analyses of protein levels by the indicated antibodies in AS CDK12 HCT116 cells treated with 5 μM 3-MB-PP1 for indicated times. Long and short exp. = long (4–14 min) and short (10–60 s) exposures, respectively. FUS and tubulin are loading controls. A representative image from three replicates is shown.

E, F Inhibition of CDK12 in AS CDK12 HCT116 cells results in down-regulation of CDK12-dependent HR genes. Graph shows RT–qPCR analysis of relative levels of mRNAs of described genes in AS CDK12 HCT116 (E) and WT CDK12 HCT116 (F) cells treated for indicated times with 3-MB-PP1. mRNA levels were normalized to *HPRT1* mRNA expression and the mRNA levels of untreated control (CTRL) cells were set to 1. n = 3 replicates, error bars indicate standard error of the mean (SEM).

Source data are available online for this figure.

investigate the putative role of CDK12 as a RNAPII CTD kinase, we treated AS CDK12 cells with 3-MB-PP1 or control vehicle for 1, 2, 3, and 6 h and monitored changes in CTD phosphorylation by probing Western blots with phospho-specific antibodies (Figs 1D and EV1D). However, we did not observe any substantial changes in the global levels of phosphorylated Ser2 or Ser5 compared to untreated cells. Only short exposures of Western blots revealed a subtle, but noticeable trend toward accumulation of P-Ser2 after 3 h and P-Ser5

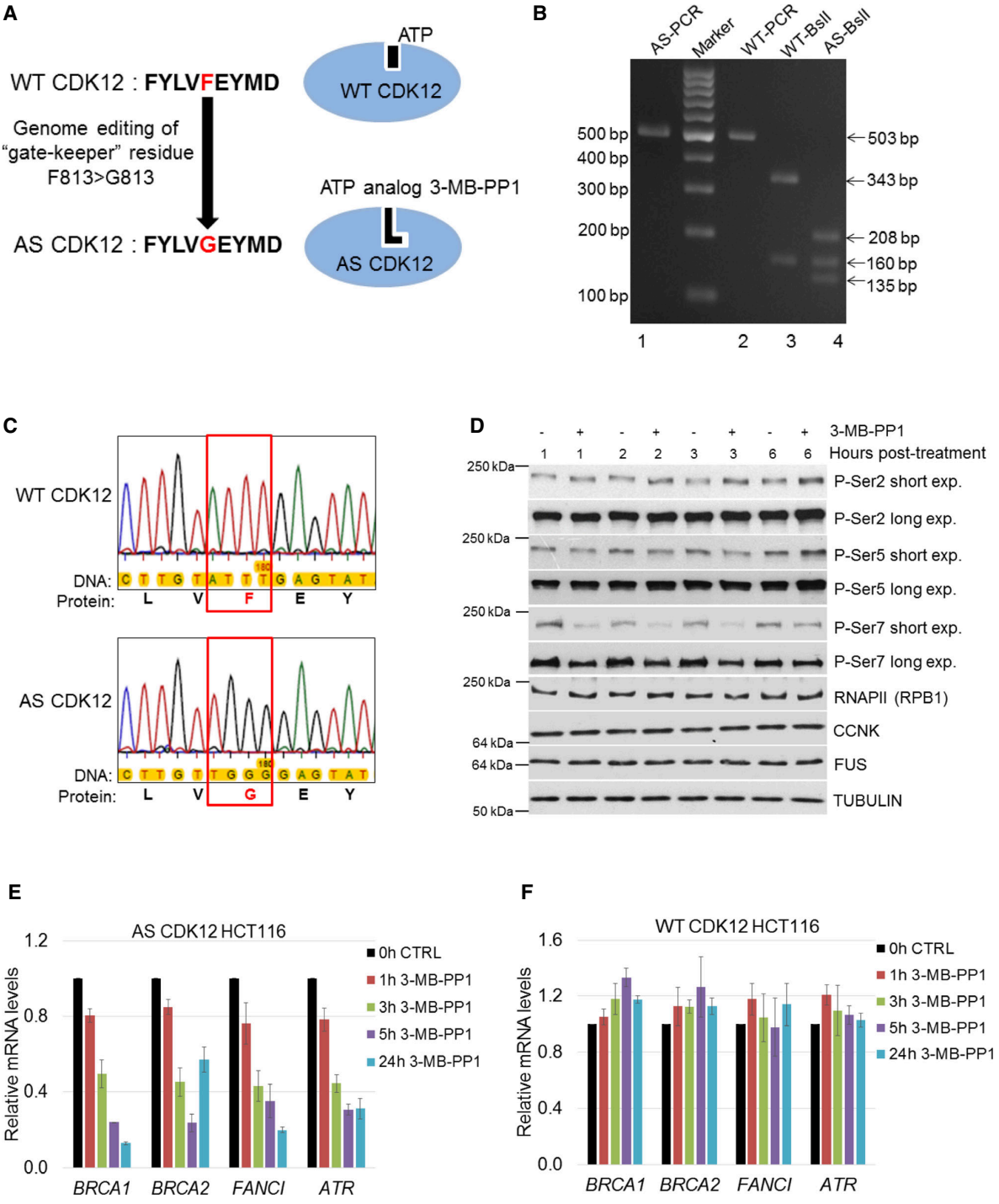


Figure 1.

at 6 h and a slight decrease of P-Ser5 at 1–3 h, respectively, consistent with previous observations in AS CDK12 HeLa cells [48]. Surprisingly, P-Ser7 levels were noticeably diminished starting with 1-h treatment but started recovering at 6 h. To functionally characterize AS CDK12 HCT116 cells, we treated them with 3-MB-PP1 for 1, 3, 5, and 24 h and monitored the expression of DNA repair genes that were previously shown to be regulated by CDK12 (*BRCA1, BRCA2, ATR,* and *FANCI*). We observed rapid down-regulation of all four CDK12-dependent genes (Fig 1E). Importantly, similarly treated WT HCT116 cells showed no down-regulation of these genes (Fig 1F), and RNA-seq of WT HCT116 cells treated with 3-MB-PP1 showed differential expression of only six protein-coding genes compared to the control (data not shown), confirming the absence of off-target effects of the ATP analog on other transcription-related kinases.

In summary, these results demonstrated the generation of a fully functional, homozygous AS CDK12 HCT116 cell line.

CDK12 kinase activity is essential for optimal G1/S progression independently of DNA damage cell cycle checkpoint

In our previous work, we noted that long-term CDK12 depletion leads to an accumulation of cells in G2/M phase, consistent with diminished transcription of CDK12-dependent DNA repair genes and activation of a DNA damage cell cycle checkpoint [11,49]. To determine whether CDK12 kinase activity directly regulates cell cycle progression, we arrested AS CDK12 HCT116 cells at G0/G1 by serum withdrawal for 72 h, released them into serum-containing media in the presence or absence of 3-MB-PP1, and harvested cells for flow cytometry analyses every 6 h after the release (Fig 2A).

In the absence of the inhibitor, the cells entered S phase in ~ 12 h, reached G2/M phase in ~ 18 h, and completed the full cell cycle in ~ 20 h (Fig 2B and C). In contrast, in the presence of 3-MB-PP1, cells started to enter S phase at 18 h, indicating a delay in G1/S progression by 6–9 h. (Fig 2B and C). WT HCT116 cells treated with 3-MB-PP1 showed no defect in cell cycle progression excluding unspecific inhibition of other kinases (Fig EV2A). Importantly, serum-synchronized WT HCT116 cells treated with the CDK12 inhibitor THZ531 (Fig EV2B), as well as AS CDK12 HeLa [48] or AS CDK12 HCT116 cells synchronized by thymidine–nocodazole and inhibited by 3-MB-PP1 also demonstrated the G1/S progression delay (Fig EV2C and data not shown). Thus, the function of CDK12 in optimal G1/S progression appears to be general, rather than cell type- or treatment-specific.

The protein levels of numerous cell cycle regulators fluctuate during cell cycle progression according to their function in a specific phase [38]. To examine whether CDK12 levels change during cell cycle progression, we arrested AS CDK12 HCT116 cells by serum starvation, released them, and analyzed CDK12 proteins by Western blotting (Fig 2D). Strikingly, CDK12 levels were highest during early G0/G1 phase, started to diminish in G1/S transition, reached lowest levels in late S phase, and started to slightly recover in G2/M (Fig 2D). Similar trends, however much less distinct, were observed for CDK13 and CCNK. We verified cell cycle synchronization and individual phases of the cell cycle by the expression of CCNE1 in G1/S and accumulation of CCNA2 in G2/M phases (Fig 2D) and by the flow cytometry DNA content profiles (Fig 2B).

To define when CDK12 kinase activity is needed for early cell cycle progression, serum-synchronized AS CDK12 HCT116 cells were released into serum-containing medium and 3-MB-PP1 was added at various times post-release, ranging from 0 to 12 h. Cell cycle progression was measured by flow cytometry at 16 h post-release (Fig 2E). Whereas treatments at 9 and 12 h had a weak or no effect on the G1/S transition, treatments within 6 h post-release delayed the transition, suggesting that CDK12 kinase activity is needed at very early G1 phase (Fig 2F). Similar results were obtained by flow cytometry analyses of BrdU-labeled cells (Fig 2G). As an additional approach, we released cells in the presence and absence of 3-MB-PP1 and washed away 3-MB-PP1 after 2, 3, 4, and 5 h (Fig EV2D). When the inhibitor was washed away between 2 and 5 h, the cells were able to progress to S phase comparably to untreated cells (Fig EV2E), indicating the requirement of CDK12 kinase activity in very early G1 phase for optimal G1/S progression.

As long-term CDK12 depletion causes down-regulation of DNA repair genes resulting in endogenous DNA damage [11,23], we asked whether the observed G1/S delay upon CDK12 inhibition was

Figure 2. CDK12 kinase activity is essential for optimal G1/S progression independently of DNA damage cell cycle checkpoint.

A Experimental outline. AS CDK12 HCT116 cells were arrested by serum starvation for 72 h and released into the serum-containing medium with or without 3-MB-PP1. DNA content was analyzed by flow cytometry at indicated time points after the release.

B CDK12 kinase activity is needed for G1/S progression in cells arrested by serum starvation. Flow cytometry profiles of control (−3-MB-PP1) or inhibitor (+3-MB-PP1) treated cells from the experiment depicted in Fig 2A. The red arrow points to the onset of the G1/S progression defect in 3-MB-PP1-treated cells. To better visualize the G1/S delay in the presence of the inhibitor, the 24-h time point is also shown. n = 3 replicates; representative result is shown.

C Quantification of cells (%) in individual cell cycle phases based on flow cytometry profiles of the representative replicate in Fig 2B.

D CDK12 protein levels peak in the G0/G1 phase of the cell cycle. Western blots show levels of proteins at indicated time points after the release of serum-starved AS CDK12 HCT116 cells. Corresponding cell cycle phases are depicted above time points. A representative Western blot from three replicates is shown.

E Experimental outline. AS CDK12 HCT116 cells were arrested by serum starvation for 72 h and released into the serum-containing medium. 3-MB-PP1 was either added or not at indicated time points after the release. Propidium iodide- or BrdU-stained DNA content was measured by flow cytometry at 16 h after the release. Note, that for the BrdU staining the 3-MB-PP1 was added only at the time of the release (0 h) and 3, 4, 5, and 6 h after the release.

F, G Inhibition of CDK12 in early G1 perturbs normal cell cycle progression. Quantification of cells (%) in cell cycle phases from flow cytometry profiles of propidium iodide (F)- and BrdU (G)-labeled cells upon addition of 3-MB-PP1 at indicated time points after serum addition in the experiment depicted in Fig 2E. CTRL in Fig 2G = control sample without 3-MB-PP1. n = 3 replicates, representative result is shown.

H Short-term CDK12 inhibition does not activate DNA damage checkpoints. Western blot analyses of phosphorylation of depicted DNA damage response markers upon inhibition of CDK12 for indicated times. CPT corresponds to 5 μM camptothecin. A representative Western blot from three replicates is shown. FUS is a loading control.

Source data are available online for this figure.

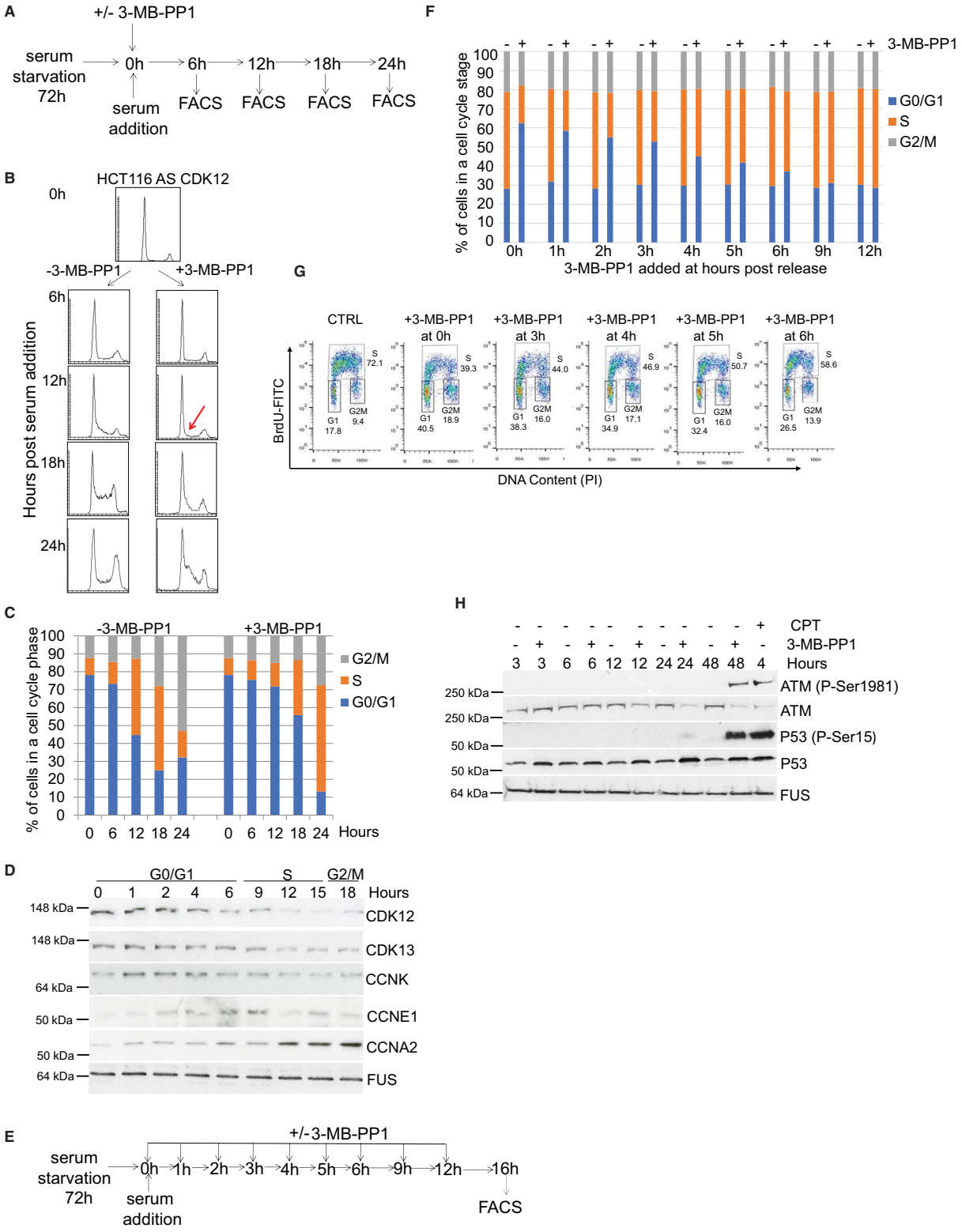

A
+/- 3-MB-PP1
serum starvation 72h
0h
6h
12h
18h
24h
serum addition
FACS
B
HCT116 AS CDK12
0h
-3-MB-PP1
+3-MB-PP1
6h
12h
18h
24h
Hours post serum addition
C
-3-MB-PP1
+3-MB-PP1
% of cells in a cell cycle phase
G2/M
S
G0/G1
Hours
D
G0/G1
S
G2/M
Hours
148 kDa
CDK12
148 kDa
CDK13
64 kDa
CCNK
50 kDa
CCNE1
50 kDa
CCNA2
64 kDa
FUS
E
+/-3-MB-PP1
serum starvation 72h
serum addition
FACS
F
3-MB-PP1
% of cells in a cell cycle stage
G0/G1
S
G2/M
3-MB-PP1 added at hours post release
G
CTRL
+3-MB-PP1 at 0h
+3-MB-PP1 at 3h
+3-MB-PP1 at 4h
+3-MB-PP1 at 5h
+3-MB-PP1 at 6h
BrdU-FITC
DNA Content (PI)
H
CPT
3-MB-PP1
Hours
250 kDa
ATM (P-Ser1981)
250 kDa
ATM
50 kDa
P53 (P-Ser15)
50 kDa
P53
64 kDa
FUS

Figure 2.

due to secondary activation of DNA damage cell cycle checkpoints [50]. However, the levels of phosphorylated P-ATM and P-P53, markers of an activated DNA damage pathway, increased in cells only after 48-h inhibition of CDK12 (Fig 2H), coincident with onset of endogenous DNA damage upon long-term CDK12 depletion [11]. These data suggest that the delay in G1/S progression is independent of secondary activation of DNA damage pathways.

CDK12 catalytic activity controls expression of core DNA replication genes

CDK12 is associated with the transcription of specific genes, particularly DNA repair genes [11,22,23]. We hypothesized that CDK12 catalytic activity is also needed for the expression of genes regulating G1/S progression. To test this hypothesis, we synchronized AS CDK12 HCT116 cells by serum starvation, released them into serum-containing media with or without 3-MB-PP1, and isolated RNA after 5 h ($n = 3$ independent replicates). We then performed 3′end RNA-seq with poly(A)-selected RNA. CDK12 inhibition resulted in the significant differential expression of 2,102 genes ($-1 > \log2$ fold-change > 1, $P < 0.01$), including 611 up-regulated and 1,491 down-regulated genes (Fig 3A and Dataset EV1).

Gene Ontology (GO) enrichment analysis of the down-regulated genes identified high enrichment not only of DNA repair mechanisms (Fig 3B, FDR q-value ≤ 0.05), but also of DNA replication and cell cycle processes (Fig 3B, in red frame). Comparable processes were found to be associated with down-regulation using gene set enrichment analysis (GSEA) [51] (Fig EV3A, in red frames). Manual inspection of the corresponding processes revealed reduced expression of most genes involved in the activation and formation of replication origin recognition complexes and pre-replication complexes (Figs 3C and EV3B). Assembly of these complexes and their activation in early G1 phase are essential for DNA replication and cell cycle progression [52]. Using RT–qPCR, we confirmed that several of these DNA replication genes were down-regulated upon CDK12 inhibition in early G1 phase (Fig 3D). In contrast, mRNA expression of control non-regulated genes but also genes inducible during G1 phase did not change significantly (Fig EV3C). These data indicate that CDK12 inhibition specifically disrupts the expression of its target genes, rather than general transcription, and suggest that CDK12 regulates DNA replication and cell cycle progression by controlling the expression of a subset of genes.

To determine whether the decrease in the transcript levels upon CDK12 inhibition is a result of decreased mRNA stability, we performed transcription inhibition using actinomycin D (ActD; Fig EV3D). Comparison of the degradation rates after transcription shut-off on select DNA repair and replication transcripts in cells either treated or not with 3-MB-PP1 revealed no difference in the relative mRNA stability (Fig EV3D). We therefore conclude that CDK12 inhibition does not influence mRNA half-lives of its target genes.

To elucidate whether the CDK12-dependent decrease in transcript levels of the DNA replication genes corresponds to lower protein levels during G1/S phase, we serum synchronized cells and released them in the absence or presence of 3-MB-PP1 and evaluated lysates after 3, 6, 9, 12, and 15 h. The tested proteins were selected based on antibody availability and their involvement in the formation and activation of origin recognition and pre-replication complexes [52]. We found that the levels of TOPBP1, CDC6, CDT1, MTBP, and CCNE2 proteins were reduced after 6 h of CDK12 inhibition compared to untreated controls, and CDC7 and ORC2 were reduced after 9 and 12 h inhibition, respectively (Figs 3E and EV3E). In contrast, the levels of ORC3, CCNE1, and GINS4 were not significantly affected (Figs 3E and EV3E). Of note, depletion of CDK12 regulatory subunit CCNK in asynchronous cells also resulted in decrease of mRNA and protein levels of the DNA replication genes (Fig EV3F and G).

Assembly of origin recognition and pre-replication complexes on the chromatin in early G1 phase and pre-replication complex activation in G1/S phase (Fig 3C) are prerequisite for the start of DNA replication [39,52]. To examine whether the reduced expression of DNA replication factors upon inhibition of CDK12 affects their loading to and association with chromatin in early cell cycle phases, we isolated the cellular chromatin fraction [53]. Cells were synchronized by serum starvation, released into media with or without 3-MB-PP1, and harvested every 3 h for 24 h, and chromatin-bound ORC6, CDC6, and CDT1 were followed by Western blotting. Indeed,

Figure 3. CDK12 catalytic activity controls expression of core DNA replication genes.

A CDK12 inhibition results in differential expression of a subset of genes. Comparison of log2 fold-changes versus log2 mean expression in 3′end RNA-seq data shows differentially regulated genes after inhibition of CDK12. Down- (log2 fold-change < -1) and up-regulated (log2 fold-change > 1) genes are shown in blue and red, respectively.

B CDK12 inhibition down-regulates DNA damage- and cell cycle-related genes. GO analysis using the Gorilla webserver of enriched cellular functions in 1,491 genes down-regulated (log2 fold-change < -1.0; $P < 0.01$) in 3′end RNA-seq data upon CDK12 inhibition. Functions related to DNA replication and cell cycle are marked by the red rectangle.

C Outline of formation and activation of DNA replication complexes in G1/S phase. Origin recognition, pre-replication, and pre-initiation complexes are depicted; genes dependent on CDK12 kinase activity (log2 fold-change < -0.85; $P < 0.01$) are shown in red.

D Validation of RNA-seq for select DNA replication genes by RT–qPCR. Graph shows relative levels of mRNAs of described genes in serum arrested and released (0 h G0/G1) AS CDK12 HCT116 cells either treated (3-MB-PP1) or not (CTRL) with the inhibitor for indicated times after the release. mRNA levels were normalized to *B2M* mRNA expression, and mRNA levels for each gene at the time of release (0 h) were set as 1. $n = 3$ replicates, error bars indicate SEM.

E Protein levels of core DNA replication factors are dependent on the CDK12 kinase activity. Western blot analyses of protein expression by the depicted antibodies in serum synchronized and released (0 h) cells either treated or not with 3-MB-PP1 for the indicated times after the release. FUS is a loading control. A representative Western blot of three replicates is shown.

F CDK12 inhibition affects loading of CDC6 and CDT1 DNA replication factors to chromatin. Western blotting analyses of chromatin association of the indicated DNA replication factors in serum synchronized and released AS CDK12 HCT116 cells treated or not with 3-MB-PP1 for the indicated times. Histone H2A serves as a loading control of chromatin fractions. A = asynchronous cells, 0 h = time of release. A representative Western blot of three replicates is shown.

Source data are available online for this figure.

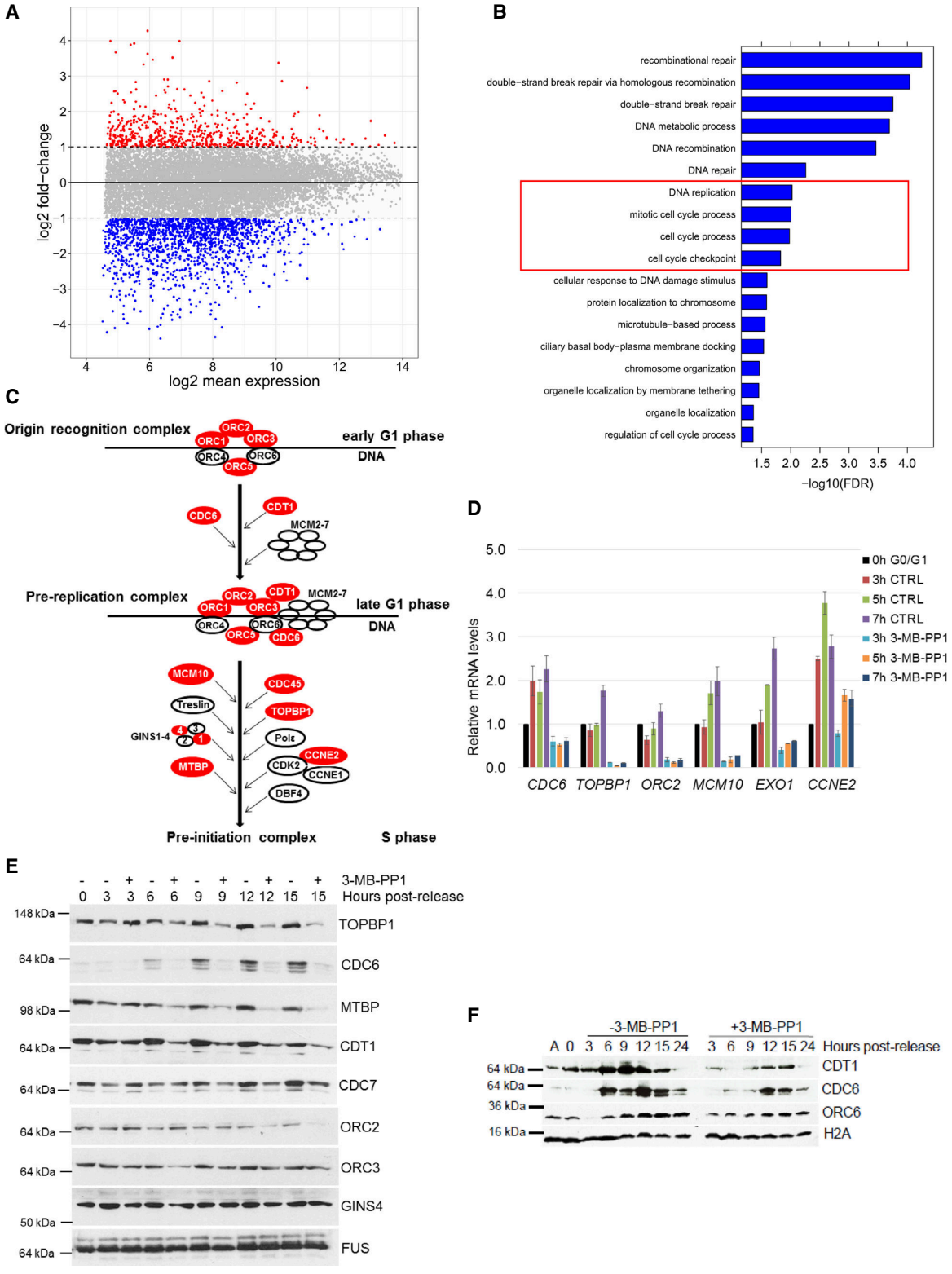

A
log2 fold−change
log2 mean expression
B
recombinational repair
double−strand break repair via homologous recombination
double−strand break repair
DNA metabolic process
DNA recombination
DNA repair
DNA replication
mitotic cell cycle process
cell cycle process
cell cycle checkpoint
cellular response to DNA damage stimulus
protein localization to chromosome
microtubule−based process
ciliary basal body−plasma membrane docking
chromosome organization
organelle localization by membrane tethering
organelle localization
regulation of cell cycle process
−log10(FDR)
C
Origin recognition complex
early G1 phase
DNA
Pre-replication complex
late G1 phase
Pre-initiation complex
S phase
D
Relative mRNA levels
0h G0/G1
3h CTRL
5h CTRL
7h CTRL
3h 3-MB-PP1
5h 3-MB-PP1
7h 3-MB-PP1
CDC6 TOPBP1 ORC2 MCM10 EXO1 CCNE2
E
3-MB-PP1
Hours post-release
TOPBP1
CDC6
MTBP
CDT1
CDC7
ORC2
ORC3
GINS4
FUS
F
-3-MB-PP1
+3-MB-PP1
Hours post-release
CDT1
CDC6
ORC6
H2A

Figure 3.

we found that CDK12 inhibition diminished and delayed the loading of CDC6 and CDT1 proteins onto chromatin relative to control (Fig 3F, compare points 6–15 h post-release in the presence or absence of 3-MB-PP1).

Altogether, our results show that CDK12 catalytic activity is required for the expression of several crucial DNA replication genes including *CDC6, CDT1,* and *TOPBP1*. CDK12 inhibition diminishes levels of these proteins, disrupting their loading on chromatin and formation of pre-replication complexes, which delays G1/S progression (Fig 2B).

A tight interplay between CDK12 kinase activity, expression of DNA replication genes, cell cycle progression, and genome stability

To further clarify the interplay between CDK12 kinase activity, DNA replication gene expression, and cell cycle progression, we performed an inhibitor wash off experiment (Fig 4A). We employed RT–qPCR and Western blotting to monitor the expression of DNA replication genes, and flow cytometry to monitor cell cycle progression. Consistent with our observations so far, CDK12 inhibition induced a strong decrease in mRNA (Fig 4B) and protein levels (Fig 4C) of DNA replication genes, and delayed S phase entry (Fig 4D). Notably, washing off the inhibitor at various times between 1 and 5 h after the release led to progressive rescue of mRNA (Fig 4B), protein expression (Fig 4C), and a gradual normalization of cell cycle progression (Fig 4D). In agreement, the inhibitor wash off after 1 h of treatment restored the chromatin association of CDC6 and CDT1 compared to the no-wash controls (Fig 4E). Altogether, these experiments revealed a tight interplay between CDK12 catalytic activity, DNA replication factors expression, and their chromatin loading and G1/S progression.

Considering this critical role for CDK12 kinase activity in G1/S progression, we asked if longer-term CDK12 inhibition affects replication of asynchronous cellular populations. Treatment of AS CDK12 HCT116 cells with 3-MB-PP1 for 24 h followed by flow cytometry analyses of BrdU-labeled cells revealed a 15% decrease of S phase stage replicating cells in comparison to the untreated control (Fig 4F). Cellular replication was affected much more strongly after 48 h of 3-MB-PP1 treatment resulting in a 35% decrease in the number of replicating cells and a 34% accumulation of G1 cells compared to the control (Fig 4F).

As disruption of every CDK12-dependent process described so far (DNA replication, cell cycle progression, DNA damage repair) is predicted to trigger DNA damage and genome instability [54], we asked whether inhibition of CDK12 would lead to increased chromosomal abnormalities. Therefore, we treated AS CDK12 HCT116 cells with 3-MB-PP1 for 24 and 48 h and performed a chromosomal aberration assay (Fig 4G and H). CDK12 inhibition led to a 3- to 4-fold increase in the number of chromosomal aberrations (e.g., gaps, chromosomal exchanges, DNA breaks, and single/bi-chromatid breakage (frag/difrag)) when compared to cells with normal CDK12 kinase activity. The increase was comparable to cells treated with hydroxyurea (Fig 4H). This result is consistent with fundamental roles of CDK12 kinase activity in maintenance of genome stability.

Altogether, these findings support the existence of a tight functional link between CDK12 catalytic activity, the regulation of genes involved in DNA replication and of cell cycle progression, and consequent DNA damage/genome instability in cells.

Inhibition of CDK12 leads to diminished RNAPII processivity on down-regulated genes

Next, we aimed to determine what transcriptional mechanism(s) affects expression of CDK12-dependent genes. It is well established that transcription of many DNA replication, cell cycle, and DNA repair genes is specifically regulated by the E2F/RB pathway. Since many CDK12-dependent DNA replication and DNA repair genes are dependent on E2F transcription factors [11,40], we examined CDK12-dependent recruitment of E2F1 and E2F3 to the promoters of DNA replication genes by ChIP-qPCR. However, we did not observe any significant change between CDK12-inhibited cells and controls (Fig EV4A). E2Fs are needed for recruitment of RNAPII to its target genes and their activation. However, CDK12 inhibition did not affect recruitment of RNAPII to the promoters of E2F-dependent genes (Fig EV4B; see below for RNAPII ChIP-seq and RNA-seq

Figure 4. A tight interplay between CDK12 kinase activity, expression of DNA replication genes, cell cycle progression, and genome stability.

A Experimental outline. AS CDK12 HCT116 cells were arrested by serum starvation for 72 h and released into the serum-containing medium with (+) or without (−) 3-MB-PP1. 3-MB-PP1 was washed away and replaced with fresh medium at indicated times after the release and samples were subject to RT–qPCR, Western blotting, and flow cytometry analyses at 7, 12, and 15 h after the release, respectively. Note that shown wash away time points (2, 3, 4, 5 h) are valid for RT–qPCR only, for Western blotting and flow cytometry 1, 2, 3, 5 h and 1, 3, 5, 7 h wash away time points were applied, respectively. All experiments were performed in at least three replicates.

B–D Removal of CDK12 inhibitor in early G1/S rescues replication gene expression and cell cycle progression. RT–qPCR (B), Western blotting (C), and flow cytometry analyses (D) of replication gene mRNA, protein levels, and cell cycle progression, respectively. RT–qPCR, Western blotting, and flow cytometry analyses were performed 7, 12, and 15 h post-release, respectively. CTRL = control samples without the 3-MB-PP1. In B, $n = 3$ and error bars indicate SEM. In (C, D) representative images from three biological replicates are shown.

E Rescued loading of CDC6 and CDT1 on chromatin after removal of CDK12 inhibitor. Western blot analyses of chromatin fractions of serum-starved AS CDK12 HCT116 cells treated with 3-MB-PP1 for 6 or 9 h or with the inhibitor washed off after 1 h of treatment. CTRL corresponds to cells not treated with the inhibitor at the time of the serum addition. All cells were harvested either 6 or 9 h after the serum addition. Histone H2A serves as a loading control of chromatin fractions, and studied DNA replication factors are indicated. A representative image of three replicates is shown.

F Inhibition of CDK12 kinase activity in cycling cells leads to decreased numbers of actively replicating cells. Asynchronous AS CDK12 HCT116 cells were grown for 24 and 48 h in the presence or absence of 3-MB-PP1, and replicating BrdU-stained cells were quantified by FACS analyses. CTRL = control samples without the 3-MB-PP1. A representative image of three replicates is shown.

G, H Prolonged CDK12 inhibition causes chromosomal aberrations in cells. Specific chromosomal aberrations in cells treated with 3-MB-PP1 (24 or 48 h), 4 mM hydroxyurea (5 h), or control solvent (CTRL) were identified by microscopy. A representative image from three biological replicates is shown (G). Total numbers of chromosomal aberrations per hundred cells of the representative replicate in (G) are quantified (H).

Source data are available online for this figure.

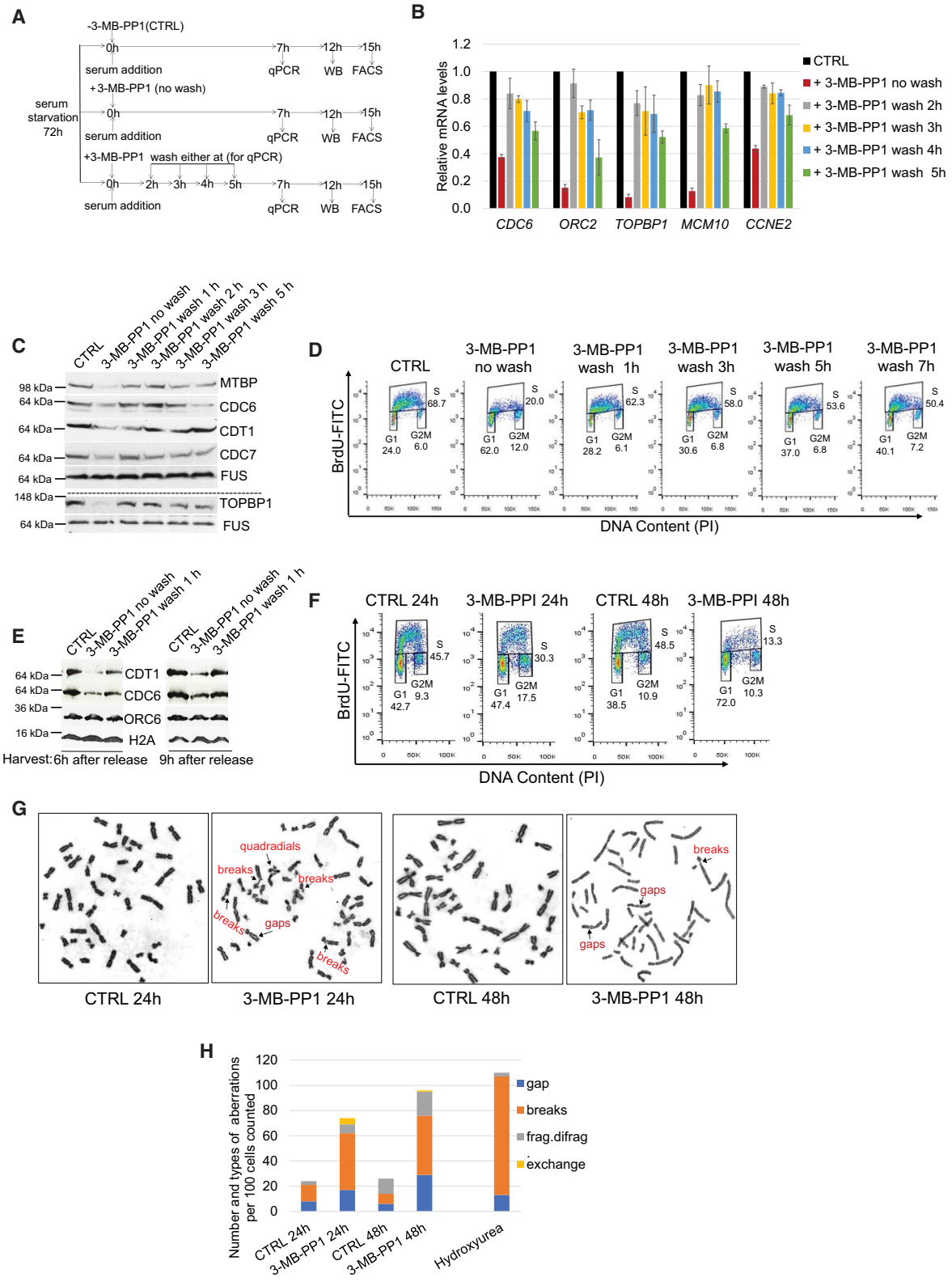

A
-3-MB-PP1(CTRL)
0h
7h
12h
15h
serum addition
qPCR
WB
FACS
+3-MB-PP1 (no wash)
serum starvation 72h
+3-MB-PP1 wash either at (for qPCR)
2h
3h
4h
5h
B
Relative mRNA levels
CTRL
+ 3-MB-PP1 no wash
+ 3-MB-PP1 wash 2h
+ 3-MB-PP1 wash 3h
+ 3-MB-PP1 wash 4h
+ 3-MB-PP1 wash 5h
CDC6
ORC2
TOPBP1
MCM10
CCNE2
C
CTRL
3-MB-PP1 no wash
3-MB-PP1 wash 1 h
3-MB-PP1 wash 2 h
3-MB-PP1 wash 3 h
3-MB-PP1 wash 5 h
MTBP
CDC6
CDT1
CDC7
FUS
TOPBP1
D
3-MB-PP1 no wash
3-MB-PP1 wash 1h
3-MB-PP1 wash 3h
3-MB-PP1 wash 5h
3-MB-PP1 wash 7h
BrdU-FITC
DNA Content (PI)
E
Harvest: 6h after release
9h after release
F
CTRL 24h
3-MB-PPI 24h
CTRL 48h
3-MB-PPI 48h
G
quadradials
breaks
gaps
CTRL 24h
3-MB-PP1 24h
CTRL 48h
3-MB-PP1 48h
H
Number and types of aberrations per 100 cells counted
gap
breaks
frag.difrag
exchange
Hydroxyurea

Figure 4.

experiments, respectively). Thus, these data suggest that CDK12 acts downstream of the E2F/RB pathway.

CDK12 has been implicated in the transcription of a subset of genes via phosphorylation of RNAPII, particularly on Ser2 and Ser5 in the CTD [11–13,16,17]. To uncover a role for CDK12 kinase activity in transcription of genes on a genome-wide level during early G1 phase, we performed ChIP-seq using antibodies for RNAPII, P-Ser2, and P-Ser5, coupled with nuclear RNA-seq (n = 3 replicates each). In contrast to 3′end RNA-seq, nuclear RNA-seq allowed analyzing changes in RNA processing and splicing and also measuring non-polyadenylated RNAs. We synchronized AS CDK12 HCT116 cells by serum starvation for 72 h, released them into serum-containing media with or without 3-MB-PP1, and collected samples at 4.5 h post-release for ChIP-seq and nuclear RNA-seq.

Nuclear RNA-seq revealed significant differential expression of 1,617 genes ($-1 >$ log2 fold-change > 1, $P < 0.01$), including 1,277 genes with diminished and 340 genes with increased expression (Fig EV5A and Dataset EV2), consistent with our observation that only a subset of genes are regulated by CDK12 kinase activity. Log2 fold-changes were highly correlated between 3′end RNA-seq and nuclear RNA-seq (Spearman rank correlation $\rho = 0.78$, Fig EV5A), and we observed significant overlap between differentially expressed genes in both experiments (Figs 5A and EV5B).

To determine whether this differential expression is due to a transcriptional defect caused by CDK12 inhibition, we analyzed the distribution of RNAPII, P-Ser2, and P-Ser5 ChIP-seq reads from −3 kb of the transcription start site (TSS) to +3 kb of the transcription termination site (TTS). Genes were divided into three groups according to their differential expression after CDK12 inhibition in the nuclear RNA-seq data: up-regulated (log2 fold-change > 1, $P < 0.01$), down-regulated (log2 fold-change < -1, $P < 0.01$), and non-regulated ($-0.1 <$ log2 fold-change < 0.1, $P > 0.01$).

Metagene plots display the expected profile of RNAPII occupancy for all three groups with a peak of paused RNAPII at the promoter (Fig 5B). Strikingly, CDK12 inhibition reduced the relative RNAPII occupancy at the 3′ends of down-regulated genes (Fig 5B). More strongly down-regulated genes had tendency toward a higher reduction in 3′end occupancy (Appendix Fig S2). Little or no occupancy difference was observed for non-regulated and up-regulated genes, respectively (Fig 5B). This phenotype is consistent with an RNAPII elongation/processivity defect at down-regulated genes.

P-Ser5 signal peaked at promoters, consistent with a role in initiating RNAPII [6], and we found that P-Ser5 occupancy was reduced significantly at 3′ends of down-regulated genes and a little at non-regulated genes when CDK12 was inhibited (Fig EV5C). However, P-Ser5 occupancy normalized to RNAPII showed no or very little changes across the three groups of genes after CDK12 inhibition (Appendix Fig S3), providing evidence that observed changes in P-Ser5 signal are only due to changes in RNAPII occupancy.

In control cells, P-Ser2 occupancy was most pronounced on gene bodies with highest enrichment at 3′ends (Fig 5C), consistent with its role in elongation and 3′end processing [6,55,56]. Importantly, in response to CDK12 inhibition, down-regulated genes showed a very strong shift of P-Ser2 occupancy into the gene body and toward the TSS (Fig 5C). The shift toward the gene body was most pronounced in strongly down-regulated genes (Appendix Fig S4). To exclude that the shift in P-Ser2 occupancy was only a consequence of the change in overall RNAPII levels, we also normalized P-Ser2 occupancy profiles to RNAPII levels (Appendix Fig S5). This showed a small but highly significant increase of normalized P-Ser2 occupancy in the gene body and a reduction at gene 3′ends for down-regulated genes and to a lesser degree for non-regulated genes (Appendix Fig S5).

SPT6 binds RNAPII via the CTD linker and stimulates transcription elongation [57–59]. To investigate whether SPT6 and RNAPII association is dependent on CDK12 kinase activity and to correlate the observed changes in RNAPII occupancies with occupancies of this well-characterized elongation factor we performed SPT6 ChIP-seq (n = 3 replicates, Fig EV5D). Metagene plots show the expected profile of SPT6 binding with a peak at the promoter and an increase at 3′ends of genes, which resembles RNAPII profiles (Fig EV5D). CDK12 inhibition reduced relative SPT6 occupancy at the 3′ends of down-regulated genes. Little or no occupancy difference was observed at non-regulated and up-regulated genes, respectively (Fig EV5D). However, SPT6 occupancy normalized to the RNAPII showed little changes for all three gene groups (Appendix Fig S6), indicating that SPT6 travels together with RNAPII on genes and SPT6-RNAPII association is independent of CDK12 kinase activity. In agreement, immunoprecipitation of SPT6 from cells showed no change in the interaction with RNAPII when CDK12 was inhibited (Fig EV5E).

The genome-wide trends in RNAPII, P-Ser2, P-Ser5, and SPT6 occupancies in down-regulated genes were clearly visible at selected CDK12-dependent genes (Fig 5D and E, and Appendix Fig S7A) including DNA replication genes (Appendix Fig S7B and C). Here,

Figure 5. Inhibition of CDK12 leads to diminished RNAPII processivity on down-regulated genes.

A Inhibition of CDK12 affects the expression of similar subsets of genes in nuclear and 3′end RNA-seq data. The Venn diagrams represent the overlap between genes significantly ($P < 0.01$) up- (log2 fold-change > 1) or down-regulated (log2 fold-change < -1) in nuclear and 3′end RNA-seq data.

B, C Genes down-regulated in nuclear RNA-seq after CDK12 inhibition have diminished relative occupancy of RNAPII at their 3′ends and higher relative occupancy of P-Ser2 in their gene bodies. Metagene analyses of RNAPII (B) and P-Ser2 (C) ChIP-seq data (see Materials and Methods). Each transcript was divided into two parts with fixed length (transcription start site (TSS) −3 kb to +1.5 kb and transcription termination site (TTS) −1.5 kb to +3 kb) and a central part with variable length corresponding to the rest of gene body (shown in %). Each part was binned into a fixed number of bins (90/180/90), and average coverage for each bin was calculated for each transcript in each sample. The curve for each transcript was normalized to a sum of one and then averaged first across genes and second across samples. Dotted lines indicate TSS, 1,500 nucleotides downstream of TSS, and 1,500 nucleotides upstream of TTS and TTS. The color track at the bottom of each subfigure indicates the significance of paired Wilcoxon tests comparing the normalized transcript coverages for each bin between untreated (CTRL) cells and cells treated with 3-MB-PP1. P-values are adjusted for multiple testing with the Bonferroni method within each subfigure; color code: red = adjusted P-value $\leq 10^{-15}$, orange = adjusted P-value $\leq 10^{-10}$, yellow = adjusted P-value $\leq 10^{-3}$.

D, E Examples of genes whose transcription processivity and expression is dependent on the CDK12 kinase activity. Nuclear RNA-seq data on the respective strand and RNAPII, P-Ser2, P-Ser5, and SPT6 ChIP-seq data for *MED13* (D), *UBE3C* (E) genes from cells either treated (red) or not (blue, CTRL) with 3-MB-PP1 were visualized with Gviz. Read counts were normalized to the total number of mapped reads per sample and averaged between replicates. Blue and red boxes below the RNA-seq data indicate the 90% distance (see Fig 7D and E and corresponding text) in control and CDK12-inhibited samples, respectively.

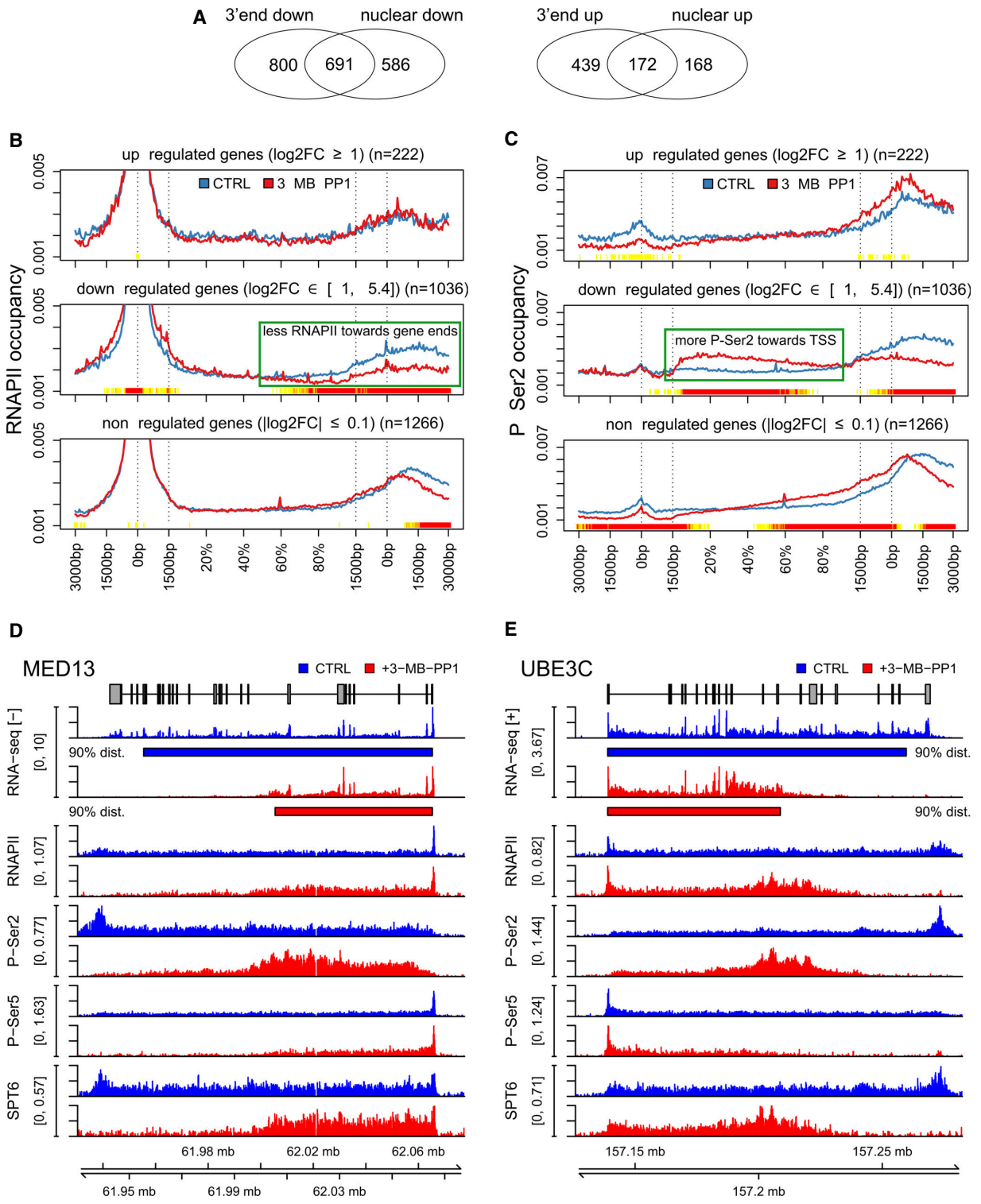

A
3'end down
nuclear down
800
691
586
3'end up
nuclear up
439
172
168
B
up regulated genes (log2FC ≥ 1) (n=222)
CTRL
3 MB PP1
down regulated genes (log2FC ∈ [1, 5.4]) (n=1036)
less RNAPII towards gene ends
non regulated genes (|log2FC| ≤ 0.1) (n=1266)
RNAPII occupancy
0.005
0.001
3000bp
1500bp
0bp
1500bp
20%
40%
60%
80%
1500bp
0bp
1500bp
3000bp
C
up regulated genes (log2FC ≥ 1) (n=222)
down regulated genes (log2FC ∈ [1, 5.4]) (n=1036)
more P-Ser2 towards TSS
non regulated genes (|log2FC| ≤ 0.1) (n=1266)
P Ser2 occupancy
0.007
0.001
D
MED13
CTRL
+3−MB−PP1
RNA−seq [−]
[0, 10]
90% dist.
RNAPII
[0, 1.07]
P−Ser2
[0, 0.77]
P−Ser5
[0, 1.63]
SPT6
[0, 0.57]
61.98 mb
62.02 mb
62.06 mb
61.95 mb
61.99 mb
62.03 mb
E
UBE3C
RNA−seq [+]
[0, 3.67]
[0, 0.82]
[0, 1.44]
[0, 1.24]
[0, 0.71]
157.15 mb
157.25 mb
157.2 mb

Figure 5.

the RNAPII, P-Ser2, P-Ser5, and SPT6 signals ended within the gene body upon CDK12 inhibition rather than after the gene 3′end. Strikingly, nuclear RNA-seq showed that CDK12 inhibition also lead to an earlier termination of transcription of these genes at roughly the genomic location in the gene body where RNAPII occupancy was lost and the broad 3′end peak of P-Ser2 signal appeared upon CDK12 inhibition. This suggests that the apparent down-regulation of the corresponding genes in both the 3′end and nuclear RNA-seq data upon CDK12 inhibition actually represents a shortening of transcripts as a consequence of an RNAPII processivity defect.

Transcript shortening upon inhibition of CDK12

As differential gene expression analysis is based on all reads mapped to exonic regions of a gene, it cannot distinguish between shortening of transcripts, resulting in fewer reads on only some exons, from overall lower transcription levels, resulting in lower levels on all exons.

To address this issue, we analyzed differential exon usage on the nuclear RNA-seq data using DEXSeq, a method to identify relative changes in exons usage [60]. CDK12 inhibition resulted in significant down-regulation of at least one exon for 2,110 genes and significant up-regulation of at least one exon for 1,550 genes ($0 > \log2$ fold-change > 0, $P < 0.01$). A comparison to differentially expressed genes included in the differential exon usage analysis [2,089 down-regulated, 1,822 up-regulated ($0 > \log2$ fold-change > 0, $P < 0.01$)] in nuclear RNA-seq showed an overlap of 924 genes (44% of down-regulated genes) that were both significantly down-regulated in expression and had significantly down-regulated exons (Fig 6A). In contrast, only 123 up-regulated genes (7%) had at least one exon significantly up-regulated. Furthermore, 1,156 genes had both up- and down-regulated exons, i.e., 75% of genes with at least one up-regulated exon and 55% of genes with at least one down-regulated exon. This can be explained by a relative decrease in the use of some exons resulting in a relative increase in the use of other exons of the same gene. Notably, the majority of these genes (59%) were also down-regulated, whereas only 7% were up-regulated.

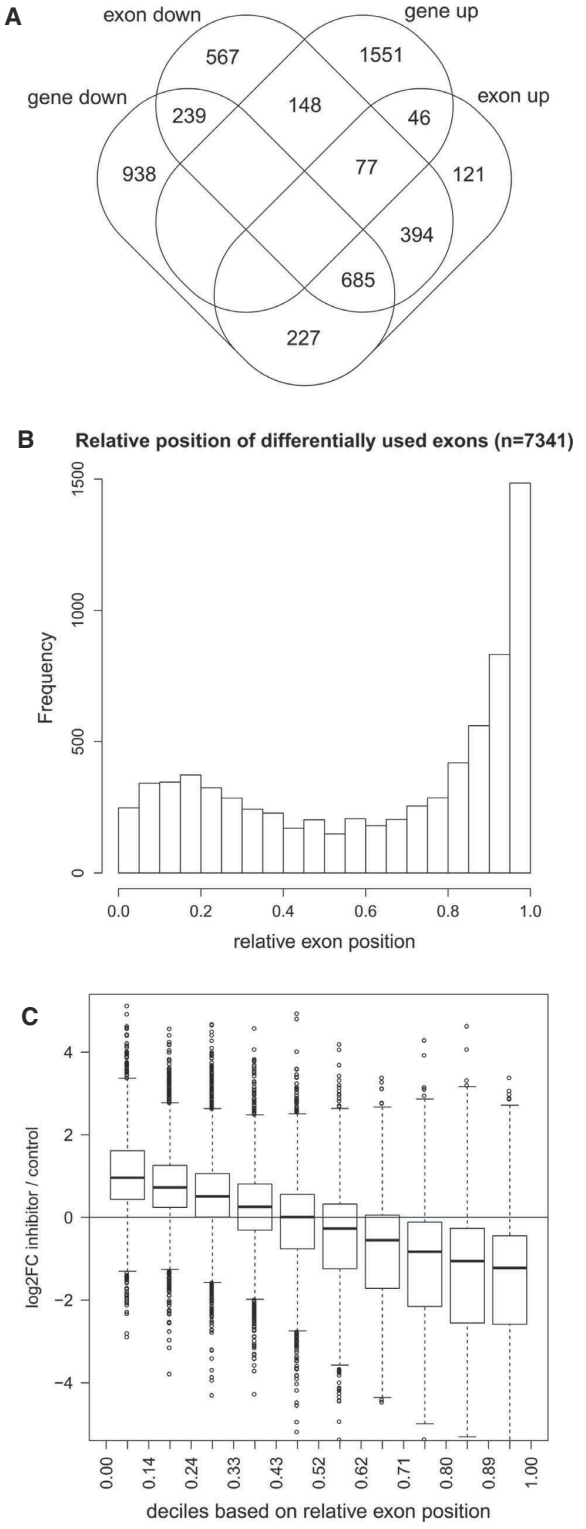


Figure 6. CDK12 inhibition results in transcript shortening of a subset of genes.

A Overlap between down-regulated genes and genes with differential exon usage upon CDK12 inhibition. Venn diagram shows the overlap between significantly differentially expressed genes (identified by DESeq2) and genes with differential exon usage (identified by DEXSeq) in nuclear RNA-seq data ($0 > \log2$ fold-change > 0, $P < 0.01$, restricted to genes included in the DEXSeq analysis).

B Differentially used exons are enriched at gene 3′ends. Graph shows the distribution of the relative genomic position of the exon on the gene (relative exon position: 0 = at gene 5′end, 1 = at gene 3′end) of differentially used exons ($0 > \log2$ fold-change > 0, $P < 0.01$).

C For down-regulated genes with differentially used exons, exons close to the 5′end and 3′end tend to be up- and down-regulated, respectively. Box plots show the log2 fold-change in exon usage after CDK12 inhibition determined by DEXSeq. Exons were grouped into deciles according to their relative exon position. $n = 3$ replicates. The boxes indicate the range between the 25th and 75th percentile (=interquartile range (IQR)) around the median (thick horizontal line) of the distribution. The whiskers (=short horizontal lines at ends of dashed vertical line) extend to the data points at most 1.5 × IQR from the box. Data points outside this range are shown as circles.

To investigate whether differential exon usage of genes reflects shortening of transcripts, we determined the relative exon position of differentially used exons within genes. We found that differentially used exons are highly enriched at 3′end of genes with a slight accumulation also toward gene 5′ends (Fig 6B). Moreover, the relative position of either down- or up-regulated exons showed exclusive accumulation at the gene 3′end and 5′end, respectively (Appendix Fig S8A–C). Down-regulated genes with at least one significantly differentially used exon (1,151 genes) showed a clear trend, with exons up-regulated at the 5′end and down-regulated at the 3′end (Fig 6C). This indicates that these genes are down-regulated because transcripts tend to get shorter in the absence of CDK12 catalytic activity. Notably, down-regulated genes without significantly differentially used exons (45% of down-regulated genes) showed a similar but less pronounced trend (Appendix Fig S8D). In summary, our findings reveal that the observed down-regulation of genes upon CDK12 inhibition generally results from transcript shortening.

When correlating differential exon usage to the ChIP-seq data, we found that genes with down- or up-regulated exons (most of the latter also had down-regulated exons) showed reduced RNAPII occupancy at the 3′end (Appendix Fig S8E) as well as a relative shift of P-Ser2 normalized to RNAPII from the gene 3′end into the gene body (Appendix Fig S8F). Altogether, our results suggest that inhibition of CDK12 kinase activity causes a shift of P-Ser2 from gene 3′ends to gene bodies and diminished RNAPII processivity, consequently leading to shorter transcripts of CDK12-dependent genes. Since P-Ser2 is important for recruitment of splicing factors to the RNAPII CTD [2,61,62], we investigated whether significantly regulated exons in genes not down-regulated might be reflective of alterations in splicing rather than shortening of transcripts. However, the distribution of exon usage changes relative to the position of the exon again showed a trend similar to down-regulated genes with a tendency for down-regulated exons near the gene 3′ends (Appendix Fig S8G). In this case, strong down-regulation of exons was only observed very close to gene 3′ends, suggesting that these genes are only slightly affected by the RNAPII processivity defect (Appendix Fig S8G).

CDK12 kinase activity is required for optimal transcription of long, poly(A)-signal-rich genes

We previously showed that long-term depletion of CDK12 leads to diminished expression of mostly longer genes [11]. To determine whether short-term inhibition of CDK12 kinase predominantly affects RNAPII processivity at longer genes, we sorted genes into deciles based on their length and evaluated the fraction of exons that are differentially used in each gene. We found that longer genes tended to have a larger fraction of differentially used exons (Fig 7A). Similar results were obtained when only the fractions of down-regulated or up-regulated exons were plotted (Appendix Fig S9A and B). This is consistent with the overlap between genes with up- and down-regulated exons, and the scenario that relative down-regulation of some exons leads to relative up-regulation of other exons in the same gene. Accordingly, genes with at least one exon down- or up-regulated tended to be longer than genes with no differentially used exon, but there was no significant difference in gene length between the two groups (Fig 7B). Down-regulated genes also tended to be longer than non-regulated and up-regulated genes (Fig 7C), consistent with the hypothesis that optimal RNAPII processivity and RNA expression in longer genes requires CDK12 catalytic activity. This conclusion is also supported by metagene plots for genes grouped according to gene length, which showed stronger changes for longer genes in RNAPII, P-Ser2, and P-Ser5 ChIP-seq occupancies after CDK12 inhibition (Appendix Figs S10–S12).

To verify that CDK12 catalytic activity controls the processivity of RNAPII predominantly at long genes, we calculated the distance

Figure 7. CDK12 kinase activity is required for optimal transcription of long, poly(A)-signal-rich genes.

A Longer genes tend to have a larger fraction of differentially used exons. Box plot shows the fraction of exons significantly differentially used for 9,026 expressed genes grouped into deciles based on the genomic length (including exons and introns) of their longest transcripts. $n = 3$ replicates. See legend in Fig 6C for the boxplot description.

B Genes with differentially used exons tend to be longer. Box plots show length of genes with no differentially used exons, or at least one exon differentially up-regulated (DEXSeq log2 fold-change ≥ 0, $P < 0.01$) or down-regulated (log2 fold-change ≤ 0, $P < 0.01$). *P*-value from a two-sided Wilcoxon rank sum test comparing median lengths between genes with either up- or down-regulated exons is indicated on top. $n = 3$ replicates. See legend in Fig 6C for the boxplot description.

C Down-regulated genes tend to be longer than not-regulated genes, while up-regulated genes show little difference. Box plots show length of genes with no differential expression ($-0.1 <$ log2 fold-change < 0.1, $P > 0.01$), up-regulated (log2 fold-change ≥ 0, $P < 0.01$), or down-regulated (log2 fold-change ≤ 0, $P < 0.01$) as determined by DESeq2. *P*-values from two-sided Wilcoxon rank sum tests comparing median lengths for up- and down-regulated genes, respectively, to non-regulated genes are indicated on top. $n = 3$ replicates. See legend in Fig 6C for the boxplot description.

D RNAPII processivity is affected not close to but at some distance from the TSS after CDK12 inhibition. The graphs compare the relative distance from the TSS where 10, 50 and 90% of read coverage is identified (=*x*% distance) in control (*x*-axis) against CDK12-inhibited (*y*-axis) cells.

E Transcripts of longer genes are more often impacted by shortening and lose a larger proportion of their length in comparison with shorter genes. The plot shows on the *x*-axis the relative change in the 90% distance (relative Δ90% distance = (90% distance in control − 90% distance in CDK12 inhibited cells)/gene length) and on the *y*-axis the percentage of genes showing a Δ90% distance equal or greater than the value on the *x*-axis. Positive and negative relative Δ90% distances on the *x*-axis indicate a shortening or extension of transcripts, respectively, after CDK12 inhibition. Genes were divided into quintiles according to gene length, and curves for quintiles are shown separately. Dotted and dashed horizontal lines indicate the percentage of genes in each quintile with a transcript shortening of at least 10 and 20%, respectively.

F Shortening of transcripts is evidenced by down-regulated poly(A) sites (PAS) in the 3′end RNA-seq data and accompanied by up-regulated upstream PAS for the majority of genes. The plot shows the fraction of genes with shortened (relative Δ90% distance ≥ 0.2), extended (absolute Δ90% distance < -50 bp), or unaffected transcripts (|absolute Δ90% distance| ≤ 25 bp) with down-, up-, and non-regulated PAS according to the 3′end RNA-seq data. For genes with shortened transcripts and down-regulated PAS in a 3′ UTR, the percentage of genes with upstream up-regulated PAS is indicated on the right. In case of multiple identified PAS, the order of preference was as indicated in the legend from top to bottom.

G DNA replication and repair genes are longer than other protein-coding genes. Box plots show the length for the indicated groups of genes (according to GO annotations). Median gene lengths for each GO category were compared against all other protein-coding genes using a one-sided Wilcoxon rank sum test (*P*-values provided in figure, n.s.: $P > 0.001$). See legend in Fig 6C for the boxplot description.

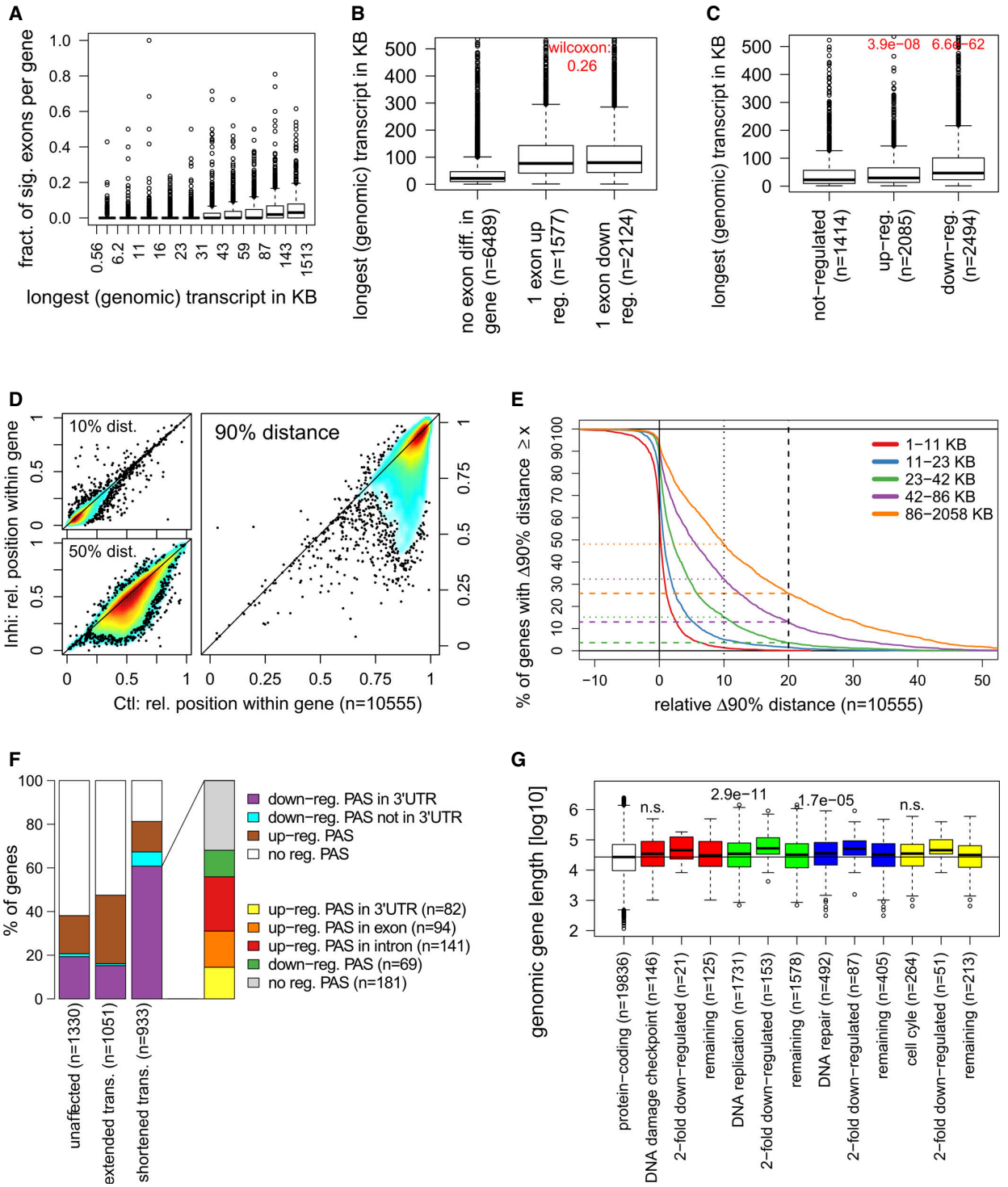


Figure 7.

to the TSS at which a certain percentage of read coverage (10, 50 or 90%) was observed for each gene in the nuclear RNA-seq data (denoted as the x% distance). When comparing control and inhibited samples, we observed little difference for the 10% distance, indicating that CDK12 inhibition does not substantially affect transcription close to the TSS (Fig 7D). In contrast, we observed a significant reduction for the 50 and 90% distances, consistent with transcripts getting shorter due to the processivity defect caused by CDK12 inhibition (Fig 7D). To find out how many genes are affected by the transcript shortening defect and to which degree transcripts were shortened, we evaluated the percentage of genes with a certain change in their 90% distance after CDK12 inhibition relative to their length (denoted relative Δ90% distance, Fig 7E and Dataset EV3). The 90% distance was used as proxy for transcripts ends as these were mostly not clearly defined after CDK12 inhibition as the RNA-seq signal tapered off over some range. Division of genes into quintiles based on their length showed that the longest genes (86–2,058 kb) are massively affected by transcripts shortening when compared to short ones (1–23 kb; Fig 7E). For instance, almost 50% of the longest genes are shortened by at least 10%, while < 5% of short genes are affected to this extent (Fig 7E). Notably, the longest genes lose a higher proportion of their transcript length: 26% of these genes are shortened in transcription by at least 20%, whereas such shortening occurs rather exceptionally (< 1%) in shorter genes (Fig 7E). Metagene analyses of ChIP-seq data demonstrated that genes with shortened transcripts (relative Δ90% distance ≥ 0.2) have reduced RNAPII occupancies at their 3′ends and show a strong shift of the P-Ser2 signal to gene bodies (Appendix Figs S13 and S14).

Next, we asked whether shortening of transcripts might also be influenced by sequence-specific properties, in particular the presence of canonical poly(A) signal sequences (AATAAA, ATTAAA). Since gene length and the abundance of poly(A) signal sequences are highly correlated (Spearman rank correlation ρ = 0.94, Appendix Fig S15A), we grouped genes according to the number of canonical poly (A) signals divided by gene length (denoted as poly(A) signal content) and then evaluated changes in the 10, 50, or 90% distance after CDK12 inhibition for each group (Appendix Fig S15B). Interestingly, we observed a correlation to the poly(A) signal content for the changes in the 90% distance, and to a lesser degree for changes in the 50% distance, with genes with a higher poly(A) signal content showing a stronger shortening of transcripts. This suggests that the presence of poly(A) signals may contribute to the shortening of transcripts and possibly explains why longer genes are more affected by the processivity defect as they contain a larger number of poly(A) signals. Since our 3′end RNA-seq data provide information on polyadenylated transcripts ends, we used these data to identify down-regulated poly(A) sites (PAS) as well as upstream PAS with increased usage after CDK12 inhibition (Fig 7F, see Materials and Methods). For 60% of genes with shortened transcripts, we found at least one down-regulated PAS in an annotated 3′ UTR in the 3′end RNA-seq data. Furthermore, 55% of these genes exhibited at least one up-regulated upstream PAS and 15% exhibited multiple up-regulated upstream PAS. Notably, in the majority of cases these upstream PAS were not found in annotated 3′ UTRs but in other exons or introns. Recently, it was reported that CDK12 suppresses intronic polyadenylation sites [63]. While our data show up-regulation of intronic PAS, considering the much larger number of potential intronic PAS compared to exonic/UTR PAS, no particular enrichment of intronic PAS was observed among upstream up-regulated PAS.

Considering the enrichment of DNA replication and repair genes as well as cell cycle genes among CDK12-dependent genes, we investigated whether genes in these groups tended to be longer than other protein-coding genes and thus more affected by the processivity defect. We found that these groups of genes tended to be longer than average protein-coding genes (Fig 7G), though the differences in median gene length were small and statistically significant only for DNA replication and DNA repair. Notably, however, down-regulated genes in each group tended to be even longer, whereas the remaining genes in each group tended to be closer to the median gene length of the other protein-coding genes.

In summary, our results show that CDK12 catalytic activity is essential for optimal RNAPII processivity at longer genes, including many DNA replication and DNA repair genes.

CDK12 inhibition decreases transcription elongation rates in bodies of genes with a RNAPII processivity defect

Since CDK12 is a regulator of transcription elongation [11,12,17], we wanted to determine whether genes with a CDK12-dependent processivity defect showed reduced elongation rates. To address this question, we measured elongation rates by RT–qPCR as the onset of a pre-mRNA expression "wave" at two different positions along the gene determined by primers at corresponding intron–exon junctions [64,65]. Initially, cells are treated with the pan-kinase inhibitor 5,6-dichlorobenzimidazole 1-β-D-ribofuranoside (DRB) to switch off the transcription cycle and synchronize RNAPII at gene promoters [64]. The inhibitor wash off releases RNAPII into gene bodies, and pre-mRNA is synthetized at a relatively uniform elongation rate of 3–5 kb per minute along individual genes [64,66]. RNA samples are taken every 3–8 min after the wash off, and the change in elongation rate is determined by monitoring the onset of pre-mRNA synthesis at specific locations in the gene defined by primer positions [64]. To assess the role of CDK12 kinase activity on elongation rates, we selected three CDK12-dependent (*TOPBP1, MCM10, UBE3C*) and two CDK12-independent (*ARID1A, SETD3*) genes and compared their pre-mRNA synthesis in AS CDK12 HCT116 cells either treated or not with 3-MB-PP1 after the DRB wash off (see Fig 8A for the experimental setup). For each gene, we designed two primer sets within its gene body, one at its 5′end and another close to its center. For the CDK12-dependent genes, the second set of primers always preceded the region where the loss/decrease of RNAPII processivity became apparent in the RNAPII ChIP-seq and RNA-seq signals (Fig 5E, and Appendix Fig S7B and C). DRB wash off in control samples resulted in an onset of pre-mRNA synthesis at expected time points (based on the location of primers) and was consistent with an expected elongation rate between 3 and 5 kb per minute along the gene body (Fig 8B). In CDK12-inhibited samples, we found a delay in the onset of pre-mRNA synthesis in all the locations tested. Surprisingly, synthesis of pre-mRNA of all investigated genes was already delayed at 5′ends by a similar time window of approximately 3–6 min (Fig 8B, compare time of upswing of blue and brown curves). This indicates that CDK12 kinase activity may play a role in an optimal release of promoter-paused RNAPII on those genes. Importantly, in the middle of gene bodies of the CDK12-independent genes the delay in pre-mRNA synthesis was comparable to the one observed at their

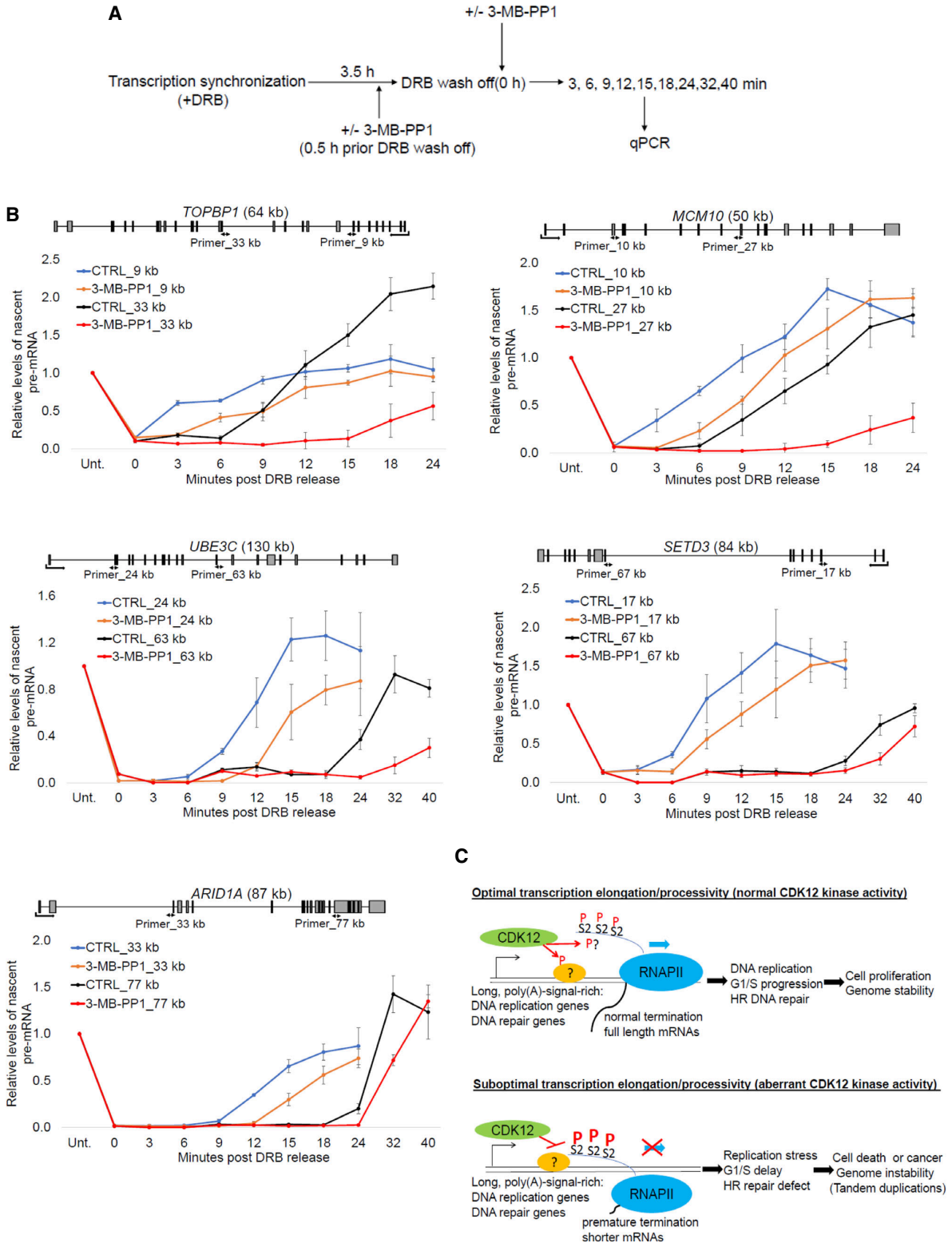


Figure 8.

Figure 8. CDK12 inhibition decreases transcription elongation rates in bodies of genes with RNAPII processivity defect.

A Experimental outline for measurement of transcription elongation rates. AS CDK12 HCT116 cells were treated with DRB for 3.5 h to synchronize RNAPII at gene promoters. The cells were either pretreated (+) or not (–) with 3-MB-PP1 0.5 h prior DRB wash off. After DRB wash off (0 h), fresh medium either supplemented (+) or not (–) with 3-MB-PP1 was added and samples were taken at indicated time points for analyses of pre-mRNA expression by RT–qPCR.

B Transcription elongation rate decreases in bodies of CDK12-dependent but not CDK12-independent genes after CDK12 inhibition. Graphs show relative levels of pre-mRNAs of described genes in AS CDK12 HCT116 cells either treated with 3-MB-PP1 or not (CTRL) for indicated times after DRB wash off. Pre-mRNA levels were normalized to the samples not treated with DRB (Unt) for which the value was set as 1. n = 3 independent experiments, error bars correspond to SEM. Positions of primers (designed to span exon–intron junctions) and their distance from the transcription start site in kb are indicated in the gene structures shown above the graphs.

C Proposed model. Schema shows groups of genes whose RNAPII processivity is particularly sensitive to CDK12 catalytic activity and cellular functions that are especially dependent on optimal expression of these genes. The situation in cells with normal and aberrant CDK12 kinase activity is depicted. CDK12 (green oval) phosphorylates (P) unknown substrate(s) (orange oval), possibly including the CTD (blue line), which results in optimal elongation and processivity (blue arrow) of RNAPII (blue oval) for CDK12-sensitive genes. Full length, functional mRNAs are synthesized (upper panel). Inhibition of CDK12 leads to hyperphosphorylation (capital P) of Ser2 (S2) in bodies of CDK12-sensitive genes, which is associated with slower elongation and premature termination. Shorter, aberrant mRNAs are made (lower panel). mRNAs are depicted as black lines.

5′ends (3–6 min), indicating that elongation rates do not change considerably on their genes bodies. This was in a contrast to the CDK12-dependent genes where the delay in pre-mRNA synthesis in the middle of the genes was much longer (at least 9 min; Fig 8B, compare time of upswing of black and red curves). This indicates that RNAPII elongation slows down in bodies of these genes when the CDK12 kinase is inhibited which likely contributes to or accompanies the observed RNAPII processivity defect.

Although these experiments were performed only on a limited number of genes, they suggest that the CDK12-dependent RNAPII processivity defect is accompanied by slower elongation rates at gene bodies of the affected genes.

Discussion

Using rapid and specific inhibition of CDK12 kinase activity in AS CDK12 cells, we uncovered a crucial role for CDK12 catalytic activity in G1/S progression. CDK12 activity is required for optimal expression of core DNA replication genes and timely formation of the pre-replication complex on chromatin. Our genome-wide studies of total and modified RNAPII suggest that CDK12 kinase does not globally control P-Ser2 levels on transcription units; however, it is crucial for RNAPII processivity on a subset of long and poly(A)-signal-rich genes, particularly those involved in DNA replication and DNA damage response. We further demonstrate that CDK12-dependent RNAPII processivity is a rate-limiting factor for optimal G1/S progression and cellular proliferation.

The general requirement of CDK12 kinase activity for optimal G1/S progression in human cells is corroborated by our finding that CDK12 expression peaks in early G1 phase (Fig 2) resembling regulation of classical cell cycle-related cyclins [39]. This could not be accounted for by activation of the DNA damage checkpoint, as its signaling occurs later than 24 h post-inhibition, after the cell cycle defect. In parallel, CDK12 kinase activity directs transcription of crucial HR repair genes including *BRCA1, BRCA2, ATM,* and Fanconi anemia genes (Fig 8C) that are also essential for dealing with replication stress by protecting and/or restarting stalled replication forks [67]. As deregulation of DNA replication and cell cycle progression leads to replication stress and genome instability [39,68,69], these findings combined with a well-established role of CDK12 in the HR DNA repair pathway have important clinical implications, as discussed below.

Recent findings show that many cancers with disrupted CDK12 catalytic activity have a unique, CDK12-inactivation-specific genome instability phenotype: tandem duplications [29–33]. There are several possible scenarios for their genesis; nevertheless, we favor the concept that they arise due to disrupted expression of both core DNA replication and HR genes upon inhibition of CDK12. This leads to an onset of replication stress that as a consequence of inefficient HR-mediated fork restart results in use of alternative repair mechanism (Fig 8C). These defects thus correspond to the onset of HR-independent genome instability resulting in the distinct tandem duplication genome rearrangements pattern observed in tumors with inactivated CDK12. They likely have catastrophic consequences for cell survival, however in some cells are occasionally compensated by a pro-growth event leading to tumorigenesis with distinct tandem duplications (Fig 8C). The outcomes of early stages of CDK12 inactivation were mimicked in AS CDK12 HCT116 cells documenting a progressive accumulation of various chromosomal defects over several rounds of replication accompanied by a gradual decrease of cellular proliferation. Notably, the recently discovered role of CDK12 in translation of many mRNAs that encode subunits of mitotic and centromere complexes contributes to these defects and adds yet another layer of complexity into the essential function of CDK12 in the maintenance of genome stability [70].

During the course of our research, two studies suggested a connection between CCNK/CDK12 and S phase cell cycle progression: CDK12 deficiency was found to be synthetically lethal in combination with inhibition of S phase checkpoint kinase CHK1 [71], further supporting our findings as activation of the checkpoint will give the cell time to repair DNA damage caused by replication stress. In another study, knockdown of CCNK was shown to lead to G1/S cell cycle arrest [72]. The proposed mechanism suggested interference with pre-replication complex assembly caused by CDK12-mediated CCNE1 phosphorylation (directly or indirectly) [72]. Our results demonstrate that CDK12 also functions upstream of the pre-replication complex assembly, as CDK12 inhibition (and also CCNK depletion, see Fig EV3F and G) in the same cell line (HCT116) strongly down-regulate mRNA and protein levels of pre-replication complex subunits, including CDC6, CDT1, TOPBP1, and MTBP. It will be important to determine whether CDK12 can directly phosphorylate CCNE1 and regulate CCNE1/CDK2 activity in early stages of replication as suggested [72]. In particular, alterations in CCNE1 also lead to the onset of a distinct tandem duplication phenotype [32].

Mechanistically, CDK12 inhibition did not affect global transcription and P-Ser2 levels, but led to a loss of RNAPII processivity accompanied by transcript shortening of a subset of genes, consistent with defective transcriptional elongation. Individual CDK12-dependent genes showed a shift of P-Ser2 peaks toward gene 5′ends approximately to the positions where RNAPII occupancy and transcription was lost, i.e., to new 3′ends of shortened transcripts. Notably, our findings resemble inhibition of CDK12 by very low (50 nM) concentrations of THZ531, when only a subset of genes, including DNA repair genes, was down-regulated without an appreciable decrease of P-Ser2 levels [17]. In contrast, we did not find wider transcriptional defects and parallel loss of Ser2-phosphorylated RNAPII as observed with higher (≥ 200 nM) THZ531 concentrations [17]. This difference might be potentially explained by a residual kinase activity in the presence of competitive 3-MB-PP1 in contrast to a complete kinase shut-off with higher concentrations of covalent THZ531 or alternatively by off-target effects of higher concentrations of THZ531.

Overall, our data indicate a role of human CDK12 that is different from that of CDK12 homologs in *Saccharomyces cerevisiae* and *Drosophila*, where the kinase is responsible for global P-Ser2 phosphorylation and regulation of elongation [12,73]. One possible explanation might be the presence of CDK13 and BRD4, redundant P-Ser2 kinases, in humans [12,20,74]. In *Schizosaccharomyces pombe*, short (5 min) inhibition of AS *Lsk1,* a non-essential CDK12 homolog, decreased Ser2 phosphorylation, but had only a subtle effect on RNAPII distribution and transcription [75]. Although we cannot completely rule out that very short (in minutes) CDK12 inhibition globally affects transcription in human cells, this seems unlikely, since bulk P-Ser2 and P-Ser5 levels in cells are either not affected or only subtly (Figs 1D and EV1D) [48]. Notably, bulk phosphorylation of Ser7, the modification implied in expression of small nuclear RNAs (snRNAs) [76], was decreased after CDK12 inhibition (Figs 1D and EV1D). In any case, our experiments using 4.5-h inhibition identified the subset of genes whose transcription is crucially dependent on CDK12 catalytic activity. Notably, we did not find any evidence that inhibition of CDK12 affects alternative last exon splicing, as observed in breast cancer cell lines upon CDK12 depletion [28]. Thus it seems likely that this function of CDK12 is independent of its kinase activity.

Inspection of individual genes sensitive to CDK12 inhibition revealed a relative accumulation of RNAPII hyperphosphorylated on Ser2 on the gene body rather than at gene 3′ends, predominantly at a longer distance from the TSS together with a sudden loss of RNAPII occupancy and transcription from a gene at approximately the same position. Although we cannot determine the order and consequence of events, we speculate that disrupted or slow elongation results in a compensatory increase of phosphorylation on Ser2 by an unknown kinase (in bulk, the time-dependent accumulation of P-Ser2 and also, to some extent P-Ser5, is visible in Figs 1D and EV1D). Alternatively, inactivation of a P-Ser2 phosphatase or its disabled recruitment, perhaps via CDK12-mediated changes in Ser7 phosphorylation, could be involved. In either scenario, the aberrant accumulation of P-Ser2 in gene bodies of long genes might represent a signal for triggering premature termination or polyadenylation (Fig 8C). We found that long genes, genes with higher numbers of canonical poly(A) signals, and subsets of DNA replication and DNA damage response genes are most reliant on CDK12 catalytic activity. Although CDK12-dependent genes are on average longer than other human genes, we believe that there must be yet another mechanistic/signaling basis for their dependence on the kinase. Given the catastrophic phenotypic effects of aberrant CDK12-mediated processivity, identification of the corresponding CDK12 substrate(s) will be of high importance.

During revision of this study, it was revealed that inducible depletion of full length CDK12 leads to enhanced usage of intronic PAS resulting in down-regulation of a subset of genes, particularly HR genes [63]. This was explained by a shortening of transcripts due to a higher occurrence of intronic PAS in these genes and their higher sensitivity to CDK12 loss. We also found that CDK12 inhibition results in transcript shortening for a subset of genes with a higher frequency of poly(A) signals. Nevertheless, we did not conclusively identify enriched intronic PAS usage compared to exonic/UTR PAS in our datasets when CDK12 was inhibited (Fig 7F). Perhaps mere inhibition of CDK12 by 3-MB-PP1 is not sufficient to trigger preferential use of intronic PAS although slower elongation and premature termination still occur on CDK12-sensitive genes. Alternatively, some of the numerous experimental differences between the studies can account for the difference.

We conclude that CDK12-dependent RNAPII processivity is a rate-limiting factor for optimal transcription of DNA replication genes and G1/S progression, which provides a novel link between regulation of transcription, cell cycle progression, and genome stability. Overall, our study has important implications for understanding the CDK12 cellular function, origins of CDK12-specific genome instability phenotype, and in longer term for the development of CDK12-specific cancer therapy.

Materials and Methods

Cell synchronization and cell cycle analysis

WT or AS CDK12 HCT116 cells were synchronized by serum starvation (for G0/G1 block) and AS CDK12 HeLa cells by thymidine–nocodazole (for mitotic block). For serum starvation, cells were plated at 50–60% confluency onto 60-mm dishes containing starvation medium (0.1% FBS containing DMEM) for 72 h and then released into medium containing 15% FBS. For mitotic block, the cells were plated at 60–70% confluency onto 60 mm dishes, and after incubation with 2 mM thymidine (Sigma, T1895) for 24 h, the cells were washed twice with PBS and released into fresh media for 3 h. This was followed by 100 ng/ml nocodazole (Sigma, M1404) block for 10 h. Then, the cells were washed twice with PBS and then released into fresh media containing 10% FBS. Synchronously progressing cells were collected at appropriate time points depending on the type of experiment. During the time of release (0 h), cells were treated with either DMSO (CTRL) or 5 μM ATP analog 3-MB-PP1 inhibitor (Merck, 529582) for the indicated times. Cell cycle profile was measured by flow cytometry based on the DNA content of cells using propidium iodide (PI) (Sigma, P4170) staining. For the PI staining, trypsinized cells were washed twice with PBS, fixed with ice-cold 70% (v/v) ethanol, and incubated at −20°C for 2 h. After washing twice with ice-cold PBS, cells were resuspended in Vindal buffer (10 mM Tris–Cl, pH = 8, 1 mM NaCl, and 0.1% Triton X-100) containing freshly added PI (50 μg/ml) and RNase A (200 μg/ml;

Qiagen, 19101) and incubated for 20 min at room temperature before measurement by *BD FACSVerse* (BD Bioscience). Cell cycle distribution was analyzed by *FLOWING version 2.1* software.

Rescue or washout assay

Serum-starved AS CDK12 HCT116 cells were released by serum addition (with DMEM containing 15% FBS; 0 h) and treated with 5 μM 3-MB-PP1 for the indicated time points. Medium containing inhibitor was subsequently removed, cells were washed carefully three times with warm PBS, and fresh medium (DMEM containing 15% FBS) was added. Cells were collected at appropriate time point for flow cytometry (0 and 15 h), immunoblotting (12 h), nuclear fractionation (6 and 9 h), or RT–qPCR (7 h).

Nuclear fractionation

AS CDK12 HCT116 cells were seeded onto 150-mm dishes and synchronized by serum starvation as described. After release into 15% fetal bovine serum (FBS) containing medium with either DMSO (CTRL) or 5 μM 3-MB-PP1 dissolved in DMSO, the cells were grown for various time points and then harvested. Cell pellets were washed twice in PBS, and small aliquots were taken away for flow cytometry analyses. Remaining cell pellets were quickly frozen in dry ice and stored at −80°C. After collecting all the time points, the samples were further processed together.

Briefly, each cell pellet was lysed in 500 μl of cytoplasmic lysis buffer on ice for 5 min [10 mM Tris–Cl pH = 8.0, 0.32 M sucrose, 3 mM $CaCl_2$, 2 mM $MgCl_2$, 0.1 mM EDTA, 1 mM DTT, 0.5% Triton X-100, and Protease inhibitor cocktail (Sigma, P8340)] and spin at 500 *g*/5 min/4°C. The supernatant containing cytoplasmic fraction was discarded, and the pellets were washed once in 500 μl of the cytoplasmic lysis buffer and once in 500 μl of the same buffer without detergent to remove any residual cytoplasmic proteins. Remaining nuclear extracts were resuspended in 80 μl of EDTA-EGTA buffer [3 mM EDTA, 0.2 mM EGTA, 1 mM DTT, and Protease inhibitor cocktail (Sigma, P8340)] and left on ice for 30 min, then spin at 10,000 *g*/5 min/4°C, and supernatant was discarded. Remaining pellets containing chromatin-bound proteins (insoluble nuclear fraction) were washed once in 300 μl EDTA-EGTA buffer and after spin at 1,700 *g*/10 min/4°C lysed in 40 μl of RIPA buffer [50 mM Tris–Cl pH = 8, 5 mM EDTA, 150 mM NaCl, 1% NP-40, 0.5% sodium deoxycholate, 0.1% SDS, 1 mM $MgCl_2$, Protease inhibitor cocktail (Sigma, P8340), and benzonase nuclease (Sigma, E1014)] for 30 min at 37°C. After addition of SDS sample buffer, the samples were sonicated 10 × 3 s (amplitude 0.20) (QSonica Q55), spin at 13,000 *g* for 1 min, and boiled at 95°C for 3 min. Protein levels in insoluble nuclear fractions were analyzed by Western blotting.

Cell lines and chemicals

HCT116 human colon carcinoma cells (ATCC) and HeLa human cervical carcinoma cells (gift from Dr. A.L. Greenleaf, Duke University Medical Center, USA [48]) were maintained in Dulbecco's modified Eagle's medium (DMEM) containing high glucose supplemented with L-glutamine, sodium pyruvate (Sigma, D6429), and 5% FBS (Sigma, F7524) at 37°C and 5% CO_2. All the chemicals were purchased from Sigma, unless specified otherwise.

Generation of AS CDK12 HCT116 cells by genome editing

To create AS CDK12 HCT116 cell line, both alleles of *CDK12* were targeted using CRISPR/Cas9 system as previously described [48,77]. Guide RNA (20-nt) targeting exon 6 of *CDK12* was designed with appropriate PAM motif (5′-NGG) as close to the F813 codon as possible. Sequences of single-guide RNA (sgRNA) used were the following: CDK12-sgRNA-1: ATA CTC AAA TAC AAG GTA AAA GG; Cdk12-sgRNA-2: GGT CCA TAT ACT CAA ATA CAA GG. The efficiency of gRNA/Cas9 targeting and activity was validated by sequencing with the following primers: CKD12-Seq 1-fwd: TAG GAC TTG AGG CAT TGT TAT TTC, CDK12-Seq 1-rev: TTA GAA CAC TTA ATA TCC CGA TGA. HCT116 cells expressing Cas9 and *CDK12* targeting sgRNA were transfected with a 166-nt-long homologous repair template that introduced desired genome changes. The homologous repair template contains: TTT to GGG mutation which results in F813G, adjacent silent change (A to T) to generate a novel *BSII* restriction site to facilitate downstream validation and a silent mutation GTA to GTT to prevent alternative splicing. Following selection, individual colonies were isolated by low density plating and expanded, and PCR genotyped using specific forward PCR primers for either WT (CDK12-PCR 1-WT-fwd: GGT GCC TTT TAC CTT GTA TTT GA) or AS (CDK12-PCR 1-AS-fwd: GTG CCT TTT ACC TTG TTG GGG AG) sequences and with the reverse primer (CDK12-PCR 1-rev: GGA GCA GGT ATG TTT CTC CCA; Fig EV1B). Positive clones were further validated by PCR of genomic DNA using the following primers: CDK12-PCR 2-fwd (GCT CCG TTG TTT ATT ATT AGG AAG G) and CDK12-PCR 2-rev (TCA CTA AAT AGT GTG TGA ATA CTG C) followed by digestion using *BsII* (Thermo Fisher Scientific, FD1204) (Fig 1B). Digested products were separated by agarose gel electrophoresis and the pattern of digestion confirmed homozygous AS CDK12 clones. Initial PCR screening was followed by Sanger sequencing with the following primers to confirm the presence of the desired mutation (Fig 1C): CDK12-PCR 3-fwd: CCC CCA TGA AGA GGT GAG TAG and CDK12-PCR 3-rev: GGA GCA GGT ATG TTT CTC CCA, and CDK12-Seq 2-fwd: GCT CCG TTG TTT ATT ATT AGG AAG G and CDK12-Seq 2-rev: TCA CTA AAT AGT GTG TGA ATA CTG C. Immunoprecipitation of CDK12 followed by Western blotting with cyclin K (CCNK) from both WT and AS CDK12 HCT116 cells was performed to check the presence of intact CDK12/CCNK complex.

BrdU incorporation assay

To differentiate between replicating and non-replicating cells based on the staining of newly synthesized DNA, BrdU (5-Bromo-2′-deoxyuridine) incorporation assay was performed as described in [78]. Briefly, BrdU (Sigma, B9205) was added to the cell culture medium at a final concentration of 10 μM and incubated for 30 min. After BrdU incorporation, cells were harvested and washed twice with 1% BSA/PBS before fixing in 70% (v/v) ethanol at −20°C for 2 h. Ethanol fixed cells were denatured with 2 N HCl containing 0.5% Triton X-100 for 30 min to yield single-stranded DNA molecules. Cells were resuspended in 0.1 M $Na_2B_4O_7.10\ H_2O$ (pH = 8.5) to neutralize acid before resuspending in 1% BSA/PBS/0.5% Tween-20. Cells were then incubated with 0.5 μg anti-BrdU FITC (clone B44, BD Bioscience, 347583) for 45 min, washed twice with 1% BSA/PBS, and stained with propidium iodide (5 μg/ml) before

measurement by *BD FACSVerse* (BD Bioscience). Data were analyzed by *FlowJo version 10* software.

Immunoblotting

For the isolation of total cellular proteins, AS CDK12 HCT116 cells (from either 100 mm or 150 mm dishes) were lysed in protein lysis buffer [20 mM HEPES-KOH, pH = 7.9, 15% glycerol, 150 mM KCl, 1 mM EDTA, 0.2% NP-40 (Sigma, 18896), 1 mM DTT, 0.5% v/v Protease inhibitor cocktail (Sigma, P8340)], sonicated, and centrifuged (10,000 *g*, 10 min, 4°C). Cellular protein concentrations were quantified using the bicinchoninic acid (BCA) protein assay. Equal amounts of proteins were loaded onto appropriate percentage of either Tris-glycine or Tris-acetate gels, and proteins were resolved by SDS–PAGE using appropriate running buffer under denaturing conditions (120 V for 90 min). For immunoblotting, proteins were electrophoretically transferred (100 V for 1 h) to 0.45-μm nitro cellulose membranes (Sigma, GE10600008). After blocking either with 5% nonfat dry milk or bovine serum albumin (BSA) in TBS-T buffer for 90 min at room temperature, the membranes were probed using antibodies raised against the indicated proteins overnight at 4°C (see the Table 1 for the complete list of antibodies used in this study). Either FUS or α-tubulin was used as loading controls. Membranes were washed and subsequently incubated with appropriate HRP (horseradish peroxidase)-conjugated secondary antibody (GE Healthcare, NA931V, NA934V or Santa Cruz, sc-2032) for 1 h at room temperature. Immunoreactive bands were detected on either Amersham Hyperfilm ECL or UltraCruz Autoradiography Film (Santa Cruz, sc-201697) using enhanced chemiluminescence reagent (Western Blotting Luminol Reagent, Santa Cruz, sc-2048).

Immunoprecipitation

WT or AS CDK12 HCT116 cells (150 mm dish per IP) were harvested in ice-cold PBS, lysed in protein lysis buffer [20 mM HEPES-KOH, pH 7.9, 15% glycerol, 150 mM KCl, 1 mM EDTA, 0.2% NP-40 (Sigma, 18896), 1 mM DTT, 0.5% v/v protease inhibitor cocktail (Sigma, P8340)], sonicated, and cleared by centrifugation (10,000 *g*, 10 min, 4°C). Cellular protein concentrations were quantified using the bicinchoninic acid (BCA) protein assay. For CDK12 IP, lysate was incubated with 2 μg of anti-CDK12 antibody (Santa cruz, sc-81834) for 2 h at 4°C, followed by incubation with pre-washed protein G sepharose beads (GE Healthcare, 17-0618-01; 20 μl per IP) for another 2 h at 4°C. Immunoprecipitates were washed three times with 1 ml protein lysis buffer, eluted from the beads with 40 μl 3× Laemmli sample buffer, and then boiled for 4 min at 95°C. SDS–PAGE resolved immunoprecipitated proteins, followed by Western blotting, and probed for indicated proteins.

SPT6 immunoprecipitation

AS CDK12 HCT116 cells (150 mm dish per IP) were treated for 4 h with either DMSO (CTRL) or 5 μM 3-MB-PP1 dissolved in DMSO. Cells were harvested in ice-cold PBS, and pellets were equalized in size, lysed in protein lysis buffer [20 mM HEPES-KOH, pH 7.4, 100 mM KCl, 0.5% Triton X-100 (Sigma, 18896), 1 mM DTT, protease inhibitor cocktail (Sigma, P8340)], sonicated, and centrifuged (10,000 *g*, 10 min, 4°C). 20 μl of protein G Dynabeads (Thermo Fisher Scientific, 10009D) per IP was washed three times in protein lysis buffer and incubated 4 h at 4°C with 1 μg of anti-SPT6 antibody (Novus, NB100-2582) per IP or without antibody as a control. Beads were washed three times with 1 ml of protein lysis buffer and incubated with lysates overnight at 4°C. Immunoprecipitates were washed three times with 1 ml of protein lysis buffer; 30 μl of 3× Laemmli buffer was added and then boiled for 3 min at 95°C. The immunoprecipitates were resolved by SDS–PAGE, and Western blots were probed with SPT6 and RNAPII antibodies.

Chromosomal aberration assay by metaphase spreads

Chromosomal aberration assay was performed as described previously [79] with AS CDK12 HCT116 cells treated with and without 5 μM 3-MB-PP1 for 24 and 48 h, and with 4 mM hydroxyurea (Sigma, H8627) for 5 h as a positive control. Briefly, at the end of the treatment the cells (from 25-cm^2 flasks) were incubated with 0.1 μg/ml KaryoMax colcemid (Thermo fisher Scientific, 15212012) for 90 min to arrest the cells in metaphase and allow chromosome spreading. Cells were swollen by treatment with hypertonic KCL (0.075 M) for 12 min at 37°C and fixed with methanol: glacial acetic acid (3:1). Cells were carefully dropped onto a microscopic slide, stained with 5% Giemsa, and air-dried. Slides were mounted with Richard-Allan Scientific Cytoseal 60 (Thermo Fisher Scientific, 8310-16) and analyzed with an Olympus BX60 microscope at 1,000× magnification.

siRNA-mediated knockdown

AS CDK12 HCT116 cells were plated at 30% confluency 7–11 h before transfection. siRNA was transfected at a final concentration of 10 nM using Lipofectamine RNAiMax (Thermo Fisher Scientific, 13778-150) according to the manufacturer's instruction. Briefly, to transfect one well in 6-well plate we mixed together 2.5 μl of siRNA (10 μM stock solution) diluted in 250 μl of Opti-MEM (Thermo Fisher Scientific, 31985-070) with 5 μl of Lipofectamine diluted in 250 μl of Opti-MEM. After 15 min, the mixture was added dropwise into the cultured cells containing 2.5 ml of media. If larger plates were used for transfections, the amount of reagents was scaled up proportionally. Control samples were transfected with non-targeting control siRNA-A (Santa Cruz, sc-37007). The levels of proteins after depletion were analyzed by Western blotting with appropriate antibodies. The list of siRNAs used in this study is specified in the Table 2.

Reverse transcription qPCR

Total RNA was isolated by TRIzol reagent (Thermo Fisher Scientific, 15596026) according to the manufacturer's protocol. 1 μg of total RNA was treated with 1 μl of DNase (Sigma, AMPD1) and reverse transcribed using 200 U SuperScript II RT (Thermo Fisher Scientific, 18064-014) with random hexamers (IDT, 51-01-18-01). Quantitative gene expression analysis was performed on AriaMx Real-Time PCR System (Agilent) using SYBR Green. In general, each reaction (final volume 11 μl) contained 5.5 μl SYBR Green JumpStart Taq ReadyMix (Sigma, S4438), 200 nM of each primer (primer sequences used in this study are specified in the Table 3), 0.28 μl H_20, and 5 μl diluted cDNA template, with the following PCR cycling conditions: 95°C for 2 min followed by 45 cycles of

Table 1. Antibodies used for ChIP, IP, and Western blotting.

Target protein	Clone	Cat. no.	ChIP	IP	WB	Source/Reference
CCNK	G-11	sc-376371	–	3 μg	1:500	Santa Cruz
CDC6	180.2	sc-9964	–	–	1:200	Santa Cruz
MTBP	B-5	sc-137201	–	–	1:600	Santa Cruz
CDT1	F-6	sc-365305	–	–	1:300	Santa Cruz
FUS	4H11	sc-47711	–	–	1:10,000	Santa Cruz
Histone 2A (H2A)		ab18255	–	–	1:10,000	Abcam
ORC6	3A4	sc-32735	–	–	1:3,000	Santa Cruz
E2F1		A300-766A	5 μg	–	–	Bethyl
E2F3	PG30	sc-56665	5 μg	–	–	Santa Cruz
CDK12	U1-4th immune	–	–	–	1:3,000	In-house made
CDK12	R-12	sc-81834	–	2 μg	1:500	Santa Cruz
Cyclin A2		4656	–	–	1:1,000	Cell Signaling
Cyclin E2		4132	–	–	1:1,000	Cell Signaling
Cyclin E1		4129	–	–	1:1,000	Cell Signaling
RNAPII	N-20	sc-899x	2 μg	–	1:1,000	Santa Cruz
Phospho-RNAPII (Ser2)	3E10	61083	3 μg	–	1:6,000	Active Motif
Phospho-RNAPII (Ser5)	3E8	61085	3 μg	–	1:8,000	Active Motif
α-Tubulin	B-7	sc-5286	–	–	1:200	Santa Cruz
ATM		2873	–	–	1:300	Cell Signaling
Phospho-ATM (Ser1981)	EP1890Y	ab81292	–	–	1:1,000	Abcam
p53	DO-1	–	–	–	1:10	In-house made
Phospho-p53 (Ser15)		9284	–	–	1:800	Cell Signaling
TOPBP1	B-7	sc-271043	–	–	1:250	Santa Cruz
CDC7	SPM171	sc-56275	–	–	1:600	Santa Cruz
ORC2	3G6	sc-32734	–	–	1:1,500	Santa Cruz
ORC3	1D6	sc-23888	–	–	1:1,500	Santa Cruz
GINS4 (SLD5)	D-7	sc-398784	–	–	1:400	Santa Cruz
MCM3	E-8	sc-390480	–	–	1:200	Santa Cruz
CDK13	N-term.	–	–	–	1:3,000	In-house made
SPT6		NB100-2582	3.5 μg	1 μg	1:4,000	Novus Biologicals
RNAPII		NBP2-32080			1:2,000	Novus Biologicals
Phospho-RNAPII (Ser7)		4E12	–	–	1:1,000	Chromotek
RNAPII (Rpb7)	C-20	sc-398213	–	–	1:100	Santa Cruz
Sheep anti-mouse IgG-HRP		NA931V	–	–	1:3,000	GE Healthcare Life Sciences
Donkey anti-rabbit IgG-HRP		NA934V	–	–	1:3,000	GE Healthcare Life Sciences
Goat anti-rat IgG-HRP		sc-2032	–	–	1:3,000	Santa Cruz

Table 2. siRNAs used in this study.

Gene	Cat. no.	Source
siCTRL A	sc-37007	Santa Cruz
siCCNK	sc-37600	Santa Cruz

denaturation at 95°C for 15 s, annealing at 55°C for 30 s, and extension at 72°C for 30 s. All reactions were performed in triplicates for each biological replicate, and melting curve analyses were routinely performed to monitor the specificity of the PCR product. The relative gene expression was determined using comparative C_T method ($2^{-\Delta\Delta C_T}$ method) with either *HPRT1* or *B2M* as normalizer.

Analysis of mRNA stability

To assess relative stability of select DNA damage and replication transcripts, AS CDK12 HCT116 cells were treated with 1 μg/ml actinomycin D (Sigma, A9415) to block transcription in the presence or absence of 5 μM 3-MB-PP1. Cells were harvested at various time points (0 to 5 h) after actinomycin D treatment

Table 3. Primers used in this study.

Name	Sequence (5′–3′)	Method used	Reference
CCNK (ex8-ex10) F	AACAGCCCAAGAAACCCTC	RT–qPCR	This study
CCNK (ex8-ex10) R	CAACGGTGGATGAGTGGTC	RT–qPCR	This study
MTBP (ex10-ex11) F	GGATTGACAAACAGTACCAAACAG	RT–qPCR	This study
MTBP (ex10-ex11) R	GTTGGGAGGTGGAATCAGTATG	RT–qPCR	This study
CCNE2 (ex3-ex4-ex5) F	AAGAGGAAAACTACCCAGGATG	RT–qPCR	This study
CCNE2 (ex3-ex4-ex5) R	ATAATGCAAGGACTGATCCCC	RT–qPCR	This study
CDC6 (+1,860) F	AGAACATGCTCTGAAAGATAAAGC	RT–qPCR	This study
CDC6 (+1,922) F	GGTGTAAGAGAAGAATTTAAGGCAA	RT–qPCR	This study
TOPBP1 (ex24-ex25) F	GCTTCATCGCTCCTACCTTG	RT–qPCR	This study
TOPBP1 (ex24-ex25) R	AGTGCTAGTCTTCGTTGCTG	RT–qPCR	This study
MCM10 (ex18-ex19-ex20) F	ACTCCCGAACAAGCACTG	RT–qPCR	This study
MCM10 (ex18-ex19-ex20) R	GTCTTTTCCTTTAGCATTCCGTC	RT–qPCR	This study
ORC2 (ex10-ex11-ex12) F	GAGAGCTAAACTGGATCAGCA	RT–qPCR	This study
ORC2 (ex10-ex11-ex12) R	GCACAATGTTGAACCCAAGG	RT–qPCR	This study
CDT1 (ex9-ex10) F	AGCGTCTTTGTGTCCGAAC	RT–qPCR	This study
CDT1 (ex9-ex10) R	AGGTGCTTCTCCATTTCCC	RT–qPCR	This study
ORC3 (ex4-ex5) F	GGGCGGTCAAATAAAACTCAG	RT–qPCR	This study
ORC3 (ex4-ex5) F	GCCTCTGTTAGACTTCCGAATG	RT–qPCR	This study
C-MYC (+1,855) F	CAC AAA CTT GAA CAG CTA CGG	RT–qPCR	This study
C-MYC (+1,941) R	GGT GAT TGC TCA GGA CAT TTC	RT–qPCR	This study
BRCA1 (+5,718) F	AGATGTGTGAGGCACCTGT	RT–qPCR	This study
BRCA1 (+5,777) R	GTCCAGCTCCTGGCACT	RT–qPCR	This study
BRCA2 (ex18-ex19) F	TTCATGGAGCAGAACTGGTG	RT–qPCR	This study
BRCA2 (ex18-ex19) R	AGGAAAAGGTCTAGGGTCAGG	RT–qPCR	This study
FANCI (ex7-ex8) F	TGTAATCCAACTCACCTCCATG	RT–qPCR	This study
FANCI (ex7-ex8) R	GAGAACCAGAAGCTGATAGACC	RT–qPCR	This study
ATR (ex34-ex35) F	CGCTGAACTGTACGTGGAAA	RT–qPCR	This study
ATR (ex34-ex35) R	CAATAAGTGCCTGGTGAACATC	RT–qPCR	This study
Exo1 (+799) F	CCTCGTGGCTCCCTATGAAG	RT–qPCR	This study
Exo1 (+872) R	AGGAGATCCGAGTCCTCTGTAA	RT–qPCR	This study
CDK6 (ex2/ex3) F	TGGAGACCTTCGAGCACC	RT–qPCR	This study
CDK6 (ex2/ex3) R	CACTCCAGGCTCTGGAACTT	RT–qPCR	This study
CCND3 (ex2/ex3) F	TACACCGACCACGCTGTCT	RT–qPCR	This study
CCND3 (ex2/ex3) R	GAAGGCCAGGAAATCATGTG	RT–qPCR	This study
CDKN1B (ex1/ex2) F	CGGCTAACTCTGAGGACAC	RT–qPCR	This study
CDKN1B (ex1/ex2) R	TGTTCTGTTGGCTCTTTTGT	RT–qPCR	This study
CDKN2A (ex2/ex3) F	GAAGGTCCCTCAGACATCCCC	RT–qPCR	This study
CDKN2A (ex2/ex3) R	CCCTGTAGGACCTTCGGTGAC	RT–qPCR	This study
E2F1 (ex5/ex6) F	CAGAGCAGATGGTTATGGTG	RT–qPCR	This study
E2F1 (ex5/ex6) R	GGCACAGGAAAACATCGATC	RT–qPCR	This study
HPRT1 (ex5/ex6) F	ACACTGGCAAAACAATGCAG	RT–qPCR	This study
HPRT1 (ex5/ex6) R	ACTTCGTGGGGTCCTTTTC	RT–qPCR	This study
B2M (ex1/ex2) F	GCATTCCTGAAGCTGACAG	RT–qPCR	This study
B2M (ex1/ex2) R	GCTGGATGACGTGAGTAAAC	RT–qPCR	This study

Table 3 (continued)

Name	Sequence (5′–3′)	Method used	Reference
GAPDH (ex1/ex3) F	GCTCTCTGCTCCTCCTGTTC	RT–qPCR	This study
GAPDH (ex1/ex3) R	ACGACCAAATCCGTTGACTC	RT–qPCR	This study
CDC6 (PR) F	GGCTGTAACTCTTCCACTGGATTG	ChIP-qPCR	This study
CDC6 (PR) R	CCCGGCCTCGATTCTGATT	ChIP-qPCR	This study
CDC6 (IR) F	AGGTTCCAATATGCATGCTAAGTA	ChIP-qPCR	This study
CDC6 (IR) R	GCCCTTAATAACCTGAAATGGTAATG	ChIP-qPCR	This study
CCNE2 (PR) F	CTACGCGCAGCAACTCCT	ChIP-qPCR	This study
CCNE2 (PR) R	CTGTCCGGAGGTGTCAGTCT	ChIP-qPCR	This study
CCNE2 (IR) F	GACTCCATGACTTCATCCTC	ChIP-qPCR	This study
CCNE2 (IR) R	TGTGACCAGCTGTGATTC	ChIP-qPCR	This study
BRCA1 (PR) F	TATTCTGAGAGGCTGCTGCTTAGCG	ChIP-qPCR	[11]
BRCA1 (PR) R	GGGCCCAGTTATCTGAGAAACCC	ChIP-qPCR	[11]
BRCA1 (IR) F	CCA AAG CCA CCT TTC TGT TCC CAT	ChIP-qPCR	[11]
BRCA1 (IR) R	TCC TGT AAG ACC CTT TGC CTG ACA	ChIP-qPCR	[11]
TOPBP1 (PR) F	GCTCCAACGAGGTAAGTGAG	ChIP-qPCR	This study
TOPBP1 (PR) R	GAAGGCCACAGAAGGCAT	ChIP-qPCR	This study
TOPBP1 (IR) F	CTGGCTCCACATCTCTTCTTC	ChIP-qPCR	This study
TOPBP1 (IR) R	TGGCTCTGCTTAATGCTACTAC	ChIP-qPCR	This study
MCM10 (PR) F	GGCGCCAGACACTCTATTT	ChIP-qPCR	This study
MCM10 (PR) R	GTCATTGGACGCCCTCTTT	ChIP-qPCR	This study
MCM10 (IR) F	CGTGCCTTTCTTAATCAGCATC	ChIP-qPCR	This study
MCM10 (IR) R	GTGCACTGAAGTAGGAGACATAG	ChIP-qPCR	This study
CDC45 (PR) F	TGAATGGCAGAGCGCTAAT	ChIP-qPCR	This study
CDC45 (PR) R	CCAGGGATCACCAACCAATAG	ChIP-qPCR	This study
CDC45 (IR) F	ACTCTGAGCCTGCATTCTTG	ChIP-qPCR	This study
CDC45 (IR) R	AGAAATGTCTGGGCCACATC	ChIP-qPCR	This study
RRM2 (PR) F	GGCATGGCACAGCCAAT	ChIP-qPCR	This study
RRM2 (PR) R	CTCACTCCAGCAGCCTTTAAATC	ChIP-qPCR	This study
RRM2 (IR) F	GGTGGGTGAACACTAGGAATC	ChIP-qPCR	This study
RRM2 (IR) R	AAGGTCGCACAGCACAA	ChIP-qPCR	This study
TOPBP1_9 kb_F	GCATTTCAAGCACCTGAAGATTTA	RT–qPCR	This study
TOPBP1_9 kb_R	AGTCAGGCTAGGAAATGCTAATG	RT–qPCR	This study
TOPBP1_33 kb_F	CCCATCTTGCTTCTCTCTCT	RT–qPCR	This study
TOPBP1_33 kb_R	GGCTGCAAGTGCATCCTATAC	RT–qPCR	This study
MCM10_10 kb_F	AAATAGGGTCCTCCCTGCTC	RT–qPCR	This study
MCM10_10 kb_R	GGTGGTCTTCATCCAACTTATCC	RT–qPCR	This study
MCM10_27 kb_F	GTGTCTGCTCACTGCTGTTT	RT–qPCR	This study
MCM10_27 kb_R	TCTTGTACTGAGCCTGGACAT	RT–qPCR	This study
UBE3C_24 kb_F	TTTCTCTGTTTGGGTGTAGGAG	RT–qPCR	This study
UBE3C_24 kb_R	ACCTCTCTCTTTCTTCTTTCTTCC	RT–qPCR	This study
UBE3C_63 kb_F	CACGGATGATCACAGGGTATG	RT–qPCR	This study
UBE3C_63 kb_R	AGCCCAGTATAAACAGGACTTAAA	RT–qPCR	This study
SETD3_17 kb_F	CAAATCCTCTTTCTTGTGCAGAC	RT–qPCR	This study
SETD3_17 kb_R	CGGACTGCTGCATTCTGTAA	RT–qPCR	This study
SETD3_67 kb_F	GCTTCATTTGGCTCTTGTTAGG	RT–qPCR	This study

Table 3 (continued)

Name	Sequence (5′–3′)	Method used	Reference
SETD3_67 kb_R	TGAGGATGGGTCTGGGAA	RT–qPCR	This study
ARID1A_33 kb_F	GGTTATATATTCAGTGGCCAGAGG	RT–qPCR	This study
ARID1A_33 kb_R	CATTGGACTGGATGGCTACAA	RT–qPCR	This study
ARID1A_77 kb_F	CCTGGGTCAAAGGGTAGATTA	RT–qPCR	This study
ARID1A_77 kb_R	CTGAGGACATGAAGGGATCA	RT–qPCR	This study
CDK12-PCR 1-WT-fwd	GGT GCC TTT TAC CTT GTA TTT GA	PCR	This study
CDK12-PCR 1-AS-fwd	GTG CCT TTT ACC TTG TTG GGG AG	PCR	This study
CDK12-PCR 1-rev	GGA GCA GGT ATG TTT CTC CCA	PCR	This study
CDK12-PCR 2-fwd	GCT CCG TTG TTT ATT ATT AGG AAG G	PCR	This study
CDK12-PCR 2-rev	TCA CTA AAT AGT GTG TGA ATA CTG C	PCR	This study
CDK12-PCR 3-fwd	CCC CCA TGA AGA GGT GAG TAG	PCR	This study
CDK12-PCR 3-rev	GGA GCA GGT ATG TTT CTC CCA	PCR	This study
CDK12-Seq 1-fwd	TAG GAC TTG AGG CAT TGT TAT TTC	Sequencing	This study
CDK12-Seq 1-rev	TTA GAA CAC TTA ATA TCC CGA TGA	Sequencing	This study
CDK12-Seq 2-fwd	GCT CCG TTG TTT ATT ATT AGG AAG G	Sequencing	This study
CDK12-Seq 2-rev	TCA CTA AAT AGT GTG TGA ATA CTG C	Sequencing	This study

by addition of TRIzol reagent. RNA was extracted and relative mRNA levels were analyzed by reverse transcription qPCR (RT–qPCR) as described above, with *HPRT1* as normalization control. Primers spanning exon-exon boundaries were used to assess the percentage of remaining mRNA present after the inhibition of transcription. The list of primer is in the Table 3.

Analysis of elongation rate

Elongation rate experiments on select genes were carried out as described [64]. Briefly, AS CDK12 HCT116 cells were grown overnight on 60-mm dishes to 70–80% confluency and treated with 100 μM DRB (Sigma, D1916) for 3.5 h to synchronize the transcription cycle at the promoter-proximal paused stage. Thirty minutes before DRB removal, the cells were pretreated with either 5 μM 3-MB-PP1 or DMSO (CTRL). After DRB removal, the cells were washed twice with PBS and released into fresh medium containing either 5 μM 3-MB-PP1 or DMSO (CTRL) for transcription restart. The cells were then directly lysed in TRIzol reagent at appropriate time points. 2 μg of total RNA was treated with DNase and reverse transcribed using 200 U SuperScript II RT with random hexamers. Pre-mRNA levels were measured by quantitative RT–qPCR using SYBR Green on AriaMx Real-Time PCR System, as described above. The relative pre-mRNA expression was determined using comparative C_T method ($2^{-\Delta\Delta C_T}$ method) with *HPRT1* as normalizer. Primers spanning exon–intron junctions of select genes were designed using the IDT software PrimerQuest (IDT). The list of primers is in the Table 3.

3′end (PolyA-selected) RNA sequencing

AS CDK12 HCT116 cells were plated on to 60-mm dishes and synchronized by serum starvation as described. At the time of release (0 h) into DMEM containing 15% FBS, cells were treated either with DMSO (CTRL) or 5 μM 3-MB-PP1 for 5 h. Total RNA was isolated from three biological replicates by TRIzol reagent (Thermo Fisher Scientific, 15596026) and purified by RNA QiAamp Spin Column (QIAGEN, 52304), according to the manufacturer's guidelines. RNA quality was assessed by TapeStation 2200 (Agilent Technologies), and only samples with a RIN values ≥ 9 were used for library preparation. PolyA-selected libraries were made from 200 ng of total RNA input using QuantSeq 3′mRNA-Seq Library Prep Kit FWD for Illumina (Lexogen, 015.24) and external multiplexing barcodes for Illumina (i7 index primers 7001-7096; Lexogen, 044.96) with 12× PCR cycles for library amplification, according to manufacturer's instructions. The fragment size and quality of the libraries were assessed by fragment analyzer (Advanced Analytical Technologies) and sequenced with 50 bp single-end reads on a single lane of an Illumina HiSeq 2500 (VBCF Vienna).

Nuclear total RNA-seq

AS CDK12 HCT116 cells were plated onto 150-mm dishes and synchronized by serum starvation for 72 h. Cells were released by adding 15% FBS containing medium with either DMSO (CTRL) or 5 μM 3-MB-PP1 diluted in DMSO. The cells were washed twice with ice-cold PBS 4.5 h after the release, scraped, pelleted at 500 g for 3 min, and lysed in 150 μl of cytoplasmic lysis buffer [10 mM Tris–Cl pH 8, 0.32 M sucrose, 3 mM $CaCl_2$, 2 mM $MgCl_2$, 0.1 mM EDTA, 1 mM DTT, 0.5% Triton X-100, 40 U/ml RNase inhibitor (Roche, 3335402001), and Protease inhibitor cocktail (Sigma, P8340)] for 5 min. Cytoplasmic RNA present in the supernatant was removed by centrifugation (500 *g* for 3 min). Nuclear pellet was washed with 90 μl of cytoplasmic lysis buffer, and supernatant was completely removed after centrifugation (500 *g* for 3 min). Nuclear RNA was isolated from the remaining nuclear pellet using Tri-Reagent (MRC, #TR118). 1 μg of RNA was treated with 1 μl of DNase (Sigma,

AMPD1). 250 ng of nuclear RNA was used for library preparation after removing ribosomal RNA with NEBNext rRNA Depletion Kit (NEB, E6310S). Sequencing libraries were prepared using the NEBNext Ultra II Directional RNA Library Prep Kit for Illumina (NEB, E7760) and NEBNext Multiplex Oligos for Illumina (NEB, E7500S and E7335S) and sequenced with 50 bp at single-end reads on Illumina HiSeq 2500 (VBCF Vienna, Austria).

Chromatin immunoprecipitation (ChIP-qPCR)

ChIP was performed with antibodies indicated in the Table 1. Briefly, 20 μl of protein G Dynabeads (Thermo Fisher Scientific, 10009D) per one immunoprecipitation was washed three times with RIPA buffer (50 mM Tris–Cl, pH 8, 150 mM NaCl, 5 mM EDTA, 1% NP-40, 0.5% sodium deoxycholate, 0.1% SDS, supplemented with protease inhibitors, Sigma, P8340), and pre-blocked with 0.2 mg/ml BSA (Thermo Fisher Scientific, AM2616) and 0.2 mg/ml salmon sperm DNA (Thermo Fisher Scientific, 15632-011) for 4 h. After pre-blocking, the beads were washed three times with RIPA buffer followed by the incubation with specific antibody for at least 4 h at 4°C.

AS CDK12 HCT116 cells were plated onto 150-mm dishes and synchronized by serum starvation as described. The cells were released and incubated with 15% FBS containing medium supplemented with either DMSO or 5 μM 3-MB-PP1 inhibitor diluted in DMSO for 4.5 h. Cells were crosslinked with 1% formaldehyde for 10 min; reaction was quenched with glycine (final concentration 125 mM) for 5 min. Cells were washed twice with ice-cold PBS, scraped, and pelleted. Each 20-μl packed cell pellet was lysed in 600 μl of RIPA buffer and sonicated 20 × 7s (amplitude 0.85) using 5/64 probe (QSonica Q55A). Clarified extracts (13,000 *g* for 10 min) were precleared with protein G Dynabeads (Thermo Fisher Scientific, 10009D) rotating for 2–4 h at 4°C and then incubated overnight with antibody pre-bound to the protein G Dynabeads. We used 1 ml of clarified extract to immunoprecipitate E2F1 or E2F3 proteins. 5% of clarified extract was saved and used as input DNA. Next, day beads were washed sequentially with low salt buffer (20 mM Tris–Cl, pH 8, 150 mM NaCl, 2 mM EDTA, 1% Triton X-100, 0.1% SDS), high salt buffer (20 mM Tris–Cl, pH 8, 500 mM NaCl, 2 mM EDTA, 1% Triton X-100, 0.1% SDS), LiCl buffer (20 mM Tris–Cl, pH 8, 250 mM LiCl, 2 mM EDTA, 1% NP-40, 1% sodium deoxycholate), and twice with TE buffer (10 mM Tris–Cl, pH 8, 1 mM EDTA). Bound complexes were eluted with 500 μl of elution buffer (1% SDS and 0.1 M $NaHCO_3$). To reverse formaldehyde crosslinks, both immunoprecipitated and input DNA were incubated at 65°C for at least 4 h with NaCl at final concentration 0.2 M and subsequently treated with proteinase K at 42°C for 2 h (10 μg/ml, Sigma P5568) with 2 μl of GlycoBlue added (Thermo Fisher Scientific, AM9516). After phenol:chloroform extraction (Sigma, P3803), both immunoprecipitated DNA and input DNAs were dissolved in 200 μl water and 5 μl of DNA served as template for each qPCR reaction. Enrichment of specific gene sequences was measured by qPCR (Agilent AriaMx Real-time PCR System) using SYBR Green JumpStart TaqReadyMix (Sigma, S4438) with following parameters: 95°C for 2 min followed by 45 cycles of denaturation at 95°C for 15 s, annealing at 55°C for 30 s, and extension at 72°C for 30 s. ChIP enrichment of specific target was always determined based on amplification efficiency and C_t value, and calculated relative to the amount of input material. All primer sequences used in this study are specified in the Table 3. qPCR was performed in triplicate for each biological replicate, and error bars represent standard error of the mean of three biological replicates.

ChIP sequencing

ChIP was performed with RNAPII, P-Ser2, P-Ser5, and SPT6 antibodies as described above. AS CDK12 HCT116 cells were plated on to 150-mm dishes and synchronized by serum starvation as mentioned above. At the time of release (0 h) into DMEM containing 15% FBS, the cells were treated either with DMSO (CTRL) or 5 μM 3-MB-PP1 for 4.5 h. For each ChIP sequencing (ChIP-seq) experiment (three biological replicates were processed for each antibody), we performed three technical replicates, and from each replicate, the immunoprecipitated DNA was dissolved in 20 μl H_2O and pooled together. DNA concentration was measured by Qubit fluorometer (Thermo Fisher Scientific), and 4 ng (3.5 ng for SPT6) of immunoprecipitated DNA was used for library preparation. ChIP-seq libraries were generated using the KAPA Biosystems Hyper Prep Kit (KK8502) with KAPA Pure Beads (KK8001), and NEBNext Multiplex Oligos for Illumina (Index Primers Set 1 and Set 2 (NEB, E7335S, E7500S) with 13× (15× for SPT6) PCR cycles for library amplification, as per manufacturer's instructions. Libraries were run on the fragment analyzer (Advanced Analytical Technologies) to check the quality and were sequenced with 50 bp single-end reads on two lanes of an Illumina HiSeq 2500 (VBCF Vienna).

RNA-seq and ChIP-seq analysis

Quality check of RNA-seq reads was performed using fastQC (available online at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc). RNA-seq reads were mapped against the human genome (hg38) and human rRNA sequences using ContextMap version 2.7.9 [80] (using BWA [81] as short read aligner and default parameters). Number of read counts per gene and exon were determined from the mapped RNA-seq reads in a strand-specific manner using featureCounts [82] and gene annotations from GENCODE version 27. Differential gene expression analysis was performed using DESeq2 [83]. Differential exon usage was determined using DEXSeq [60]. *P*-values were adjusted for multiple testing using the method by Benjamini and Hochberg [84], and genes and exons with an adjusted *P*-value ≤ 0.01 were considered significantly differentially expressed and used, respectively. Functional enrichment analysis of differentially expressed genes for Gene Ontology terms was performed with the GOrilla webserver [85]. In addition, gene set enrichment analysis (GSEA) [51] based on log2 fold-changes of all genes was performed. Analysis workflows were implemented and run using the Watchdog workflow management system [86].

Regulated poly(A) sites (PAS) were identified from 3′end RNA-seq data in the following way: First, occurrences of polyadenylation signal sequences as defined by [87] as well as occurrences of at least 10 consecutive As (to exclude internal poly(A) priming) were identified in the genome on both strands. Second, windows around the poly(A) signal sequences (−300 bp upstream of signal to 50 bp downstream of signal to include the actual PAS) and oligo-As (−350 bp upstream of oligo-A until end of the 10 As) were defined. All overlapping poly(A) signal windows and oligo-A windows were merged and poly(A) signal windows overlapping with an oligo-A

window were removed. Third, read counts were determined for remaining poly(A) signal windows using featureCounts in each 3′end RNA-seq sample and differential gene expression analysis was performed using DEseq2 as described above.

ChIP-seq reads were aligned to the human genome (hg38) using BWA [81]. Reads with an alignment score < 20 were discarded. Read coverage per genome position was calculated using the bedtools genomecov tool [88]. ChIP-seq and RNA-seq read coverage was visualized using Gviz [89]. For this purpose, read counts were normalized to the total number of mapped reads and averaged between replicates. Creation of other figures and statistical analysis of RNA-seq and ChIP-seq data were performed in R [90].

X% distance (i.e., 10, 50 and 90% distance) for ChIP-seq and nuclear RNA-seq data were calculated as the minimum distance in bps from the transcription start site (TSS) at which X% of the total read coverage of the gene was obtained. Absolute ΔX% distance was defined as the difference of X% distance in control minus the X% distance in inhibitor-treated cells. Relative ΔX% distance was defined as absolute ΔX% distance divided by gene length.

Metagene analysis

The metagene analysis of read coverage distribution in ChIP-seq data was restricted to high confident transcripts of protein-coding genes annotated in GENCODE version 27. Transcripts shorter than 3,180 bp were excluded. For each gene, we selected the transcript with the most read counts in the RNAPII ChIP-seq samples (normalized to library size) in the ±3 kb regions around the transcription start site (TSS) and transcription termination site (TTS). For each gene, the regions −3 kb to +1.5 kb of the TSS and −1.5 kb to +3 kb of the TTS were divided into 50 bp bins (180 bins in total) and the remainder of the gene body (+1.5 kb of TSS to −1.5 kb of TTS) into 180 bins of variable length in order to compare genes with different lengths. For each bin, the average coverage per genome position was then calculated and normalized to the total sum of average coverages per bin such that the sum of all bins was 1. Finally, metagene plots were created by averaging results for corresponding bins across all genes considered. To determine statistical significance of differences between inhibitor and control, paired Wilcoxon signed rank tests were performed for each bin comparing normalized coverage values for each gene for this bin with and without the inhibitor. P-values were adjusted for multiple testing with the Bonferroni method across all bins within each subfigure and are color-coded in the bottom track of each subfigure: red = adj. P-value $\leq 10^{-15}$; orange = adj. P-value $\leq 10^{-10}$; yellow: adj. P-value $\leq 10^{-3}$.

Statistical analysis

All experiments were performed at least in three or more biological replicates. Results are reported as means ± standard error of the mean (SEM) unless stated otherwise. All graphics and statistics (except for RNA-seq and ChIP-seq) were generated using *Microsoft Excel*.

Data availability

All RNA-seq and ChIP-seq data have been submitted to the Gene Expression Omnibus (GEO) and are available under the accession GSE120072. A UCSC genome browser session showing the mapped RNA-seq and ChIP-seq data is available at: https://genome.ucsc.edu/s/CFriedel/CDK12.

Expanded View for this article is available online.

Acknowledgements

We thank all members of the Blazek laboratory for discussions throughout the project and helpful comments on the article. We also wish to thank Tomas Loja for help with flow cytometry, Kamila Reblova for help with ChIP-seq data visualization, Stjepan Uldrijan for P53 and Dasa Bohaciakova for phospho-P53 antibodies, VBCF Vienna for sequencing, and Core Facility Bioinformatics of CEITEC Masaryk University is gratefully acknowledged for the obtaining of the scientific data presented in this paper. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures". The work was supported by the following grants: the project CZ-OPENSCREEN: National Infrastructure for Chemical Biology (identification code: LM2015063), the project no. LQ1605 from the National Program of Sustainability II (MEYS CR) to K.P.; the Czech Science Foundation ("17-13692S"), the CEITEC [Project "CEITEC-Central-European Institute of Technology" (CZ.1.05/1.1.00/02.0068)], the Grant agency of Masaryk university (MUNI/E/0514/2019) to D.B.; European Regional Development Fund—Project "MSCAfellow@MUNI" (CZ.02.2.69/0.0/0.0/17_050/0008496) to A.M.; the Deutsche Forschungsgemeinschaft [FR2938/7-1 and CRC 1123 (Z2)] to C.C.F; and Czech Science Foundation (17-17720S), Wellcome Trust Collaborative Grant (206292/E/17/Z), and National Program of Sustainability II (MEYS CR, project no. LQ1605) to L.K.

Author contributions

APCM, KPi, and MR performed experiments. MK performed bioinformatics analyses under supervision of CCF and with some input from JO and DB. DB conceived the study, acquired funding, and wrote the article with support of CCF, APCM, and KPi. KB and LK contributed to design of experiments, and PK synthetized THZ531 under supervision of KPa. All authors discussed the design of experiments, analyzed the data, and commented on the article.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Fuda NJ, Ardehali MB, Lis JT (2009) Defining mechanisms that regulate RNA polymerase II transcription *in vivo*. *Nature* 461: 186 – 192
2. Harlen KM, Churchman LS (2017) The code and beyond: transcription regulation by the RNA polymerase II carboxy-terminal domain. *Nat Rev Mol Cell Biol* 18: 263 – 273
3. Adelman K, Lis JT (2012) Promoter-proximal pausing of RNA polymerase II: emerging roles in metazoans. *Nat Rev Genet* 13: 720 – 731
4. Buratowski S (2003) The CTD code. *Nat Struct Biol* 10: 679 – 680
5. Egloff S, Murphy S (2008) Cracking the RNA polymerase II CTD code. *Trends Genet* 24: 280 – 288
6. Eick D, Geyer M (2013) The RNA polymerase II carboxy-terminal domain (CTD) code. *Chem Rev* 113: 8456 – 8490
7. Zaborowska J, Egloff S, Murphy S (2016) The pol II CTD: new twists in the tail. *Nat Struct Mol Biol* 23: 771 – 777

8. Peterlin BM, Price DH (2006) Controlling the elongation phase of transcription with P-TEFb. *Mol Cell* 23: 297 – 305
9. Larochelle S, Amat R, Glover-Cutter K, Sanso M, Zhang C, Allen JJ, Shokat KM, Bentley DL, Fisher RP (2012) Cyclin-dependent kinase control of the initiation-to-elongation switch of RNA polymerase II. *Nat Struct Mol Biol* 19: 1108 – 1115
10. Drogat J, Hermand D (2012) Gene-specific requirement of RNA polymerase II CTD phosphorylation. *Mol Microbiol* 84: 995 – 1004
11. Blazek D, Kohoutek J, Bartholomeeusen K, Johansen E, Hulinkova P, Luo Z, Cimermancic P, Ule J, Peterlin BM (2011) The Cyclin K/Cdk12 complex maintains genomic stability via regulation of expression of DNA damage response genes. *Genes Dev* 25: 2158 – 2172
12. Bartkowiak B, Liu P, Phatnani HP, Fuda NJ, Cooper JJ, Price DH, Adelman K, Lis JT, Greenleaf AL (2010) CDK12 is a transcription elongation-associated CTD kinase, the metazoan ortholog of yeast Ctk1. *Genes Dev* 24: 2303 – 2316
13. Bosken CA, Farnung L, Hintermair C, Merzel Schachter M, Vogel-Bachmayr K, Blazek D, Anand K, Fisher RP, Eick D, Geyer M (2014) The structure and substrate specificity of human Cdk12/Cyclin K. *Nat Commun* 5: 3505
14. Cheng SW, Kuzyk MA, Moradian A, Ichu TA, Chang VC, Tien JF, Vollett SE, Griffith M, Marra MA, Morin GB (2012) Interaction of cyclin-dependent kinase 12/CrkRS with cyclin K1 is required for the phosphorylation of the C-terminal domain of RNA polymerase II. *Mol Cell Biol* 32: 4691 – 4704
15. Yu M, Yang W, Ni T, Tang Z, Nakadai T, Zhu J, Roeder RG (2015) RNA polymerase II-associated factor 1 regulates the release and phosphorylation of paused RNA polymerase II. *Science* 350: 1383 – 1386
16. Greifenberg AK, Honig D, Pilarova K, Duster R, Bartholomeeusen K, Bosken CA, Anand K, Blazek D, Geyer M (2016) Structural and functional analysis of the Cdk13/Cyclin K complex. *Cell Rep* 14: 320 – 331
17. Zhang T, Kwiatkowski N, Olson CM, Dixon-Clarke SE, Abraham BJ, Greifenberg AK, Ficarro SB, Elkins JM, Liang Y, Hannett NM *et al* (2016) Covalent targeting of remote cysteine residues to develop CDK12 and CDK13 inhibitors. *Nat Chem Biol* 12: 876 – 884
18. Davidson L, Muniz L, West S (2014) 3′ end formation of pre-mRNA and phosphorylation of Ser2 on the RNA polymerase II CTD are reciprocally coupled in human cells. *Genes Dev* 28: 342 – 356
19. Edwards MC, Wong C, Elledge SJ (1998) Human cyclin K, a novel RNA polymerase II-associated cyclin possessing both carboxy-terminal domain kinase and Cdk-activating kinase activity. *Mol Cell Biol* 18: 4291 – 4300
20. Kohoutek J, Blazek D (2012) Cyclin K goes with Cdk12 and Cdk13. *Cell Div* 7: 12
21. Ekumi KM, Paculova H, Lenasi T, Pospichalova V, Bosken CA, Rybarikova J, Bryja V, Geyer M, Blazek D, Barboric M (2015) Ovarian carcinoma CDK12 mutations misregulate expression of DNA repair genes via deficient formation and function of the Cdk12/CycK complex. *Nucleic Acids Res* 43: 2575 – 2589
22. Juan HC, Lin Y, Chen HR, Fann MJ (2016) Cdk12 is essential for embryonic development and the maintenance of genomic stability. *Cell Death Differ* 23: 1038 – 1048
23. Liang K, Gao X, Gilmore JM, Florens L, Washburn MP, Smith E, Shilatifard A (2015) Characterization of human cyclin-dependent kinase 12 (CDK12) and CDK13 complexes in C-terminal domain phosphorylation, gene transcription, and RNA processing. *Mol Cell Biol* 35: 928 – 938
24. Hoshii T, Cifani P, Feng Z, Huang CH, Koche R, Chen CW, Delaney CD, Lowe SW, Kentsis A, Armstrong SA (2018) A non-catalytic function of SETD1A regulates cyclin K and the DNA damage response. *Cell* 172: 1007 – 1021 e17
25. Eifler TT, Shao W, Bartholomeeusen K, Fujinaga K, Jager S, Johnson J, Luo Z, Krogan N, Peterlin BM (2014) CDK12 increases 3′ end processing of growth factor-induced c-FOS transcripts. *Mol Cell Biol* 35: 468 – 478
26. Ko TK, Kelly E, Pines J (2001) CrkRS: a novel conserved Cdc2-related protein kinase that colocalises with SC35 speckles. *J Cell Sci* 114: 2591 – 2603
27. Chen HH, Wang YC, Fann MJ (2006) Identification and characterization of the CDK12/cyclin L1 complex involved in alternative splicing regulation. *Mol Cell Biol* 26: 2736 – 2745
28. Tien JF, Mazloomian A, Cheng SG, Hughes CS, Chow CCT, Canapi LT, Oloumi A, Trigo-Gonzalez G, Bashashati A, Xu J *et al* (2017) CDK12 regulates alternative last exon mRNA splicing and promotes breast cancer cell invasion. *Nucleic Acids Res* 45: 6698 – 6716
29. Popova T, Manie E, Boeva V, Battistella A, Goundiam O, Smith NK, Mueller CR, Raynal V, Mariani O, Sastre-Garau X *et al* (2016) Ovarian cancers harboring inactivating mutations in CDK12 display a distinct genomic instability pattern characterized by large tandem duplications. *Can Res* 76: 1882 – 1891
30. Wu YM, Cieslik M, Lonigro RJ, Vats P, Reimers MA, Cao X, Ning Y, Wang L, Kunju LP, de Sarkar N *et al* (2018) Inactivation of CDK12 delineates a distinct immunogenic class of advanced prostate cancer. *Cell* 173: 1770 – 1782 e14
31. Menghi F, Barthel FP, Yadav V, Tang M, Ji B, Tang Z, Carter GW, Ruan Y, Scully R, Verhaak RGW *et al* (2018) The tandem duplicator phenotype is a prevalent genome-wide cancer configuration driven by distinct gene mutations. *Cancer Cell* 34: 197 – 210 e5
32. Rao M, Powers S (2018) Tandem duplications may supply the missing genetic alterations in many triple-negative breast and gynecological cancers. *Cancer Cell* 34: 179 – 180
33. Menghi F, Inaki K, Woo X, Kumar PA, Grzeda KR, Malhotra A, Yadav V, Kim H, Marquez EJ, Ucar D *et al* (2016) The tandem duplicator phenotype as a distinct genomic configuration in cancer. *Proc Natl Acad Sci USA* 113: E2373 – E2382
34. Johnson SF, Cruz C, Greifenberg AK, Dust S, Stover DG, Chi D, Primack B, Cao S, Bernhardy AJ, Coulson R *et al* (2016) CDK12 inhibition reverses *de novo* and acquired PARP inhibitor resistance in BRCA wild-type and mutated models of triple-negative breast cancer. *Cell Rep* 17: 2367 – 2381
35. Joshi PM, Sutor SL, Huntoon CJ, Karnitz LM (2014) Ovarian cancer-associated mutations disable catalytic activity of CDK12, a kinase that promotes homologous recombination repair and resistance to cisplatin and poly(ADP-ribose) polymerase inhibitors. *J Biol Chem* 289: 9247 – 9253
36. Bajrami I, Frankum JR, Konde A, Miller RE, Rehman FL, Brough R, Campbell J, Sims D, Rafiq R, Hooper S *et al* (2014) Genome-wide profiling of genetic synthetic lethality identifies CDK12 as a novel determinant of PARP1/2 inhibitor sensitivity. *Can Res* 74: 287 – 297
37. Iniguez AB, Stolte B, Wang EJ, Conway AS, Alexe G, Dharia NV, Kwiatkowski N, Zhang T, Abraham BJ, Mora J *et al* (2018) EWS/FLI confers tumor cell synthetic lethality to CDK12 inhibition in ewing sarcoma. *Cancer Cell* 33: 202 – 216 e6
38. Malumbres M, Barbacid M (2009) Cell cycle, CDKs and cancer: a changing paradigm. *Nat Rev Cancer* 9: 153 – 166
39. Bertoli C, Skotheim JM, de Bruin RA (2013) Control of cell cycle transcription during G1 and S phases. *Nat Rev Mol Cell Biol* 14: 518 – 528

40. Bracken AP, Ciro M, Cocito A, Helin K (2004) E2F target genes: unraveling the biology. *Trends Biochem Sci* 29: 409 – 417
41. Tatsumi Y, Sugimoto N, Yugawa T, Narisawa-Saito M, Kiyono T, Fujita M (2006) Deregulation of Cdt1 induces chromosomal damage without rereplication and leads to chromosomal instability. *J Cell Sci* 119: 3128 – 3140
42. Liontos M, Koutsami M, Sideridou M, Evangelou K, Kletsas D, Levy B, Kotsinas A, Nahum O, Zoumpourlis V, Kouloukoussa M *et al* (2007) Deregulated overexpression of hCdt1 and hCdc6 promotes malignant behavior. *Can Res* 67: 10899 – 10909
43. Tsantoulis PK, Gorgoulis VG (2005) Involvement of E2F transcription factor family in cancer. *Eur J Cancer* 41: 2403 – 2414
44. Baxley RM, Bielinsky AK (2017) Mcm10: a dynamic scaffold at eukaryotic replication forks. *Genes (Basel)* 8: E73
45. Lopez MS, Kliegman JI, Shokat KM (2014) The logic and design of analog-sensitive kinases and their small molecule inhibitors. *Methods Enzymol* 548: 189 – 213
46. Larochelle S, Batliner J, Gamble MJ, Barboza NM, Kraybill BC, Blethrow JD, Shokat KM, Fisher RP (2006) Dichotomous but stringent substrate selection by the dual-function Cdk7 complex revealed by chemical genetics. *Nat Struct Mol Biol* 13: 55 – 62
47. Galbraith MD, Andrysik Z, Pandey A, Hoh M, Bonner EA, Hill AA, Sullivan KD, Espinosa JM (2017) CDK8 kinase activity promotes glycolysis. *Cell Rep* 21: 1495 – 1506
48. Bartkowiak B, Yan C, Greenleaf AL (2015) Engineering an analog-sensitive CDK12 cell line using CRISPR/Cas. *Biochem Biophys Acta* 1849: 1179 – 1187
49. Blazek D (2012) The cyclin K/Cdk12 complex: an emerging new player in the maintenance of genome stability. *Cell Cycle* 11: 1049 – 1050
50. Shiloh Y, Ziv Y (2013) The ATM protein kinase: regulating the cellular response to genotoxic stress, and more. *Nat Rev Mol Cell Biol* 14: 197 – 210
51. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES *et al* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA* 102: 15545 – 15550
52. Fragkos M, Ganier O, Coulombe P, Mechali M (2015) DNA replication origin activation in space and time. *Nat Rev Mol Cell Biol* 16: 360 – 374
53. Beltran M, Yates CM, Skalska L, Dawson M, Reis FP, Viiri K, Fisher CL, Sibley CR, Foster BM, Bartke T *et al* (2016) The interaction of PRC2 with RNA or chromatin is mutually antagonistic. *Genome Res* 26: 896 – 907
54. Jackson SP, Bartek J (2009) The DNA-damage response in human biology and disease. *Nature* 461: 1071 – 1078
55. Rahl PB, Lin CY, Seila AC, Flynn RA, McCuine S, Burge CB, Sharp PA, Young RA (2010) c-Myc regulates transcriptional pause release. *Cell* 141: 432 – 445
56. Sanso M, Fisher RP (2013) Pause, play, repeat: CDKs push RNAP II's buttons. *Transcription* 4: 146 – 152
57. Ardehali MB, Yao J, Adelman K, Fuda NJ, Petesch SJ, Webb WW, Lis JT (2009) Spt6 enhances the elongation rate of RNA polymerase II *in vivo*. *EMBO J* 28: 1067 – 1077
58. Vos SM, Farnung L, Boehning M, Wigge C, Linden A, Urlaub H, Cramer P (2018) Structure of activated transcription complex Pol II-DSIF-PAF-SPT6. *Nature* 560: 607 – 612
59. Yoh SM, Cho H, Pickle L, Evans RM, Jones KA (2007) The Spt6 SH2 domain binds Ser2-P RNAPII to direct Iws1-dependent mRNA splicing and export. *Genes Dev* 21: 160 – 174
60. Anders S, Reyes A, Huber W (2012) Detecting differential usage of exons from RNA-seq data. *Genome Res* 22: 2008 – 2017
61. Gu B, Eick D, Bensaude O (2013) CTD serine-2 plays a critical role in splicing and termination factor recruitment to RNA polymerase II *in vivo*. *Nucleic Acids Res* 41: 1591 – 1603
62. Herzel L, Ottoz DSM, Alpert T, Neugebauer KM (2017) Splicing and transcription touch base: co-transcriptional spliceosome assembly and function. *Nat Rev Mol Cell Biol* 18: 637 – 650
63. Dubbury SJ, Boutz PL, Sharp PA (2018) CDK12 regulates DNA repair genes by suppressing intronic polyadenylation. *Nature* 564: 141 – 145
64. Singh J, Padgett RA (2009) Rates of *in situ* transcription and splicing in large human genes. *Nat Struct Mol Biol* 16: 1128 – 1133
65. Fitz J, Neumann T, Pavri R (2018) Regulation of RNA polymerase II processivity by Spt5 is restricted to a narrow window during elongation. *EMBO J* 37: e97965
66. Jonkers I, Kwak H, Lis JT (2014) Genome-wide dynamics of Pol II elongation and its interplay with promoter proximal pausing, chromatin, and exons. *Elife* 3: e02407
67. Branzei D, Szakal B (2017) Building up and breaking down: mechanisms controlling recombination during replication. *Crit Rev Biochem Mol Biol* 52: 381 – 394
68. Gaillard H, Garcia-Muse T, Aguilera A (2015) Replication stress and cancer. *Nat Rev Cancer* 15: 276 – 289
69. Zeman MK, Cimprich KA (2014) Causes and consequences of replication stress. *Nat Cell Biol* 16: 2 – 9
70. Choi SH, Martinez TF, Kim S, Donaldson C, Shokhirev MN, Saghatelian A, Jones KA (2019) CDK12 phosphorylates 4E-BP1 to enable mTORC1-dependent translation and mitotic genome stability. *Genes Dev* 33: 418 – 435
71. Paculova H, Kramara J, Simeckova S, Fedr R, Soucek K, Hylse O, Paruch K, Svoboda M, Mistrik M, Kohoutek J (2017) BRCA1 or CDK12 loss sensitizes cells to CHK1 inhibitors. *Tumour Biol* 39: 1010428317727479
72. Lei T, Zhang P, Zhang X, Xiao X, Zhang J, Qiu T, Dai Q, Zhang Y, Min L, Li Q *et al* (2018) Cyclin K regulates prereplicative complex assembly to promote mammalian cell proliferation. *Nat Commun* 9: 1876
73. Qiu H, Hu C, Hinnebusch AG (2009) Phosphorylation of the Pol II CTD by KIN28 enhances BUR1/BUR2 recruitment and Ser2 CTD phosphorylation near promoters. *Mol Cell* 33: 752 – 762
74. Devaiah BN, Lewis BA, Cherman N, Hewitt MC, Albrecht BK, Robey PG, Ozato K, Sims III RJ, Singer DS (2012) BRD4 is an atypical kinase that phosphorylates serine2 of the RNA polymerase II carboxy-terminal domain. *Proc Natl Acad Sci USA* 109: 6927 – 6932
75. Booth GT, Parua PK, Sanso M, Fisher RP, Lis JT (2018) Cdk9 regulates a promoter-proximal checkpoint to modulate RNA polymerase II elongation rate in fission yeast. *Nat Commun* 9: 543
76. Egloff S, O'Reilly D, Chapman RD, Taylor A, Tanzhaus K, Pitts L, Eick D, Murphy S (2007) Serine-7 of the RNA polymerase II CTD is specifically required for snRNA gene expression. *Science* 318: 1777 – 1779
77. Ran FA, Hsu PD, Wright J, Agarwala V, Scott DA, Zhang F (2013) Genome engineering using the CRISPR-Cas9 system. *Nat Protoc* 8: 2281 – 2308
78. Gratzner HG (1982) Monoclonal antibody to 5-bromo- and 5-iododeoxyuridine: a new reagent for detection of DNA replication. *Science* 218: 474 – 475
79. Schlacher K, Christ N, Siaud N, Egashira A, Wu H, Jasin M (2011) Double-strand break repair-independent role for BRCA2 in

blocking stalled replication fork degradation by MRE11. *Cell* 145: 529 – 542

80. Bonfert T, Kirner E, Csaba G, Zimmer R, Friedel CC (2015) ContextMap 2: fast and accurate context-based RNA-seq mapping. *BMC Bioinformatics* 16: 122
81. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25: 1754 – 1760
82. Liao Y, Smyth GK, Shi W (2014) featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* 30: 923 – 930
83. Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* 15: 550
84. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate – a practical and powerful approach to multiple testing. *J R Stat Soc Series B Stat Methodol* 57: 289 – 300
85. Eden E, Navon R, Steinfeld I, Lipson D, Yakhini Z (2009) GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics* 10: 48
86. Kluge M, Friedel CC (2018) Watchdog – a workflow management system for the distributed analysis of large-scale experimental data. *BMC Bioinformatics* 19: 97
87. Gruber AR, Martin G, Keller W, Zavolan M (2014) Means to an end: mechanisms of alternative polyadenylation of messenger RNA precursors. *Wiley Interdiscip Rev RNA* 5: 183 – 196
88. Quinlan AR, Hall IM (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* 26: 841 – 842
89. Hahne F, Ivanek R (2016) Visualizing genomic data using gviz and bioconductor. *Methods Mol Biol* 1418: 335 – 351
90. R Core Team (2016) *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing

Expanded View Figures

Figure EV1. Preparation and characterization of AS CDK12 HCT116 cell line.

A Depiction of *CDK12* locus, genome editing, and genotyping strategy. Schema of *CDK12* locus, with exon numbers shown above the *CDK12* gene depiction (top). Primers used for genotyping PCR surrounding exon 6 of *CDK12* gene are shown as horizontal arrows, PCR product is depicted as full horizontal line, and *BslI* restriction sites are indicated by vertical arrows. *BslI* restriction site created by genome editing is shown in green. Size (bp) of genotyping PCR product and *BslI* restriction fragments are indicated (middle). DNA subjected to genome editing and corresponding protein sequences in exon 6 of *CDK12* genes are shown; the underlined DNA sequence in WT *CDK12* allele underwent genome editing to create silent mutation preventing alternative splicing (nucleotide in blue), *BslI* restriction site, and to convert F813 to G813 (nucleotides in red) in AS CDK12. Engineered G813 in AS CDK12 is indicated in red (bottom).

B Characterization of AS CDK12 clone by a AS primer-specific PCR. Exon 6 in *CDK12* gene is shown as a black box. Edited DNA in the AS CDK12 is marked by a red vertical line in the exon 6. Genotyping primers specific for WT (black arrows) and AS CDK12 (red arrow) are shown, and genotyping PCR product is depicted by a dashed line with size (in bp) indicated above (top). Ethidium bromide-stained agarose gel visualizing 352 bp PCR product from PCR mixture using either WT- (left) or AS-specific (right) forward primer (bottom).

C CCNK/CDK12 complex shows comparable properties in the AS and WT CDK12 HCT116 cell lines. Western blot analysis of protein levels (input) and association [determined by immunoprecipitation (IP)] of CCNK and CDK12 in the indicated cell lines. No Ab corresponds to a control immunoprecipitation without antibody. A representative image of three replicates is shown.

D Quantification of individual P-Ser modifications in the CTD of RNAPII after CDK12 inhibition. Amounts of individual proteins and CTD modifications presented in Fig 1D and in another two biological replicates from short film exposures were quantified by ImageJ software. All protein levels were normalized to a corresponding tubulin loading control, and samples without treatment in each time point (CTRL) were considered as 1; $n = 3$ biological replicates and error bars are standard error of the mean (SEM).

Source data are available online for this figure.

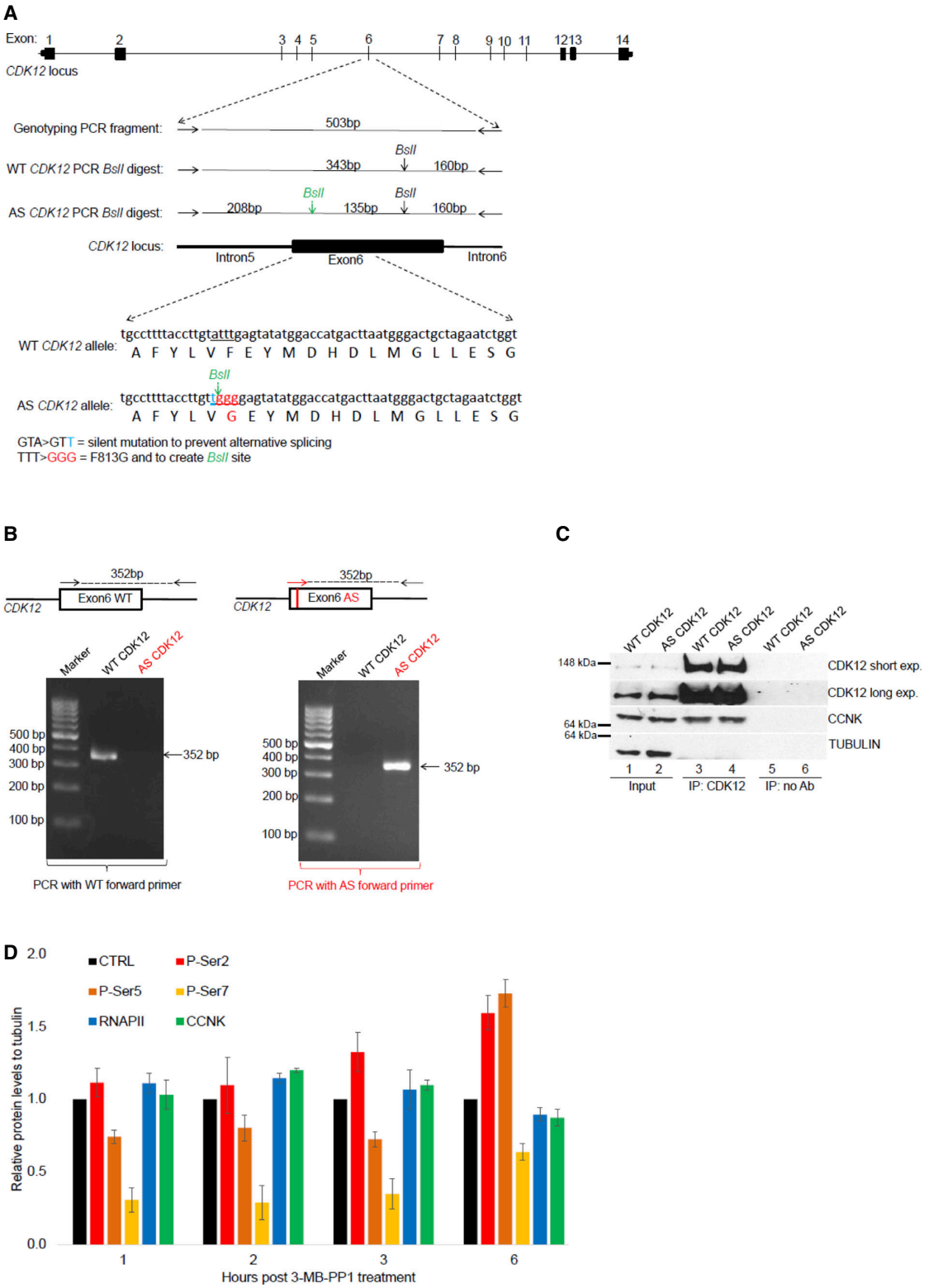

A
Exon: 1 2 3 4 5 6 7 8 9 10 11 12 13 14
CDK12 locus
Genotyping PCR fragment: 503bp
WT CDK12 PCR BslI digest: 343bp BslI 160bp
AS CDK12 PCR BslI digest: 208bp BslI 135bp BslI 160bp
CDK12 locus: Intron5 Exon6 Intron6
WT CDK12 allele: tgccttttaccttgtatttgagtatatggaccatgacttaatgggactgctagaatctggt
A F Y L V F E Y M D H D L M G L L E S G
AS CDK12 allele: tgccttttaccttgttggggagtatatggaccatgacttaatgggactgctagaatctggt
A F Y L V G E Y M D H D L M G L L E S G
GTA>GTT = silent mutation to prevent alternative splicing
TTT>GGG = F813G and to create BslI site
B
352bp
CDK12
Exon6 WT
Exon6 AS
Marker
WT CDK12
AS CDK12
500 bp
400 bp
300 bp
200 bp
100 bp
352 bp
PCR with WT forward primer
PCR with AS forward primer
C
WT CDK12
AS CDK12
148 kDa
64 kDa
64 kDa
CDK12 short exp.
CDK12 long exp.
CCNK
TUBULIN
1 2 3 4 5 6
Input
IP: CDK12
IP: no Ab
D
CTRL
P-Ser2
P-Ser5
P-Ser7
RNAPII
CCNK
Relative protein levels to tubulin
2.0
1.5
1.0
0.5
0.0
1 2 3 6
Hours post 3-MB-PP1 treatment

Figure EV1.

Figure EV2. CDK12 kinase activity is essential for optimal G1/S progression.

A 3-MB-PP1 does not affect cell cycle progression in WT HCT116 cells. The experiment was performed as shown in Fig 2A. n = 3; representative result is shown.

B THZ531 causes G1/S progression defect in WT HCT116 cells arrested by serum starvation. Flow cytometry profiles of control (−THZ531) or 350 nM THZ531(+THZ531)-treated cells from the experiment outlined in Fig 2A. Red arrow points to the onset of the G1/S progression defect in THZ531-treated cells. n = 3 replicates; representative result is shown.

C CDK12 inhibition delays G1/S progression in thymidine/nocodazole-arrested AS CDK12 HeLa cells. Flow cytometry profiles of control (−3-MB-PP1) or 3-MB-PP1 (+3-MB-PP1) treated cells from the experiment shown in Fig 2A. Red arrow points to the onset of the G1/S progression defect in 3-MB-PP1-treated cells. n = 3 replicates; representative result is shown.

D Experimental outline. AS CDK12 HCT116 cells were arrested by serum starvation for 72 h and released into the serum-containing medium with (+) or without (−) 3-MB-PP1. 3-MB-PP1 was washed away and replaced with fresh medium at indicated times after the release, and all samples were subjected to flow cytometry analyses at 15 h after the release.

E G1/S progression delay can be rescued by removal of CDK12 inhibitor at early G1 phase. Flow cytometry profiles of propidium iodide-labeled cells from the experiment depicted in Fig EV2D. CTRL = control samples without the 3-MB-PP1. n = 3 replicates; representative result is shown.

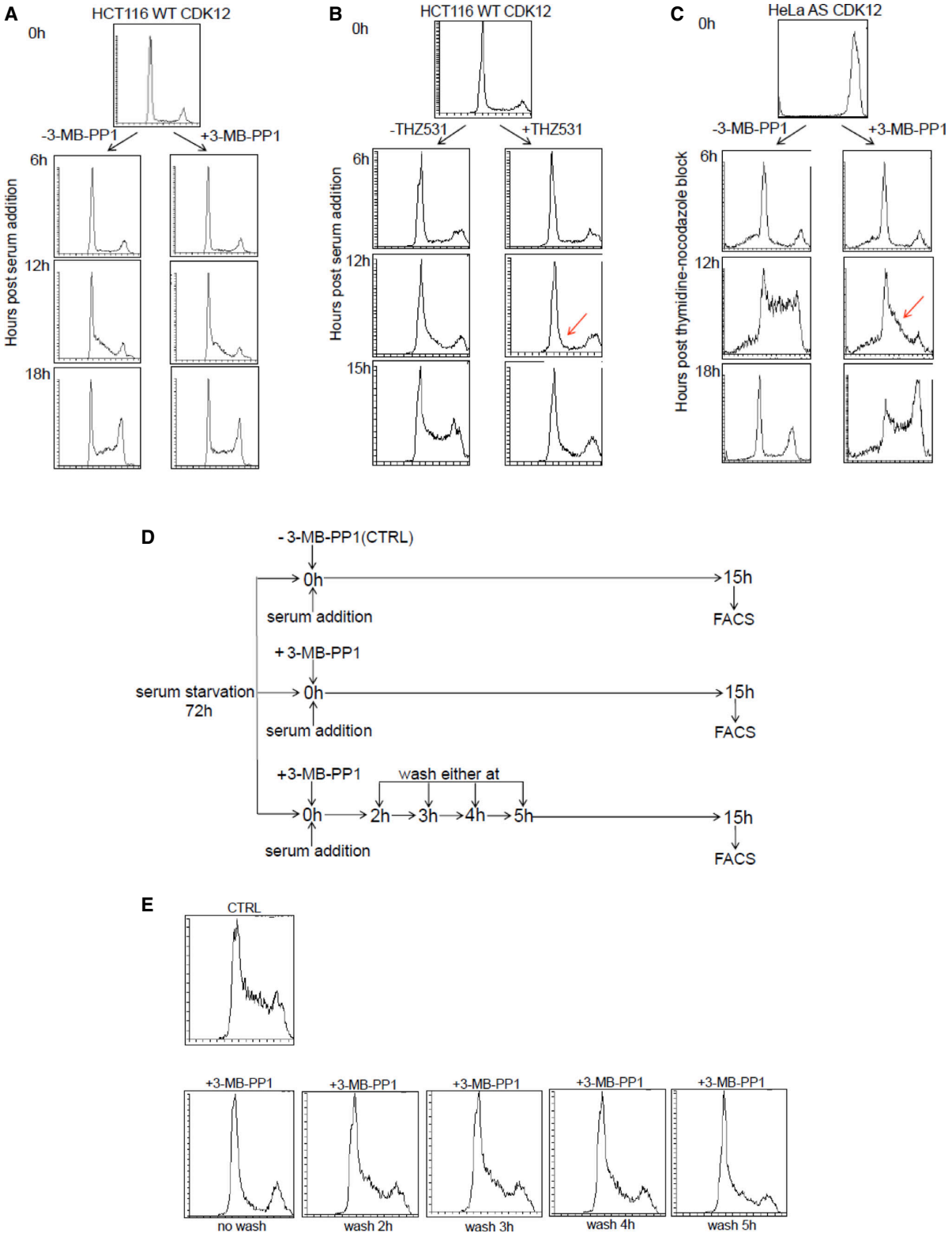

A
HCT116 WT CDK12
0h
-3-MB-PP1
+3-MB-PP1
6h
12h
18h
Hours post serum addition
B
HCT116 WT CDK12
0h
-THZ531
+THZ531
6h
12h
15h
Hours post serum addition
C
HeLa AS CDK12
0h
-3-MB-PP1
+3-MB-PP1
6h
12h
18h
Hours post thymidine-nocodazole block
D
- 3-MB-PP1(CTRL)
0h
15h
serum addition
FACS
+ 3-MB-PP1
serum starvation
72h
0h
15h
serum addition
FACS
+3-MB-PP1
wash either at
0h
2h
3h
4h
5h
15h
serum addition
FACS
E
CTRL
+3-MB-PP1
+3-MB-PP1
+3-MB-PP1
+3-MB-PP1
+3-MB-PP1
no wash
wash 2h
wash 3h
wash 4h
wash 5h

Figure EV2.

Figure EV3. CDK12 catalytic activity controls expression of core DNA replication genes.

A CDK12 inhibition down-regulates DNA replication-related genes. GSEA analysis based on log2 fold-changes in 3′end RNA-seq data upon CDK12 inhibition. Normalized enrichment scores (NES) are shown for significant GO terms (FDR q-val < 0.05) with negative NES, i.e., associated with down-regulation. Functions related to DNA replication are marked by the red rectangles.

B Expression of crucial DNA replication genes is dependent on the CDK12 kinase activity. Comparison of log2 fold-changes versus log2 mean expression in 3′end RNA-seq data and depicts down-regulated DNA replication genes (−0.85 > log2 fold-change, P < 0.01) after 5-h CDK12 inhibition.

C Validation of 3′end RNA-seq for select non-regulated genes by RT–qPCR. See Fig 3D for legend. n = 3 replicates, error bars represent SEM.

D Inhibition of CDK12 kinase does not affect mRNA degradation of select DNA repair and replication transcripts. AS CDK12 HCT116 cells were treated with ActD (1 μg/ml) either in the presence (red line) or absence (CTRL) (blue line) of 3-MB-PP1. Total mRNA was isolated at indicated time points, and levels of indicated mRNAs normalized to *HPRT1* were measured by RT–qPCR. Graphs present mRNA levels relative to untreated cells (time 0 h set to 1). n = 3 independent experiments, error bars are SEM.

E Expression of core DNA replication proteins is dependent on the CDK12 kinase activity. See legend in Fig 3E.

F, G CCNK depletion diminishes mRNA and protein expression of DNA replication genes. RT–qPCR of mRNA levels (F) and Western blot of protein levels (G) in AS CDK12 HCT116 cells treated with control (CTRL) or CCNK siRNAs for 36 h. mRNA levels were normalized to *GAPDH* mRNA expression. n = 3 replicates for RT–qPCR (F), error bars indicate SEM. In (G), a representative experiment from three replicates is shown.

Source data are available online for this figure.

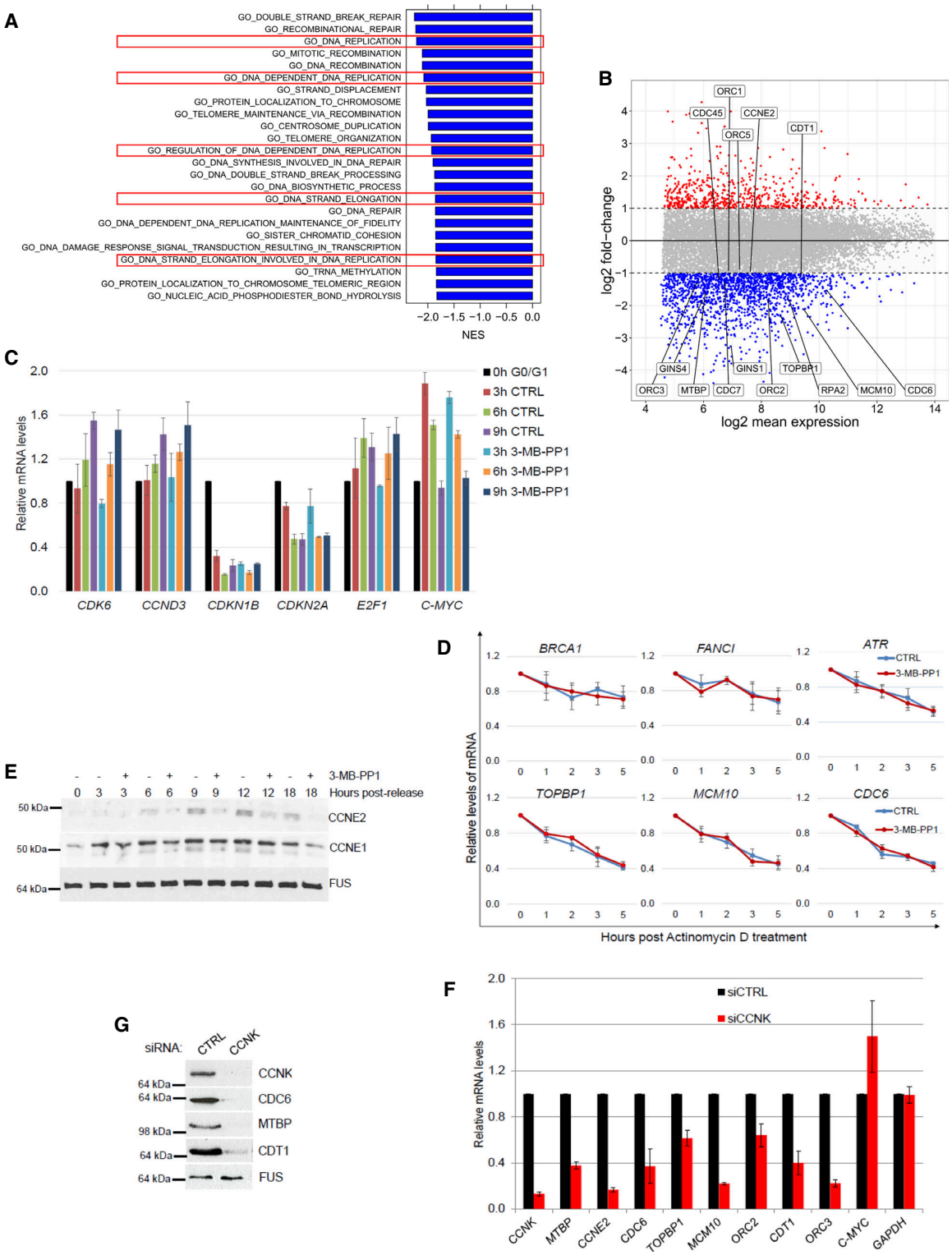

A
GO_DOUBLE_STRAND_BREAK_REPAIR
GO_RECOMBINATIONAL_REPAIR
GO_DNA_REPLICATION
GO_MITOTIC_RECOMBINATION
GO_DNA_RECOMBINATION
GO_DNA_DEPENDENT_DNA_REPLICATION
GO_STRAND_DISPLACEMENT
GO_PROTEIN_LOCALIZATION_TO_CHROMOSOME
GO_TELOMERE_MAINTENANCE_VIA_RECOMBINATION
GO_CENTROSOME_DUPLICATION
GO_TELOMERE_ORGANIZATION
GO_REGULATION_OF_DNA_DEPENDENT_DNA_REPLICATION
GO_DNA_SYNTHESIS_INVOLVED_IN_DNA_REPAIR
GO_DNA_DOUBLE_STRAND_BREAK_PROCESSING
GO_DNA_BIOSYNTHETIC_PROCESS
GO_DNA_STRAND_ELONGATION
GO_DNA_REPAIR
GO_DNA_DEPENDENT_DNA_REPLICATION_MAINTENANCE_OF_FIDELITY
GO_SISTER_CHROMATID_COHESION
GO_DNA_DAMAGE_RESPONSE_SIGNAL_TRANSDUCTION_RESULTING_IN_TRANSCRIPTION
GO_DNA_STRAND_ELONGATION_INVOLVED_IN_DNA_REPLICATION
GO_TRNA_METHYLATION
GO_PROTEIN_LOCALIZATION_TO_CHROMOSOME_TELOMERIC_REGION
GO_NUCLEIC_ACID_PHOSPHODIESTER_BOND_HYDROLYSIS
−2.0 −1.5 −1.0 −0.5 0.0
NES
B
ORC1
CDC45
CCNE2
ORC5
CDT1
GINS4
GINS1
TOPBP1
ORC3
MTBP
CDC7
ORC2
RPA2
MCM10
CDC6
log2 fold−change
log2 mean expression
C
Relative mRNA levels
0h G0/G1
3h CTRL
6h CTRL
9h CTRL
3h 3-MB-PP1
6h 3-MB-PP1
9h 3-MB-PP1
CDK6
CCND3
CDKN1B
CDKN2A
E2F1
C-MYC
D
BRCA1
FANCI
ATR
TOPBP1
MCM10
CDC6
CTRL
3-MB-PP1
Relative levels of mRNA
Hours post Actinomycin D treatment
E
3-MB-PP1
Hours post-release
50 kDa
CCNE2
50 kDa
CCNE1
64 kDa
FUS
F
siCTRL
siCCNK
Relative mRNA levels
CCNK
MTBP
CCNE2
CDC6
TOPBP1
MCM10
ORC2
CDT1
ORC3
C-MYC
GAPDH
G
siRNA:
CTRL
CCNK
64 kDa
CCNK
64 kDa
CDC6
98 kDa
MTBP
64 kDa
CDT1
64 kDa
FUS

Figure EV3.

Figure EV4. CDK12 directs expression of replication and DNA damage response genes downstream of the E2F/RB pathway.

A CDK12 directs expression of DNA replication genes downstream of the E2F/RB pathway. Graphs present ChIP-qPCR data for E2F1 and E2F3 in AS CDK12 HCT116 cells either treated or not with 3-MB-PP1 for 4 h. qPCR primers were designed at promoters of indicated genes. *n* = 3 replicates; error bars represent SEM. Ir is intergenic region; noAb corresponds to no antibody immunoprecipitation control.

B CDK12 inhibition does not lead to differential recruitment of RNAPII to E2F target genes. The plots show log2 fold-changes of RNAPII occupancy on promoters of E2F target genes (*y*-axis) plotted against corresponding log2 fold-changes in mRNA expression from nuclear RNA-seq (*x*-axis). Promoter occupancy was quantified as read counts in the ±3 kb regions around the transcription start site (TSS). For each gene, we selected the transcript with the most read counts in the RNAPII ChIP-seq samples (normalized to library size) in the ±3 kb regions around the TSS and transcription termination site (TTS). Corresponding RNAPII ChIP-seq and nuclear RNA-seq experiments are presented in Fig 5A and B. E2F target genes were obtained from Bracken *et al* [40]; rho = Spearman rank correlation coefficient.

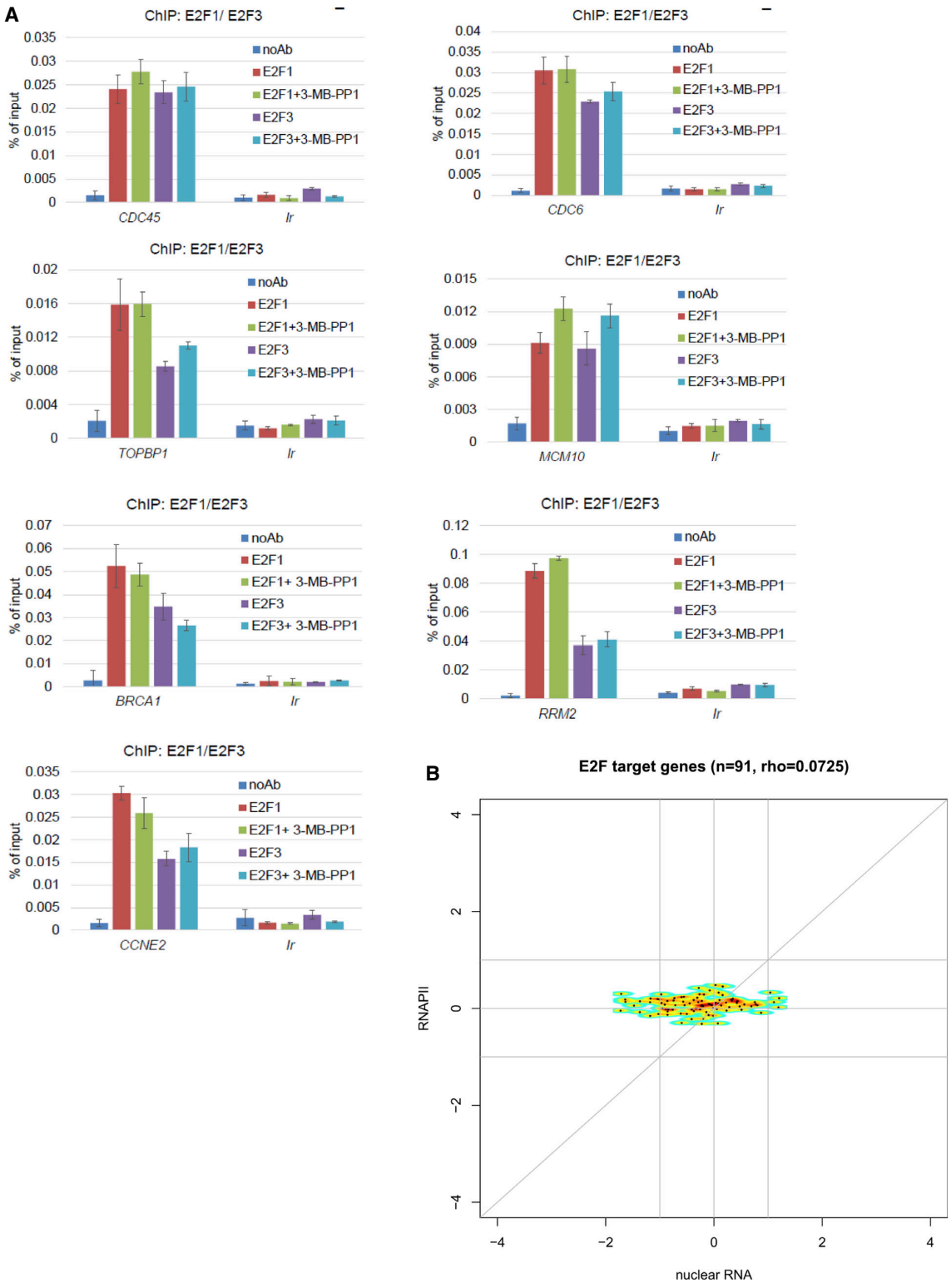

A
ChIP: E2F1/ E2F3
% of input
noAb
E2F1
E2F1+3-MB-PP1
E2F3
E2F3+3-MB-PP1
CDC45
Ir
ChIP: E2F1/E2F3
CDC6
TOPBP1
MCM10
BRCA1
RRM2
CCNE2
E2F1+ 3-MB-PP1
E2F3+ 3-MB-PP1
B
E2F target genes (n=91, rho=0.0725)
RNAPII
nuclear RNA

Figure EV4.

Figure EV5. Inhibition of CDK12 leads to diminished RNAPII processivity on down-regulated genes.

A High correlation between gene expression changes in nuclear and 3′end RNA-seq data. Graph compares log2 fold-changes in nuclear and 3′end RNA-seq data determined with DESeq2. rho = Spearman rank correlation coefficient.

B Inhibition of CDK12 affects the expression of similar subsets of genes in nuclear and 3′end RNA-seq data. See Fig 5A for legend. Venn diagrams are shown for significantly down-regulated (log2 fold-change < 0, $P \leq 0.01$) and up-regulated (log2 fold-change > 0, $P \leq 0.01$) genes.

C P-Ser5 occupancy shows shifts after CDK12 inhibition. Metagene analysis of P-Ser5 ChIP-seq data as described in Fig 5B and C.

D SPT6 shows diminished relative occupancy at 3′ends of down-regulated genes upon CDK12 inhibition. Metagene analysis of SPT6 ChIP-seq data as described in Fig 5B and C.

E CDK12 inhibition does not affect SPT6/RNAPII association in cells. Western blot analyses of SPT6 and RNAPII interaction after 4-h treatment with the 3-MB-PP1 in AS CDK12 HCT116 cells. Representative image from three replicates is shown.

Source data are available online for this figure.

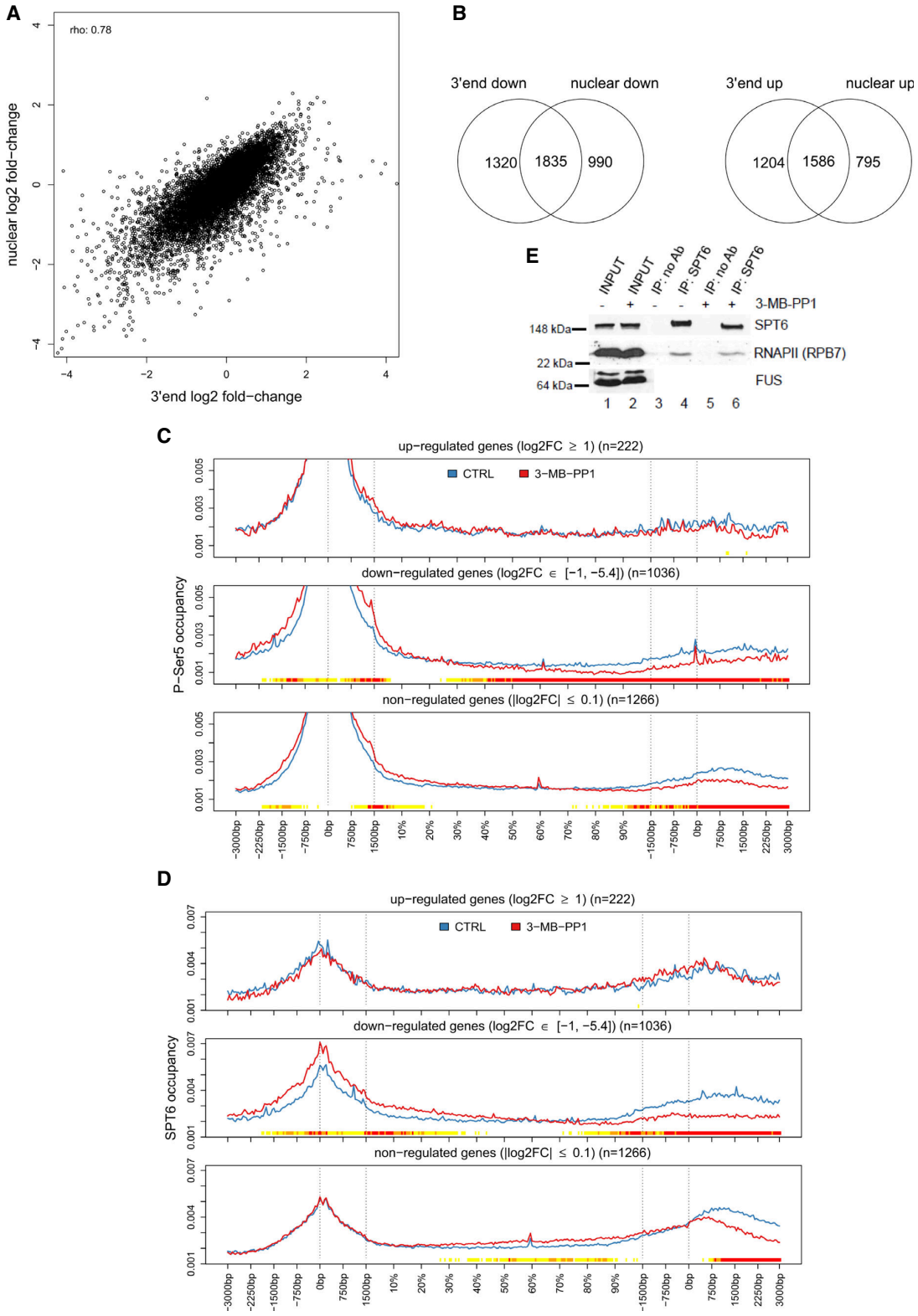

A
rho: 0.78
nuclear log2 fold−change
3'end log2 fold−change
B
3'end down
nuclear down
1320
1835
990
3'end up
nuclear up
1204
1586
795
E
INPUT
INPUT
IP: no Ab
IP: SPT6
IP: no Ab
IP: SPT6
3-MB-PP1
148 kDa
SPT6
RNAPII (RPB7)
22 kDa
64 kDa
FUS
1 2 3 4 5 6
C
up−regulated genes (log2FC ≥ 1) (n=222)
CTRL
3−MB−PP1
down−regulated genes (log2FC ∈ [−1, −5.4]) (n=1036)
non−regulated genes (|log2FC| ≤ 0.1) (n=1266)
P−Ser5 occupancy
D
up−regulated genes (log2FC ≥ 1) (n=222)
CTRL
3−MB−PP1
down−regulated genes (log2FC ∈ [−1, −5.4]) (n=1036)
non−regulated genes (|log2FC| ≤ 0.1) (n=1266)
SPT6 occupancy

Figure EV5.

CDK12 controls G1/S progression by regulating RNAPII processivity at core DNA replication genes

Anil Paul Chirackal Manavalan[1], Kveta Pilarova[1], Michael Kluge[2], Koen Bartholomeeusen[1,6], Michal Rajecky[1], Jan Oppelt[1], Prashant Khirsariya[4,5], Kamil Paruch[4,5], Lumir Krejci[3,5,7], Caroline C. Friedel[2], Dalibor Blazek[1,8]

TABLE OF CONTENTS:

Appendix Figure Legends

Appendix Figure S1. Preparation and characterization of AS CDK12 HCT116 cell line.

a, Sanger sequencing of WT CDK12 HCT116 clones.

b, Sanger sequencing of AS CDK12 HCT116 clones.

Appendix Figure S2. Genes that are more strongly down-regulated have a tendency towards more reduced occupancy of RNAPII at their 3'ends.

Metagene analysis of ChIP-seq occupancies of RNAPII as described in Fig. 5b, c on groups of genes with the indicated log2 fold-change of expression in nuclear RNA-seq.

Appendix Figure S3. P-Ser5 occupancy normalized to RNAPII shows no or very little changes across genes after CDK12 inhibition.

Metagene analysis as described in Fig. 5b, c of ChIP-seq P-Ser5 occupancies normalized to RNAPII.

Appendix Figures S4. The shift of P-Ser2 into the gene body is most pronounced in strongly down-regulated genes.

Metagene analysis as described in Fig. 5b, c of ChIP-seq P-Ser2 on groups of genes with the indicated log2 fold-change of expression in nuclear RNA-seq.

Appendix Figures S5. P-Ser2 occupancy normalized to RNAPII shows small but highly significant changes after CDK12 inhibition.

Metagene analysis as described in Fig. 5b, c of ChIP-seq P-Ser2 occupancies normalized to RNAPII.

Appendix Figure S6. SPT6 travels together with RNAPII on genes independently of CDK12 kinase activity.

Metagene analysis as described in Fig. 5b, c of ChIP-seq SPT6 profiles normalized to RNAPII.

Appendix Figure S7. Examples of genes whose transcription processivity and expression is dependent on the CDK12 kinase activity.

a, b, c, Examples of genes whose transcription processivity and expression is dependent on the CDK12 kinase activity. Nuclear RNA-seq data and RNAPII, P-Ser2, P-Ser5 and SPT6 ChIP-seq data for *SWSWAP* (**a**), *TOPBP1* (**b**) and *MCM10* (**c**) as described in Figs. 5d, e.

Appendix Figure S8. CDK12 inhibition results in transcript shortening of a subset of genes.

a, Differentially used down-regulated exons are predominantly present at gene 3´ends. Graph shows the distribution of relative exon positions as described in Fig. 6b of differentially used down-regulated exons (according to DEXSeq, log2 fold-change ≤-1, p≤0.01, n=3473 genes).

b, Differentially used up-regulated exons are predominantly present at gene 5´ends. Graph shows the distribution of relative exon positions as described in Fig. 6b of differentially used up-regulated exons (according to DEXSeq, log2 fold-change≥1, p≤0.01, n=2017 genes).

c, Inhibition of CDK12 kinase activity results in shorting of a subset of transcripts. Box plot shows the relative position of all exons (n=282614 exons) in comparison to exons identified as either up-regulated (log2 fold-change≥1, p≤0.01, n=2017 genes) or down-regulated (log2 fold-change≤-1, p≤0.01, n=3473 genes) by DEXSeq. n=3 replicates.

d, In down-regulated genes without a significantly differentially used exons, the exons close to 5´ and 3´ends also tend to be weakly up- and down-regulated, respectively. Box plots show the log2 fold-change in exon usage after CDK12 inhibition determined by DEXSeq for exons in genes without differentially used exons (p≥0.01 for all exons). Exons were grouped into deciles according to their relative exon position. n=3 replicates.

e, f, Genes with up- or down-regulated exons (at least one exon with log2 fold-change ≥1 or ≤-1, respectively, p≤0.01) show similar shifts of RNAPII and P-Ser2 occupancy in comparison to genes without differentially used exons (p≥0.01 for all exons). Metagene analyses of RNAPII (**e**) and P-Ser2 (**f**) ChIP-seq data as described in Fig. 5b, c. P-Ser2 occupancy is normalized to RNAPII occupancy.

g, Distribution of exon usage changes in genes not down-regulated but with significantly regulated exons shows a similar trend as for the down-regulated genes. Box plot shows log2 fold-changes in exon usage after CDK12 inhibition determined by DEXSeq for genes not down-regulated. Exons were grouped into deciles according to their relative exon position. n=3 replicates.

Appendix Figure S9. CDK12 kinase activity is required for optimal transcription of long genes.

a, b, Longer genes tend to have a larger fraction of differentially used exons. The same analysis as in the Fig. 7a using only down-regulated (**a**) or upregulated (**b**) exons, respectively. n=3 replicates.

Appendix Figures S10-S12. Longer genes show stronger changes in RNAPII, P-Ser2 and P-Ser5 ChIP-seq occupancy after CDK12 inhibition. Metagene analysis as described in Fig. 5b, c of RNAPII (S10), P-Ser2 (S11) and P-Ser5 (S12) ChIP-seq on groups of genes with the indicated length.

Appendix Figure S13 and S14. Genes with shortened transcripts have reduced RNAPII occupancy at 3′ ends and show a shift of the P-Ser2 signal towards gene bodies. Metagene analysis as described in Fig. 5b, c of RNAPII (S13) and P-Ser2 (S14) ChIP-seq occupancies on groups of genes with the indicated changes in their transcript length. Absolute Δ90% distance = 90% distance in control - 90% distance in CDK12-inhibited cells (positive values indicate shortening of transcripts in CDK12-inhibited cells). Relative Δ90% = absolute Δ90% divided by gene length.

Appendix Figure S15. CDK12 kinase activity is required for optimal transcription of long, poly(A)-signal-rich genes.

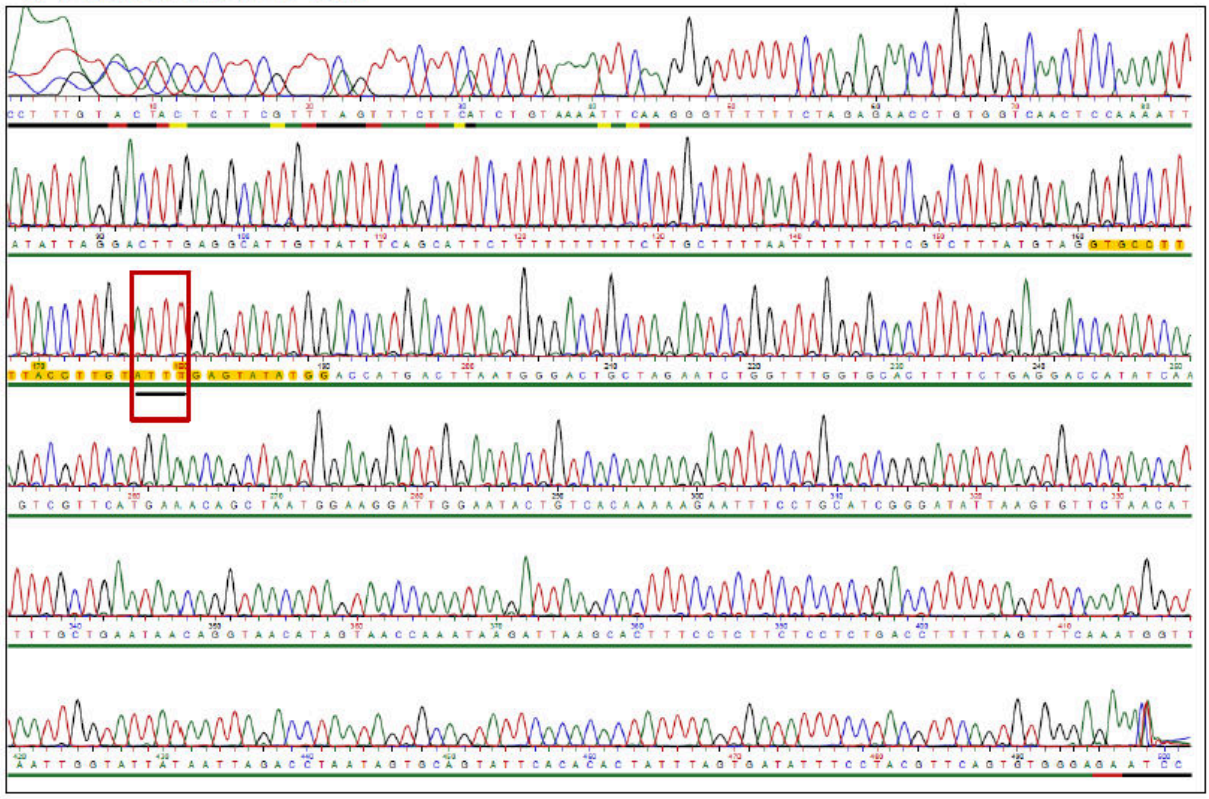
a, Gene length and abundance of canonical poly(A) signals are correlated. Length and number of canonical poly(A) signal sequences (AATAAA, ATTAAA) in protein coding genes is plotted against each other. rho=Spearman rank correlation coefficient.

b, Presence of poly(A) signals contributes to the shortening of transcripts. Box plots show the difference in the 10, 50 and 90% distance divided by gene length between control and CDK12 inhibited cells. Genes were grouped into quantiles according to the number of poly(A) signals (AATAAA, ATTAAA) per kilobase (kb) of gene length. n=3 replicates.

Appendix Fig. S1

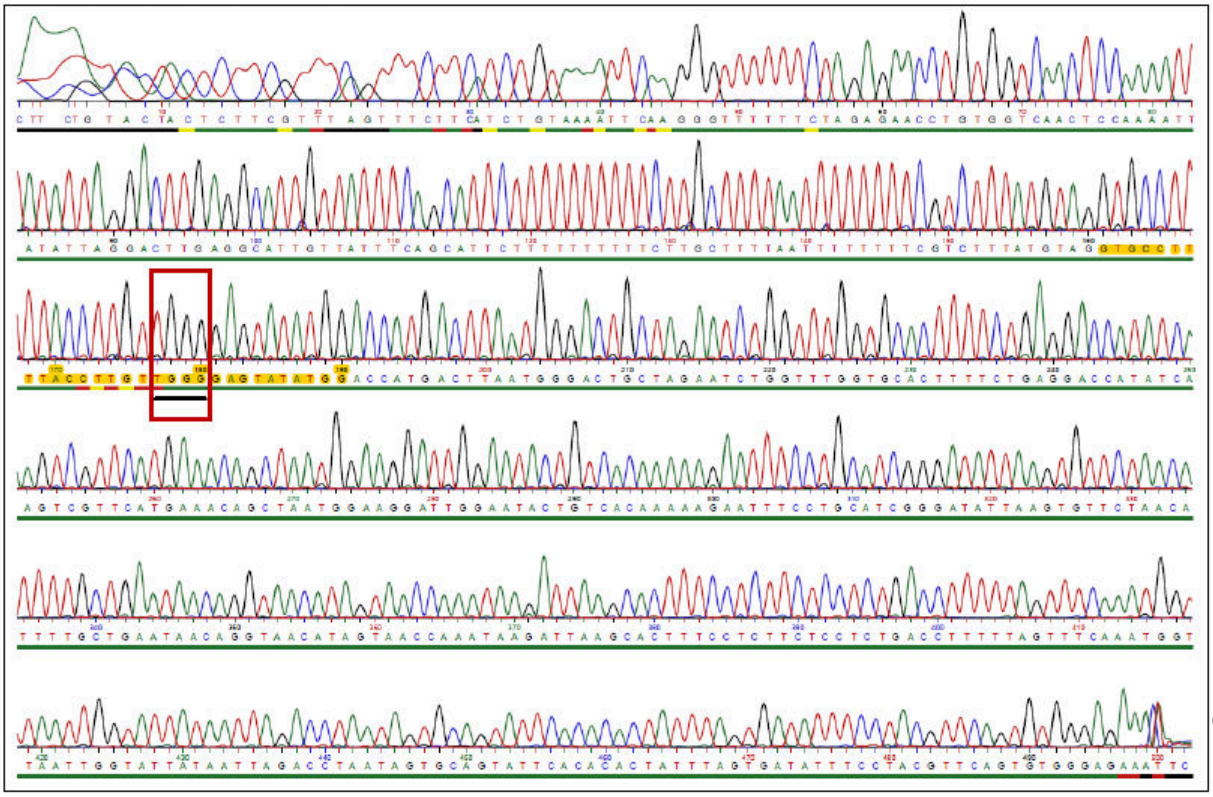
A

WT CDK12 HCT116 cells

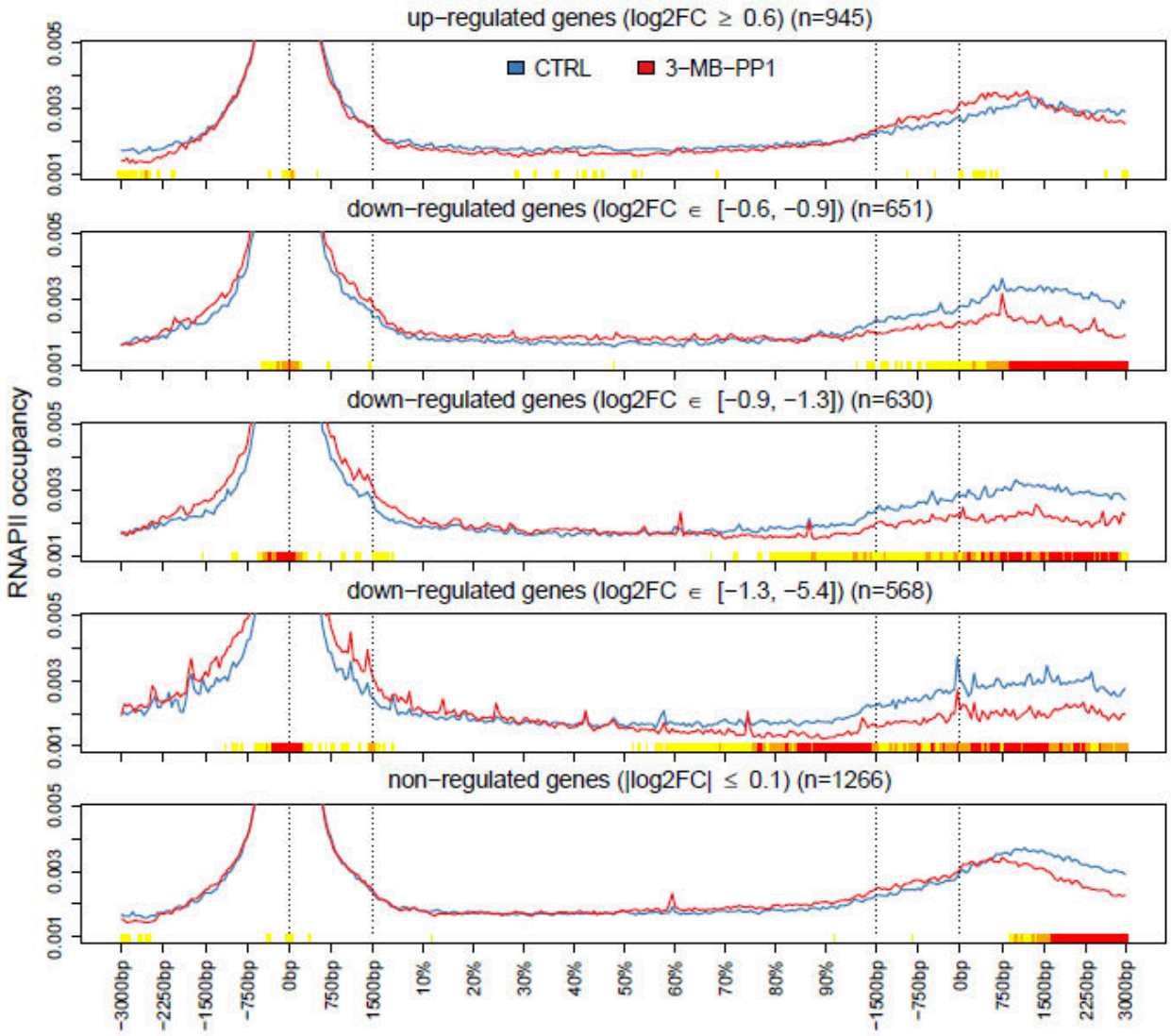


B

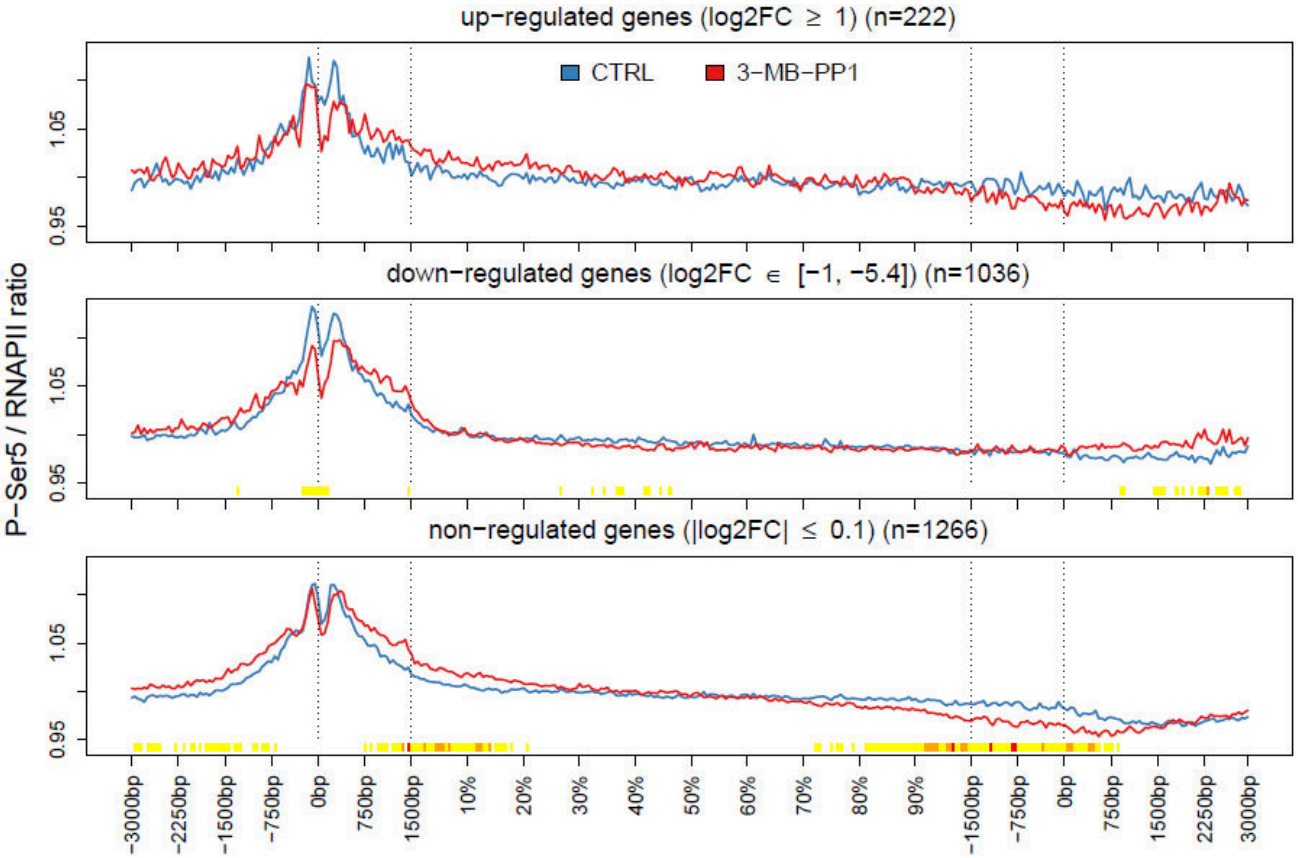
AS CDK12 HCT116 cells



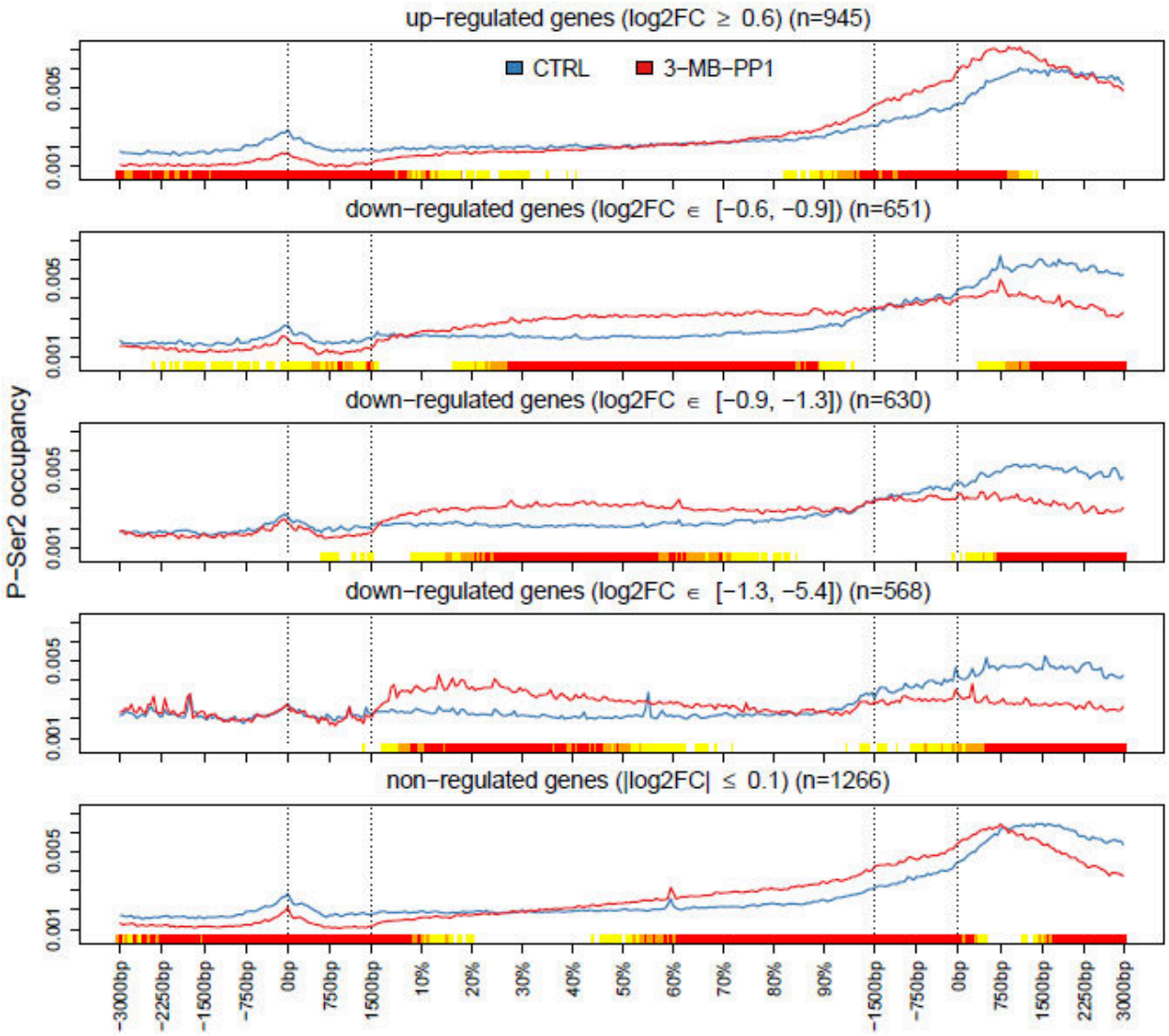
Appendix Fig. S2



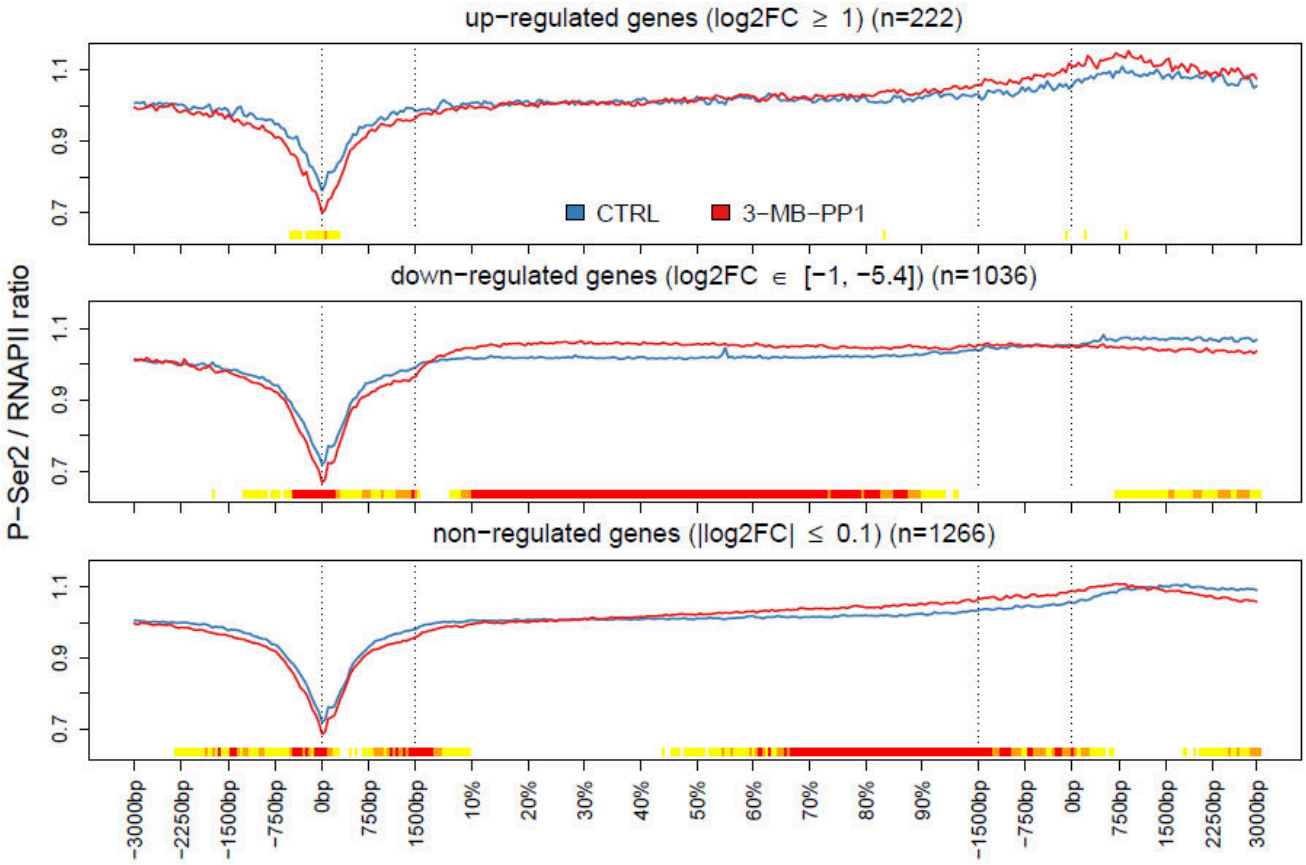
Appendix Fig. S3



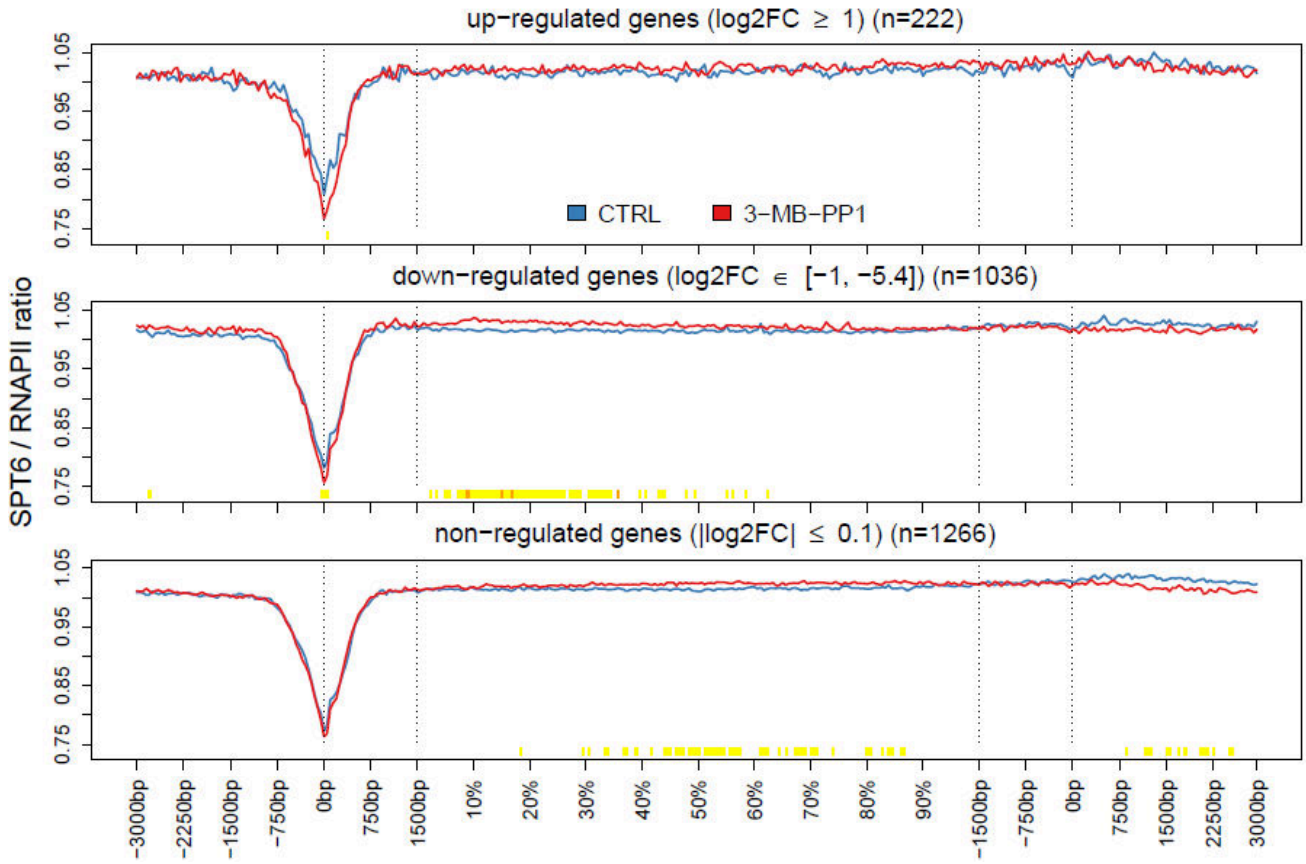
Appendix Fig. S4



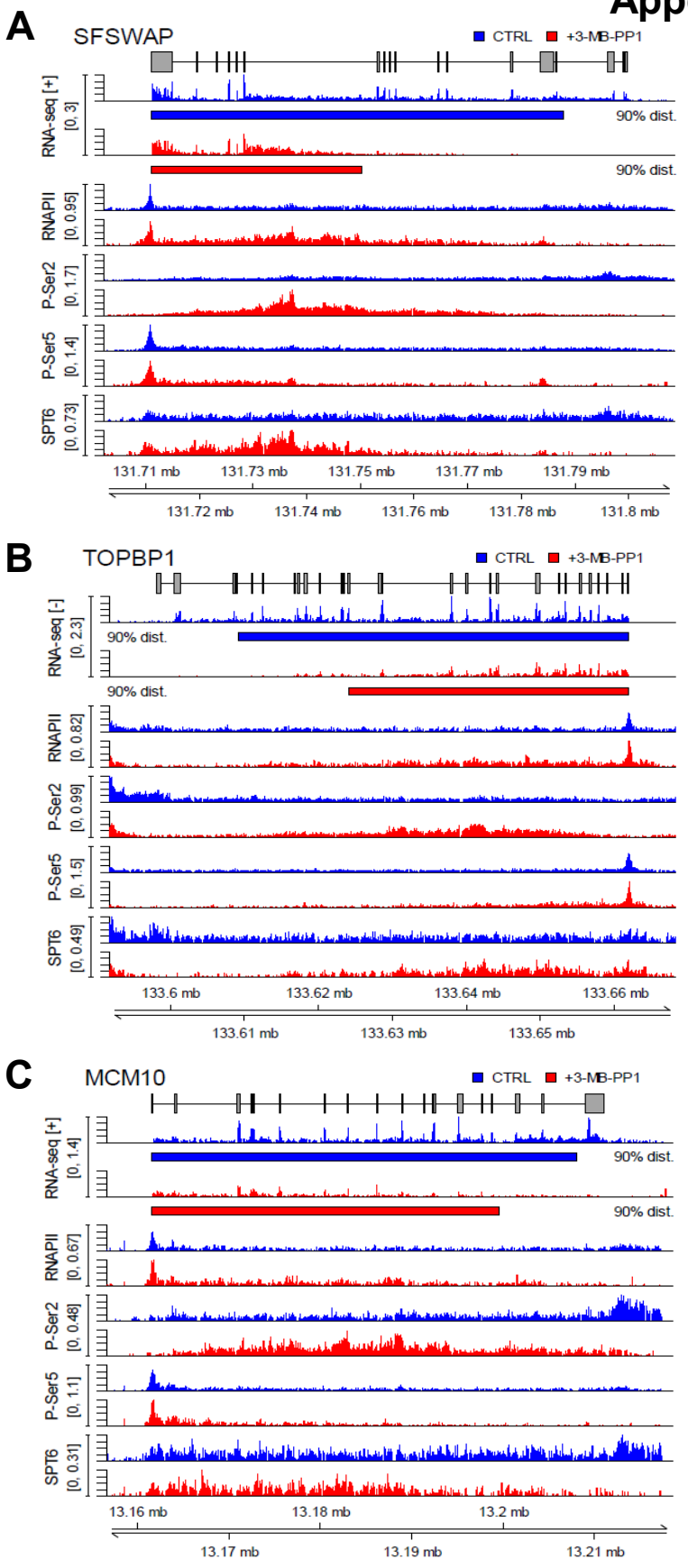
Appendix Fig. S5



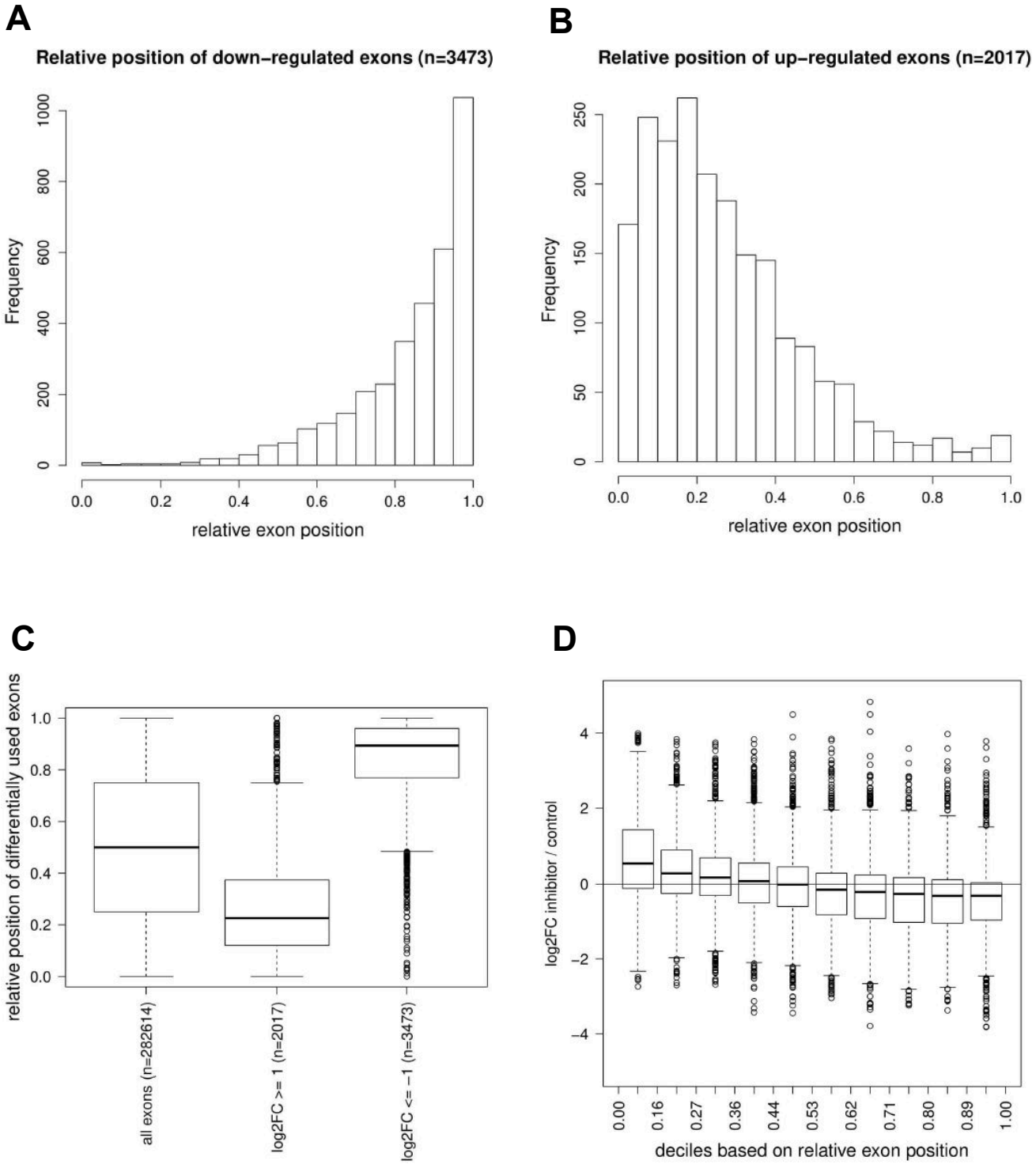
Appendix Fig. S6



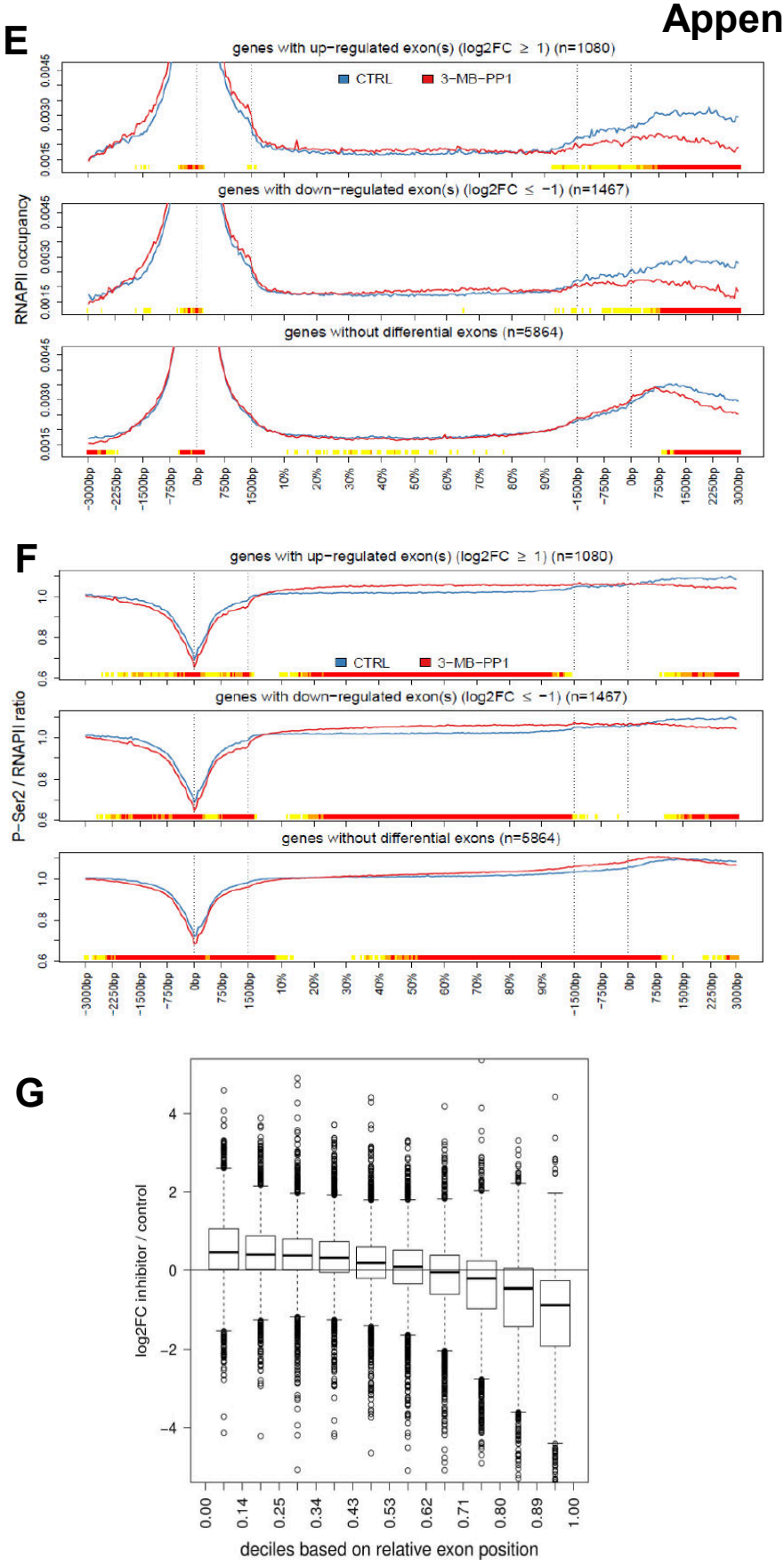
Appendix Fig. S7



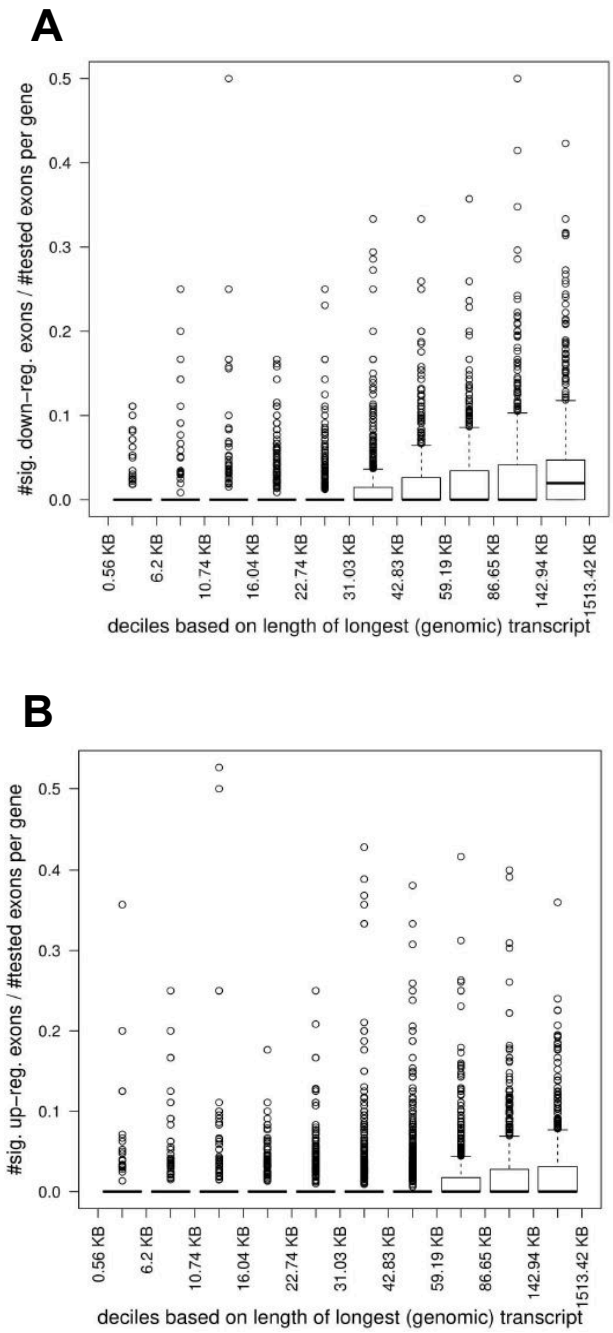
Appendix Fig. S8



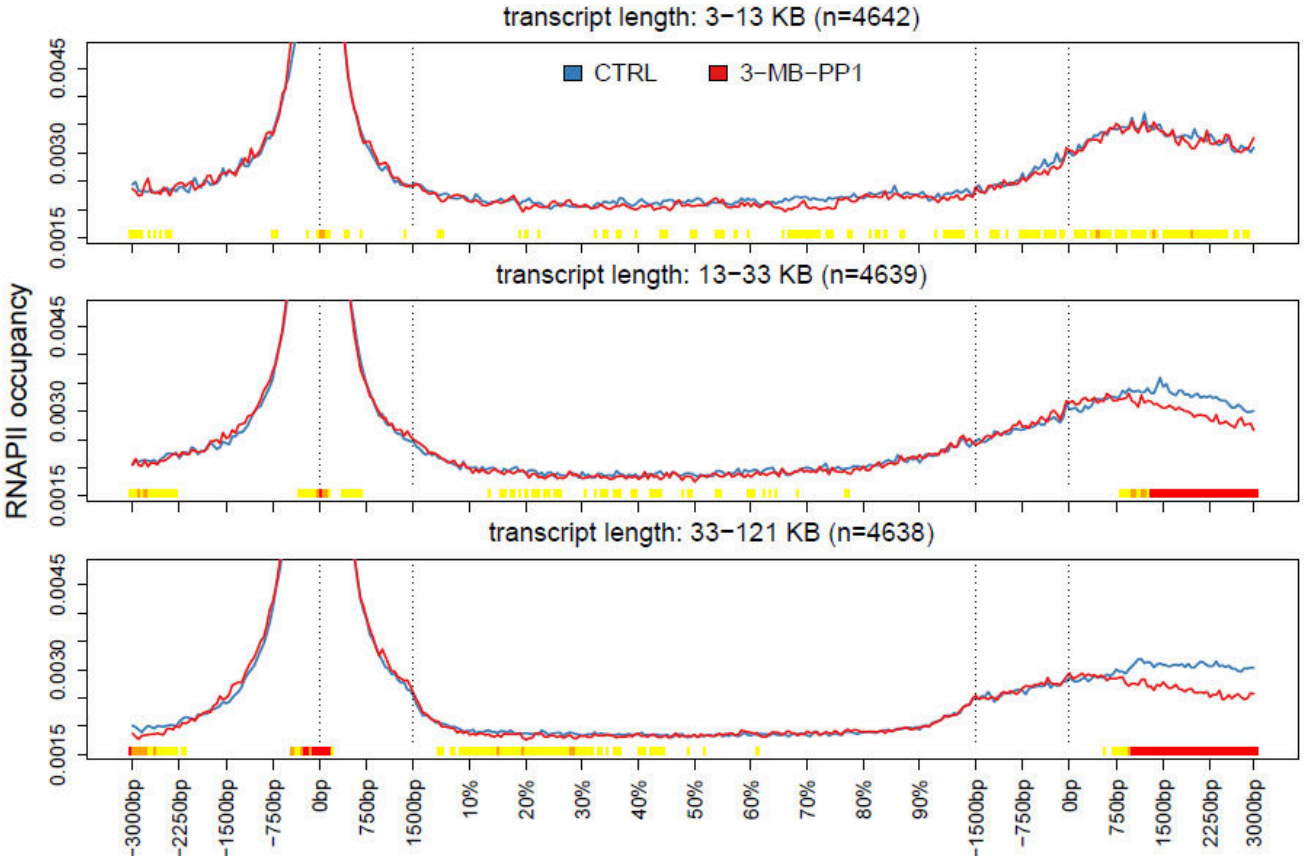
Appendix Fig. S8



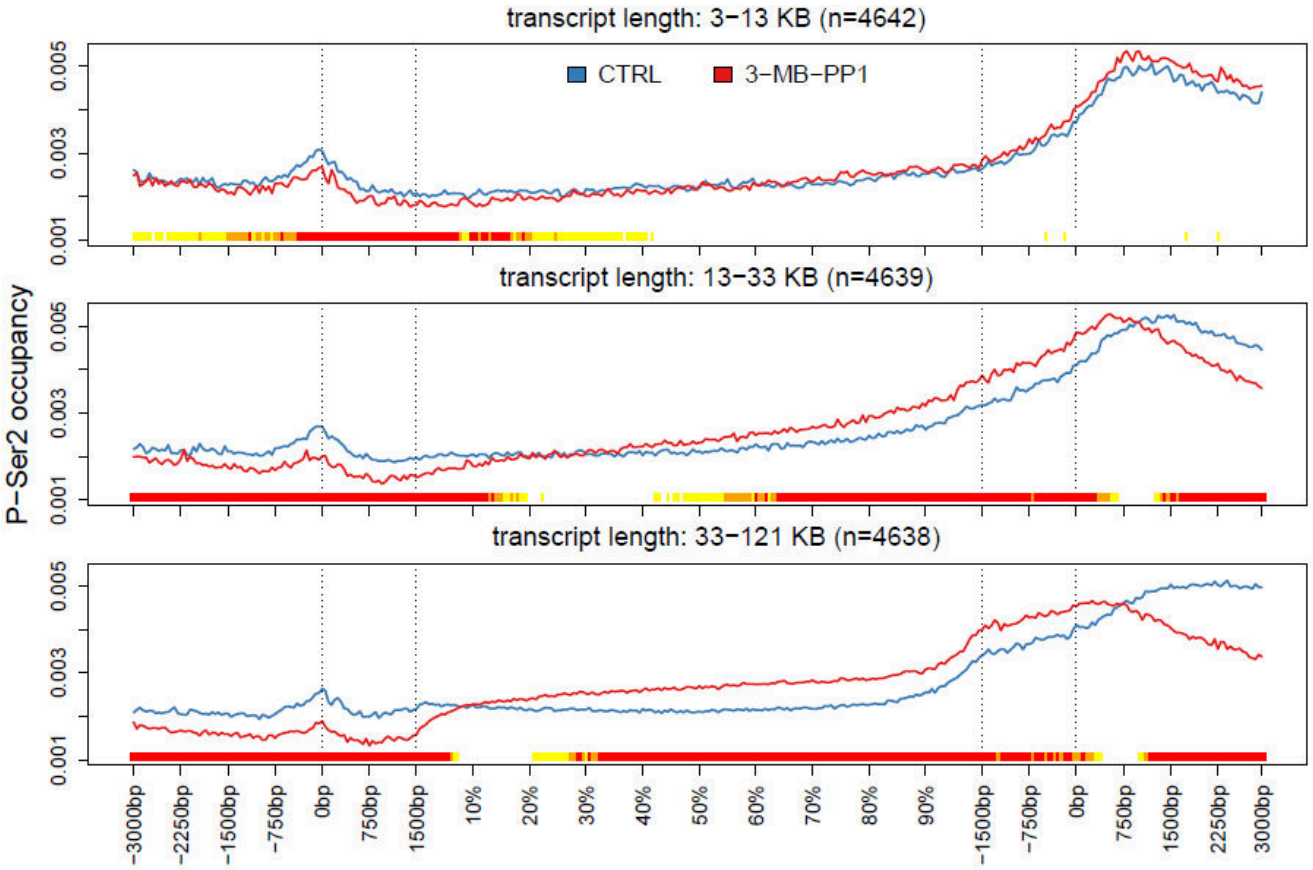
Appendix Fig. S9



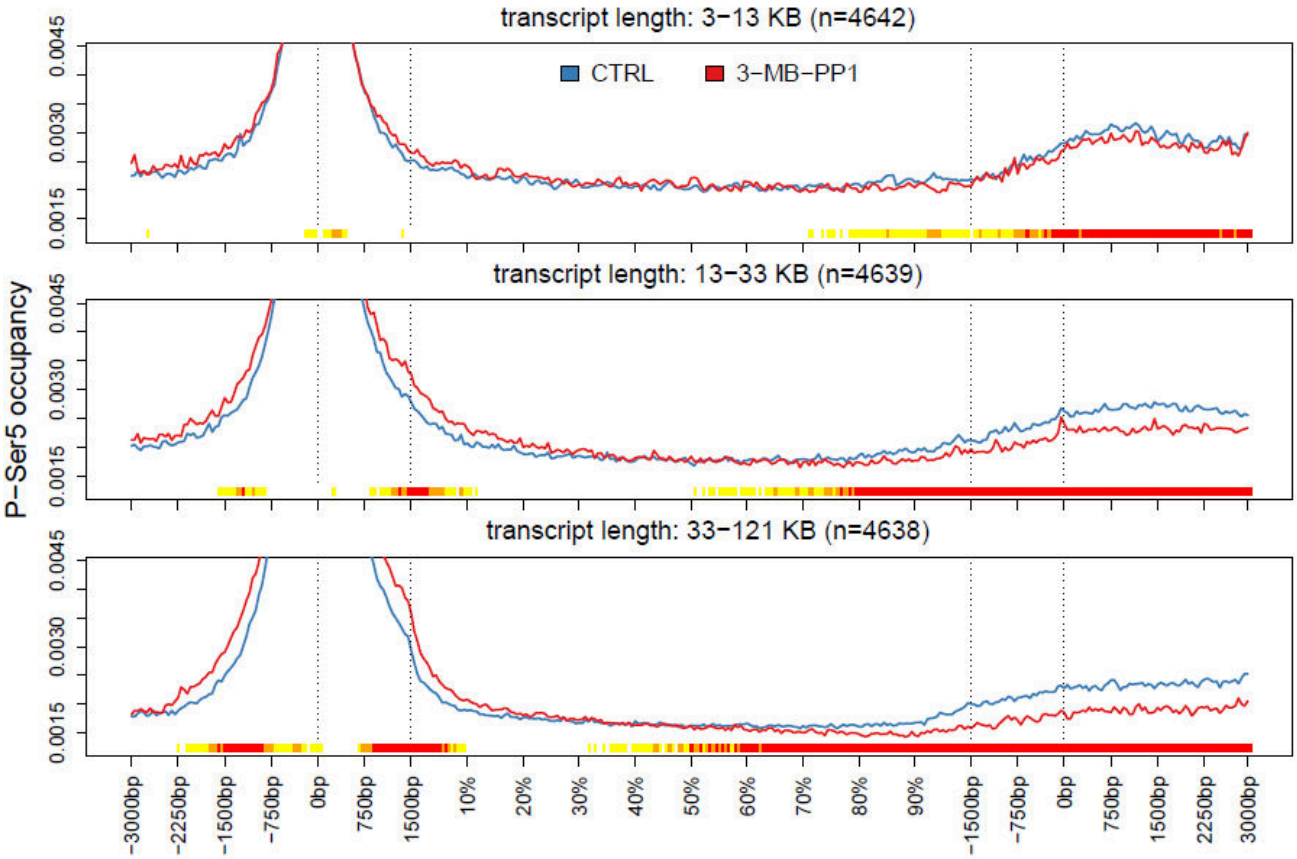
Appendix Fig. S10



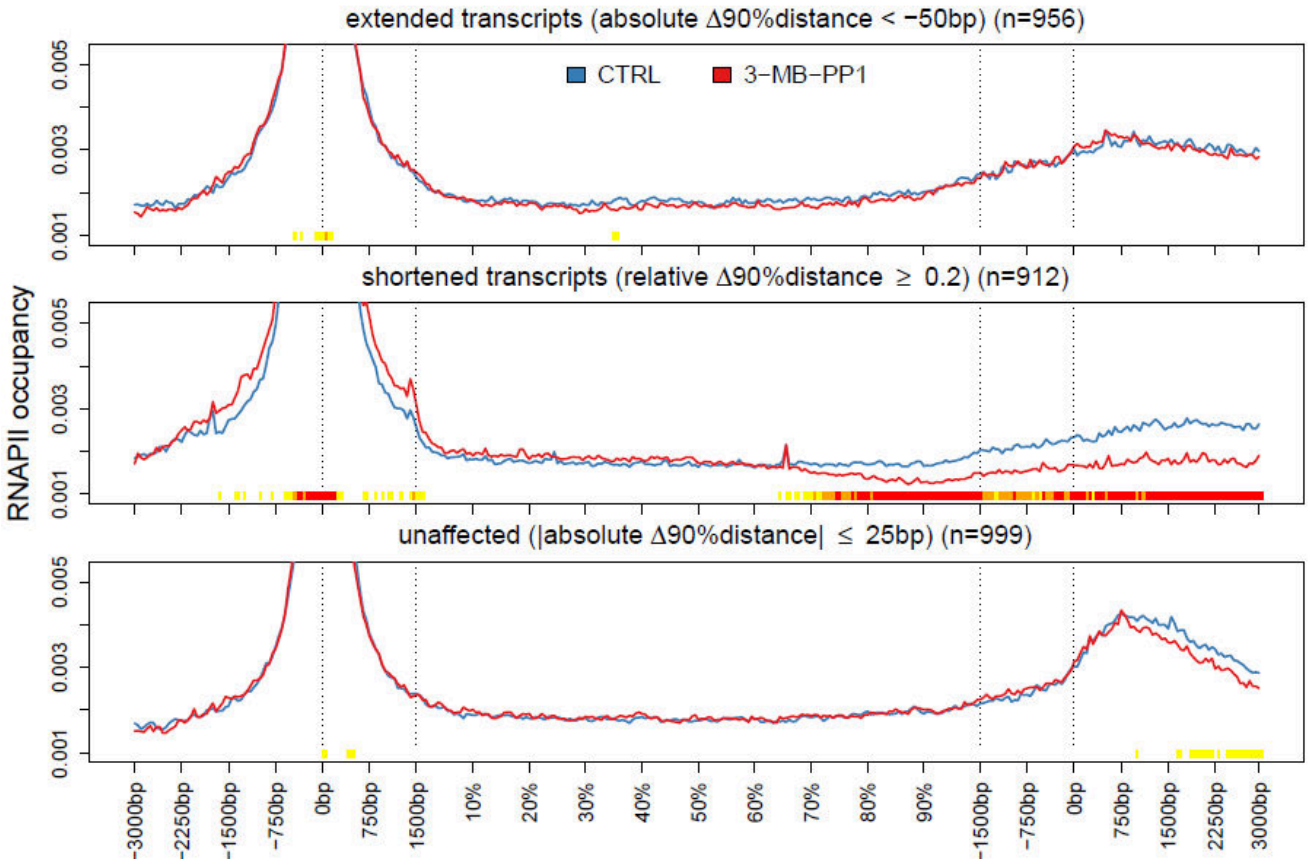
Appendix Fig. S11



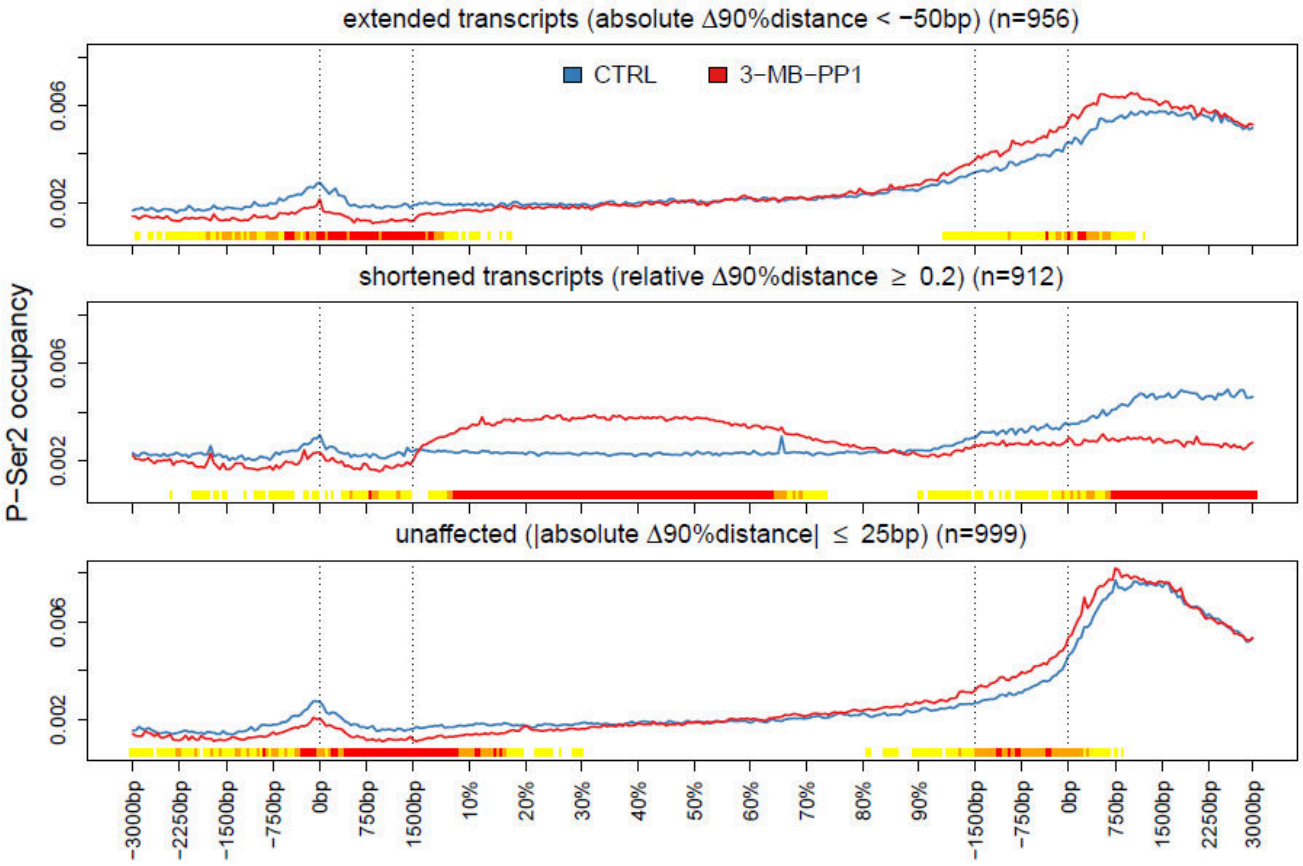
Appendix Fig. S12



Appendix Fig. S13



Appendix Fig. S14



Appendix Fig. S15