circtools — a one-stop software solution for circular RNA research

Supplementary material

Tobias Jakobi*[1,2], Alexey Uvarovskii†[1,2], and Christoph Dieterich‡[1,2]

[1]Department of Cardiology, Angiology, and Pneumology, University Hospital Heidelberg, Heidelberg, Germany
[2]DZHK (German Centre for Cardiovascular Research), Partner Site Heidelberg/Mannheim, Heidelberg, Germany

Download and source code via http://circ.tools
Online documentation available via http://docs.circ.tools
Latest version of this document via http://supplement.circ.tools

Contents

*Tobias.Jakobi@med.Uni-Heidelberg.DE
†Alexey.Uvarovskii@med.Uni-Heidelberg.DE
‡Christoph.Dieterich@med.Uni-Heidelberg.DE

Introduction

These pages contains supplementary information (i.e. commands, figures and tables) supporting our main manuscript regarding `circtools` — a one-stop software solution for circular RNA research. Within this document the actual command line instructions for the seven available modules (listed in Table 1) are documented in the Methods section. The Results sections sheds some more light on the results obtainable by using `circtools` and gives example on how to interpret them.

Module	Description
`detect`	Detection of circRNAs from mapped reads (Cheng *et al.*, 2016)
`quickcheck`	Overview of circRNA counts after mapping & detection
`circtest`	Statistical testing, e.g. host gene-independent expression (Cheng *et al.*, 2016)
`reconstruct`	Reconstruction of circRNA structures (Metge *et al.*, 2017)
`exon`	Identification of differential exon RNase R resistance
`enrich`	Test for significantly enriched features in circRNAs
`primex`	Design of circRNA specific qRT-PCR primer pairs

Table 1: Overview of available `circtools` modules and their respective functionality.

Source code, example data, and documentation

The complete, comprehensive, and regularly updated documentation of `circtools` can always be found under http://docs.circ.tools. The program's source code and the issue tracking system is hosted on the GitHub platform and can be reached via http://circ.tools. The example data used throughout this document are available via http://data.circ.tools. If you encounter any problems using `circtools`, please let us know at http://support.circ.tools or contact us via circtools@dieterichlab.org

Installation

circtools is written in Python2 (>= 2.7, `detect` and `reconstruct` module) and Python3 (>=3.4, all other modules). The tool has a number of external dependencies, mostly standard bioinformatics tools and packages. The installation will, by default, try to install all required dependencies.

Installation is performed via python3 setup.py install. No sudo access is required if the installation is suffixed with `--user` which will install the package in a user-writeable folder. In this case, the binaries should be installed to `/home/$USER/.local/bin/` (for Debian-based systems).

Supported operating systems

`circtools` was developed and tested on Debian Jessie 8 64 Bit and Debian Stretch 9 64 bit. macOS support is currently (09/2018) being tested and is already available in the `mac-dev` branch of the github repository (however, the macOS functionality cannot be fully guaranteed yet).

Installation from PyPi (preferred)

The default installation will install everything needed to run circtools *except R, STAR, or Stringtie* (see below). If you like you may install circtools locally (first call) or globally (second call, SU required).

```
pip3 install circtools --user # does not require root access

pip3 install circtools # will require root access
```

Installation from GitHub

The GitHub installation will install the most recent version directly from the source repository. Use this method if you want the latest fixes and features.

```
git clone https://github.com/dieterich-lab/circtools.git
cd circtools
pip3 install . --verbose --user
```

Updating circtools

You may want to update the circtools package if new versions are published. Like for the initial installation there are two ways to update circtools:

```
pip3 install circtools --user --upgrade

cd /path/to/circtools/repo/
git pull
pip3 install . install --verbose --user --upgrade
```

Required dependencies

External tools

- bedtools [>= 2.27.1] required by the enrichment module
- R [>= 3.3] required by visualization scripts and the primer design module
- STAR [>= 2.6.0] required by the `detect` and `reconstruct` module to map RNA-seq reads against a reference genome and detect back splice junctions
- Stringtie [>= 1.3.3b, optional] required by the `exon` module to carry out exon level analyses.

The installation procedure will automatically install two additional Python-based dependencies: DCC and FUCHS by temporarily cloning the repositories and installing both tools via setuptools to `/home/$USER/.local/bin/`. Both tools **require Python 2** in order to run.

The primer design module as well as the exon analysis and circRNA testing module require a working installation of R with BioConductor. All R packages required are automatically installed during the setup.

> **important**
>
> The setup scripts assumes that the folder for R plugins is writeable (either in the user's home or the system folder).

Python packages

- **For circRNA detection**
  - HTSeq>=0.11.0
  - pysam>=0.13.0
  - numpy>=1.8.2
  - pandas>=0.18.1

- **For circRNA reconstruction**
  - HTSeq>=0.11.0
  - pysam>=0.13.0
  - numpy>=1.8.2
  - pathos>=0.2.1
- **For circRNA enrichment**
  - pybedtools>=0.7.10
  - statsmodels>=0.8.0
- **For circRNA primer design**
  - BioPython>=1.71

Detailed installation

Getting the source code

Step 1: Clone source code from GitHub:

```
git clone https://github.com/dieterich-lab/circtools.git
```

Installation

Step 2: Install circtools using the provided installation script. The `--user` flag installs circtools in your home folder, thus making sure you do not require any administrative rights during the installation:

```
cd circtools
pip3 install . install --verbose --user
```

R environment

Step 3: Setting up R environment. In order for the automatic installation of R packages to work we need to set the package directory to a user-writeable path. The setup automatically sets that path to `/home/$USER/.R/`.

Dependencies

Step 4: The setup script is designed to make sure that the environment is setup correctly to run circtools. The circtools setup will automatically install CircTest, primex, DCC and FUCHS.

Finishing up

Step 5: Adding installation folder to `$PATH`. In order for circtools to find all executables, the setup will add the folder `/home/$USER/.local/bin/` automatically to your `.bashrc` file

This closes the circtools installation. To verify that circtools has been correctly installed, try to call circtools for the first time:

```
$> circtools --help
usage: circtools [-V]<command>
[<args>]
```

Use case 1: circRNAs in the murine heart

The first use case in this supplementary document is located online at http://docs.circ.tools in order to provide a continously maintained and updated sample data set that can be used to test and verify the installation of `circtools`. The online documentation contains chapters for each of `circtools`' modules (click on the module name to follow the link to the online documentation):

- detect
- reconstruct

- quickcheck
- circtest
- enrichment
- exon
- primex

In the tutorial, we use the data set from Jakobi *et al.* (2016) as an example. The data are paired-end, stranded RiboMinus RNA-seq data from Mus musculus, consisting of samples of four ages (2, 3, 6, and 12 month) collected from the whole hearts. The data can be downloaded from the NCBI SRA (accession number SRP071584). While the circtools suite does not offer specific module for the initial data processing, the short tutorial should give the user an idea on how to get the sequencing data in shape for the main `circtools` pipeline.

Use case 2: K562 and HepG2 cell lines

Data

All analyses within for this work have been carried out using total RNA from two cell lines, K562 and HepG2. K562 is a human immortalized leukemia cell line, known for its proteomic resemblance to undifferentiated granulocytes and erythrocytes. HepG2 is a human liver cancer cell line that secretes major plasma proteins and is used as a model to study intracellular trafficking and dynamics. Both cell line are also used within the ENCODE consortium (The ENCODE Project Consortium, 2012), therefore allowing to use additional data generated from these two cell lines.

rRNA depletion

Ribosomal RNA was depleted by using 20 µg of total RNA with the Ribo-Zero rRNA Removal Kit (Human/-Mouse/Rat, Epicentre, Madison, WI, United States) and 5 µg input material per reaction. The depletion was performed in parallel, afterwards samples were pooled.

Digestion of linear RNA

For digestion of linear RNA, $3/4$ of the depleted total RNA was treated with RNase R, whereas the remaining $1/4$ of the depleted RNA only received water and acts as a negative control. Both samples were treated equally for all further steps. After heating to 70 °C and cooling to 35 °C, 10× RNase R reaction buffer and RNase R (Epicentre; Cat No. RNR07250) were added to the 3-quarter fraction, followed by heating to 37 °C for 40 min.

Sequencing library preparation

Samples were cleaned up using Agencourt RNAClean XP beads (Beckman Coulter GmbH, Krefeld, North Rhine-Westphalia, Germany). Subsequent library preparation was performed with ScriptSeq-v2 RNA-Seq Library Preparation Kit (Epicentre). PCR amplification was performed with 15 cycles, without any further size selection. Sequencing was carried out on an Illumina HiSeq2500 system for 150 cycles using paired-end mode. Three replicates per sample have been sequenced.

Preprocessing of raw sequencing files

The raw sequencing files were quality checked with FASTQC. Afterwards, residual sequencing adapter fragments and low quality bases were removed using Flexbar 3.0 (Roehr *et al.*, 2017).

```
flexbar −r mate1.gz −p mate2.gz −t output/  −n 15 −z GZ −m 30 −u 0  −q TAIL
−qt 28 −as "AGATCGGAAGAG" −qf sanger −j
```

Listing 1: Command line used for trimming residual adapter sequences and low quality regions using Flexbar.

Methods

Mapping of raw sequencing files

Mapping of the preprocessed sequencing data was carried out using STAR version 2.5.3a (Dobin *et al.*, 2013) as outlined in the circtools documentation in the chapter "Detection - Mapping of the fastq files with STAR".

circtools `detect`

For the `detect` module, a few files have to be prepared, namely the `bam_files.txt`, `samplesheet`, `mate1`, and `mate2`. Please refer to the section "Detection of circular RNAs from chimeric.out.junction files with circtools" of the main documentation for details on how to prepare these files. Briefly, the files contain the list of different input files for all of the samples.

```
circtools detect @samplesheet -ss -T 10 -Pi -mt1 @mate1 -mt2 @mate2 -D -an
GRCh38.85.gtf -A GRCh38_85.fa -B @bam_files.txt -R GRCh38_repeatmasker.gtf
-M -Nr 2 2 -fg -G -t temp -F -L 20
```

Listing 2: Command line used for the circRNA detection process using the `detect` module

circtools `quickcheck`

For the `quickcheck` module STAR log files are required since they contain detailed information regarding the mapping process. The path to the STAR mappings is supplied via `-s stars_mappings/`, the directory for the additionally required results of the `detect` module is specified via `-d detect/`. Furthermore the labeling, the order of replicates, and the columns to use from the of the circRNA detection data are provided.

```
circtools quickcheck -d detect/ -s stars_mappings/
-l HepG2_RNaseR-,HepG2_RNaseR+,K562_RNaseR-,K562_RNaseR+
-g 1,2,1,2,1,2,3,4,3,4,3,4 -o /results/ -R 16,17,18,19,20,21
```

Listing 3: Command line used for a quality check of the detection/mapping process with the `quickcheck` module

circtools `circtest`

The `circtest` module features input flags similar to the `quickcheck` module. The labeling, replicates, and the grouping of columns have to be provided additionally to the folder containing the `detect` results.

```
circtools circtest -d detect/
-l HepG2_RNaseR-,HepG2_RNaseR+,K562_RNaseR-,K562_RNaseR+
-c 4,5,6,7,8,9,10,11,12,13,14,15
-g 1,2,1,2,1,2,3,4,3,4,3,4 -n circtest -o results/ -m 2000
```

Listing 4: Command line used for a `circtest` module to identify significantly enriched circRNA in the RNase R-treated sample compared to the untreated samples

circtools `reconstruct`

The `reconstruct` module requires processed results from the `detect` module as well as output generated during read mapping. An overview of the required input can be found in the "Required input data" section of the module's main documentation.

```
circtools reconstruct -O results/ -r 2 -q 2 -p refseq -e 3 -T /tmp/
-D detect/CircRNACount -J 2047_$LIB_STARmapping.Chimeric.out.junction -F
2047_$LIB_mate1_STARmapping.Chimeric.out.junction -R
2047_$LIB_mate2_STARmappingChimeric.out.junction.fixed
-B $LIB_merged.bam -A hg38.RefSeq.exons.bed -N $LIB -P 40
```

Listing 5: Command line used for the `reconstruct` module to carry out the structurual analysis of the circRNAs. LIB is to be replaced with the library letter.

circtools exon

The exon module requires additional files generated with StringTie Pertea *et al.* (2015). The directory to these results is supplied to the circtools call.

```
stringtie 2047_$LIB_STARmapping/Aligned.noS.bam -v -f 0.2 -p 20
-G GRCh38.85.gtf -e -B
-o 2047_$LIB_STARmapping_StringTieBallgown/ballgown.gtf
-C 2047_$LIB_STARmapping_StringTieBallgown/reference_transcripts.gtf
```

Listing 6: StingTie command line used for generating the required count tables for the exon module.

The labeling, replicates, and the grouping of data columns have to be provided additionally to the folder containing the detect results and the path to the files containing the significantly enriched circRNAs from the circtest module.

```
circtools exon -d detect/ -r 1,1,2,2,3,3
-l HepG2_RNaseR-,HepG2_RNaseR+,K562_RNaseR-,K562_RNaseR+
-c 10,11,12,13,14,15 -g 1,2,1,2,1,2 -C enrichment_total.csv
-b stringtie_latest/ -G GRCh38.85.gtf -o results/
```

Listing 7: Command line used for the exon module to identify differentially spliced exons within the detected circRNAs

circtools enrich

The enrich module uses precomputed peaks from BED narrow peak files [1] to screen for circRNAs that might be enriched for peaks / binding sites of certain RNA binding proteins. The module uses the significant enriched circRNAs from the circtest as primary source and is run multiple times for each of the RBPs. For each RBP and cell line 2,000 random iterations were performed to be able to distinguish between a significant enrichment and random background.

```
circtools enrich -c enrichment_total.csv -b eclip/$RBP_$CELL.bed
-a GRCh38.85.gtf -g hg38.chrom.sizes -i 2000 -I exon -p 20 -P 1 -T 1
-o results/ -F $RBP_$CELL
-t /scratch/global_tmp/
```

Listing 8: Command line used for the enrich module to identify circRNAs that may be enriched for binding sites of RNA binding proteins. Coordinates are based on eCLIP peaks.

Since the computations were performed for each individual RBP and cell line the results have to be merged. A simple awk command can be used to combine all CSV output files into one final table.

```
awkt '{{print FILENAME"\t"$0}}' *2000*.csv |
sed 's/_2000_.*.csv//g' |
grep -v circRNA_host_gene  > final_table.csv
```

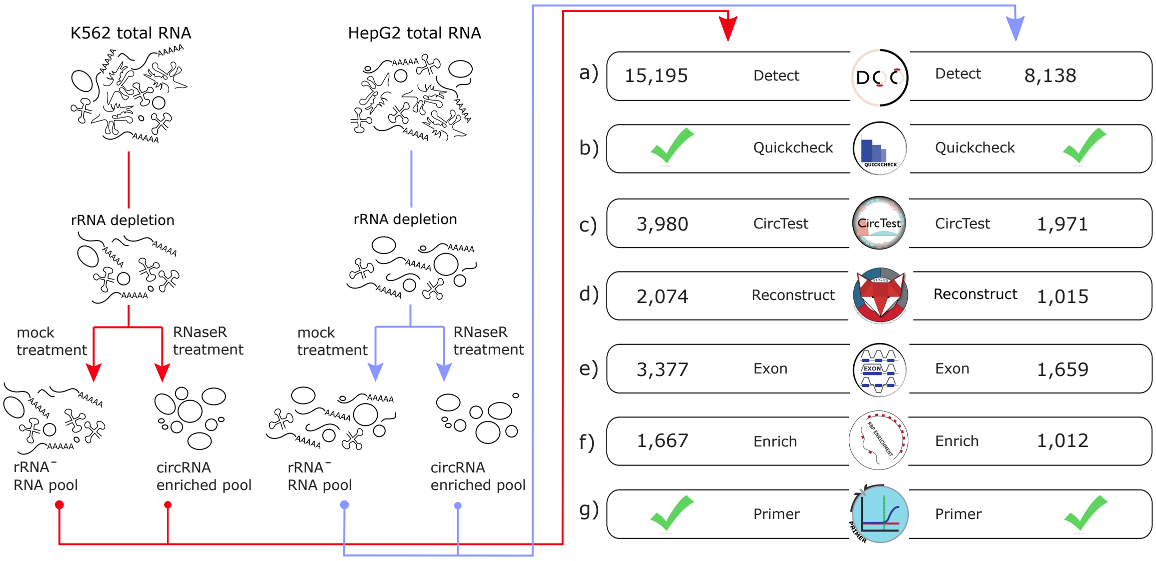
Listing 9: awk command line used to combine the CSV output files of all RNA binding proteins into the final results table.

circtools primex

In a final step, primer pairs were generated exemplary for the circRNA circN4BP2L2. As input, the results from the detect module are recommended, together with the exact ID of the circRNA of interest and the genome annotation used for a correct display of the circRNA.

```
circtools primex -d detect/CircCoordinates -f GRCh38_85.fa -g
GRCh38_85.gtf -O hs -o primer/ -i N4BP2L2_13_32517857_32527532_-
-T "N4BP2L2 primer"
```

Listing 10: Command line used for the enrich module to generate a set of primers for circRNA circN4BP2L2



Supplementary Figure 1: Visualization of an example circtools workflow with the seven modules. Input for the workflow are for two RNase R treated samples and two untreated samples (each from K562 and HepG2 total RNA). **a)** `detect` module: Shown are the number of detected BSJ per sample given the threshold of 2 replicates >= 2 BSJ. **b)** `quickcheck` module: the module is intended to allow a fast assessment of the quality of the library and the success of the read mapping phase. An example is shown in Supplementary Fig.2 a) and b). **c)** `circtest` module: Shown are the number of significantly enriched circRNAs by BSJ counts in the RNase R treated sample compared to the untreated sample given an FDR < 0.05. **d)** `reconstruct` module: shown are the numbers of completely reconstructed (i.e. coverage score $= 1.0$) circRNAs shared between all three replicates of each of the RNase R-treated samples. **e)** `exon` module: Shown are the numbers of exons enriched in the RNAse R-treated libraries compared to the untreated samples. **f)** `enrich` module: Number of circRNAs that show significant ($p < 0.05$) enrichment of RBP peaks in exons of the annotated circRNA region (linear $p > 0.05$, examples shown in Figure 4) **g)** `primex` module: the module allows for quick and easy design of primers specific to selected circRNAs. Example shown in Supplementary Fig.5.

Results

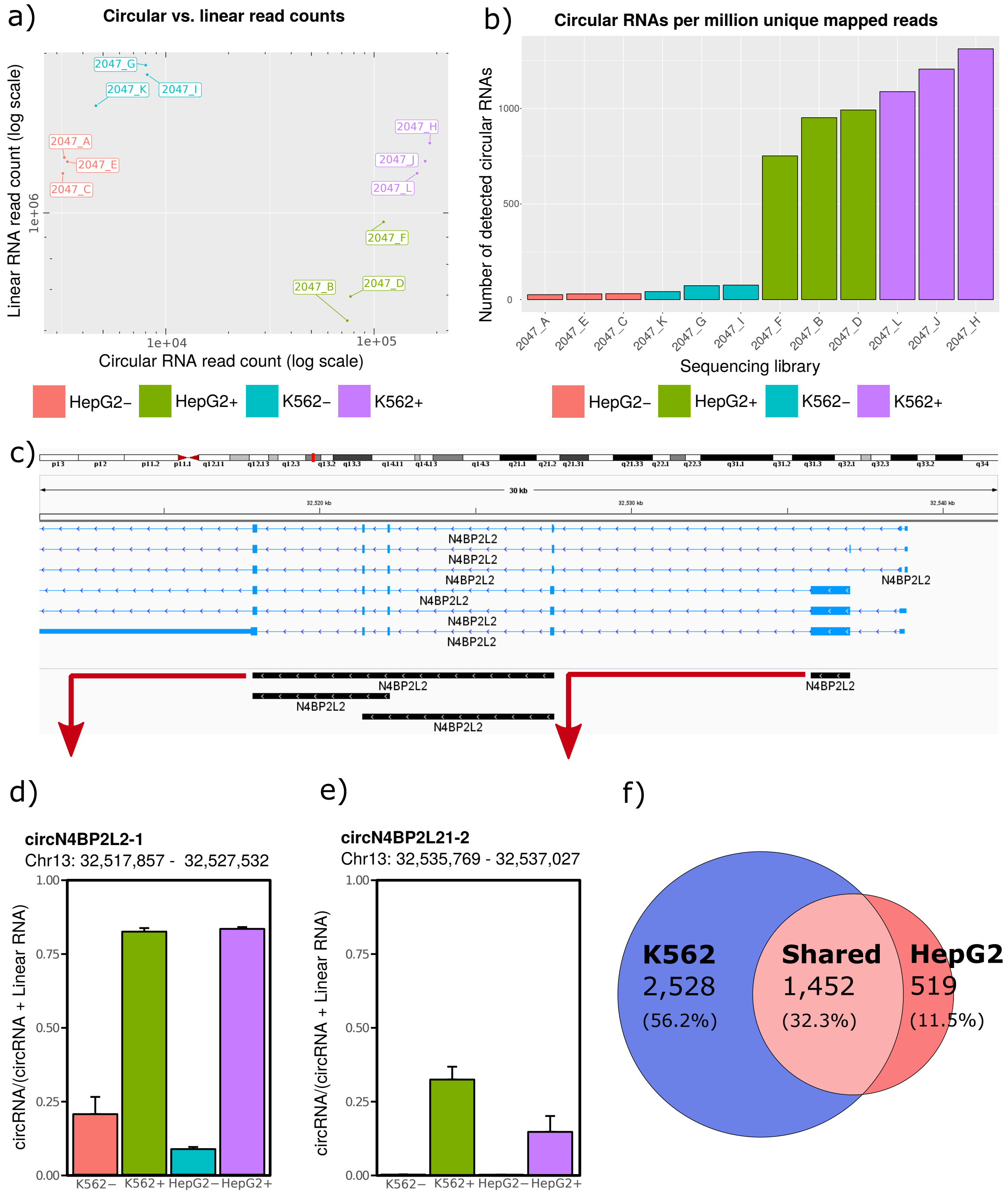
circtools `detect`

An overview of the complete project work flow is shown in Supplementary Fig.1. Using the “CircSeq” approach (Jeck *et al.*, 2013) RNA pools of two cell lines were sequenced. The initial circRNA prediction of the `detect` module yielded 20,118 circRNAs for both cell lines. After filtering this set for at least 2 reads per back splice junction (BSJ) in at least 2 replicates the final set of high-confidence predictions contains 15,195 circRNAs for K562 and 8,138 for the HepG2 cell line (Supplementary Fig.1 a). A direct comparison of the detected circRNAs shows $1/3$ of the circRNAs is shared between both cell lines. More than 50% of the detected circRNAs are specific to K562 while only sightly more than 10% are specific to the HepG2 cells (Supplementary Fig.2 f). All predicted circRNAs are stored in a BED-like format, therefore making it easy to directly visualize the predictions in any compatible visualization tool. e.g. IGV (Robinson *et al.* (2011), Supplementary Fig.2 c).

circtools `quickcheck`

The initial `quickcheck` results show a typical outcome for high quality CircSeq libraries. While libraries A, E, and E (HepG2), as well as K, G, and I (K562) show very high absolute counts for linear reads and also cluster closely together, the RNase R-treated libraries B, D, F (HepG2) and H, J, L (K562) are strongly enriched for circRNA read counts (Supplementary Fig.2 a). This is in accordance with the number of circRNAs detected per million uniquely mapped reads. Although the number of circRNA per million mapped reads is slightly variable between the three replicates, the overall trend of highly elevated circRNA counts in the RNase R-treated samples is clearly visible (Supplementary Fig.2 b). The circRNA read count metric also confirms the lower overall number of circRNAs detected for the HepG2 cell line since the absolute counts for K562 are higher for all three replicates than for any of the HepG2 samples. However, taken the results from the `quickcheck` module, all libraries passed the initial checks (Supplementary Fig.1 b).

[1] https://tinyurl.com/eclip-data



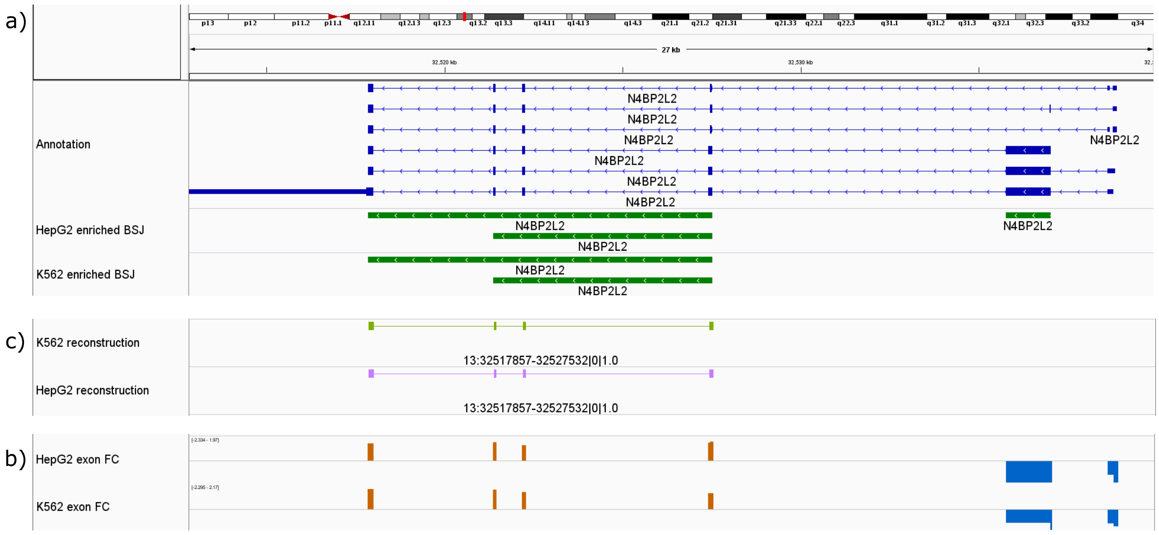
Supplementary Figure 2: Initial quality assessment of library setup, read mapping phase, and visualization of circRNA detection phase. **a)** Comparison of mapped reads for linear transcripts with those reads spanning a BSJ. **b)** Assessment of the library preparation employing the CircSeq approach using the `quickcheck` module. **c)** Graphical representation of circRNA detection results, shown are detected circRNAs isoforms originating from the N4BP2L2 host gene. **d)** Detailed view of the enrichment of circN4BP2L2-1 in the RNase R-treated samples. **e)** Detailed view of the enrichment of circN4BP2L2-2 in the RNase R-treated samples. **f)** Overlapping and distinct circRNAs detected in K562 and HepG2 samples.

circtools `circtest`

After quality checking, candidate circRNA sets were processed using the `circtest` module to obtain the number of significantly enriched circRNAs by BSJ counts in the RNase R-treated sample vs. the untreated sample. For the K562 cell line 3,980 circRNAs passed the test at $p < 0.05$, for HepG2 1,971 circRNAs were marked as significant (Supplementary Fig.1 c). In addition to these global counts of enriched circRNAs, the `circtest` module also generates a PDF file with a graphical representation of the relative enrichment for the top enriched candidates. Exemplary shown are circN4BP2L2-1 (Supplementary Fig.2 d) and circN4BP2L2-2 (Supplementary Fig.2 d). While both circRNAs originate from the same host gene, N4BP2L2, and are part of the set of high-confidence candidates, circN4BP2L2-1 shows a higher enrichment in both K562 and HepG2 compared to circN4BP2L2-2.

circtools `reconstruct`

Subsequently the results of the `detect` module were passed over to the `reconstruct` module in order to reconstruct internal circRNA structures. Using a cutoff of 100% coverage of all exons by reads, the internal structure of 2,074 and respectively 1,015 circRNAs was reconstructed for K562 and HepG2 (Supplementary Fig.1 d). The results are available on a genome-wide scale, thus giving the researcher an overview of all reconstructed circRNAs as well as on a per-circRNA level (see the "Output produced by circtools reconstruct" section of the module's main documentation for a detailed summary of all available output files). BED12-compatible output of the module can be used to visualize the reconstructed circRNAs with a suitable genome browser and allows also for a direct comparison to the initial prediction (Supplementary Fig.3 a) & b)



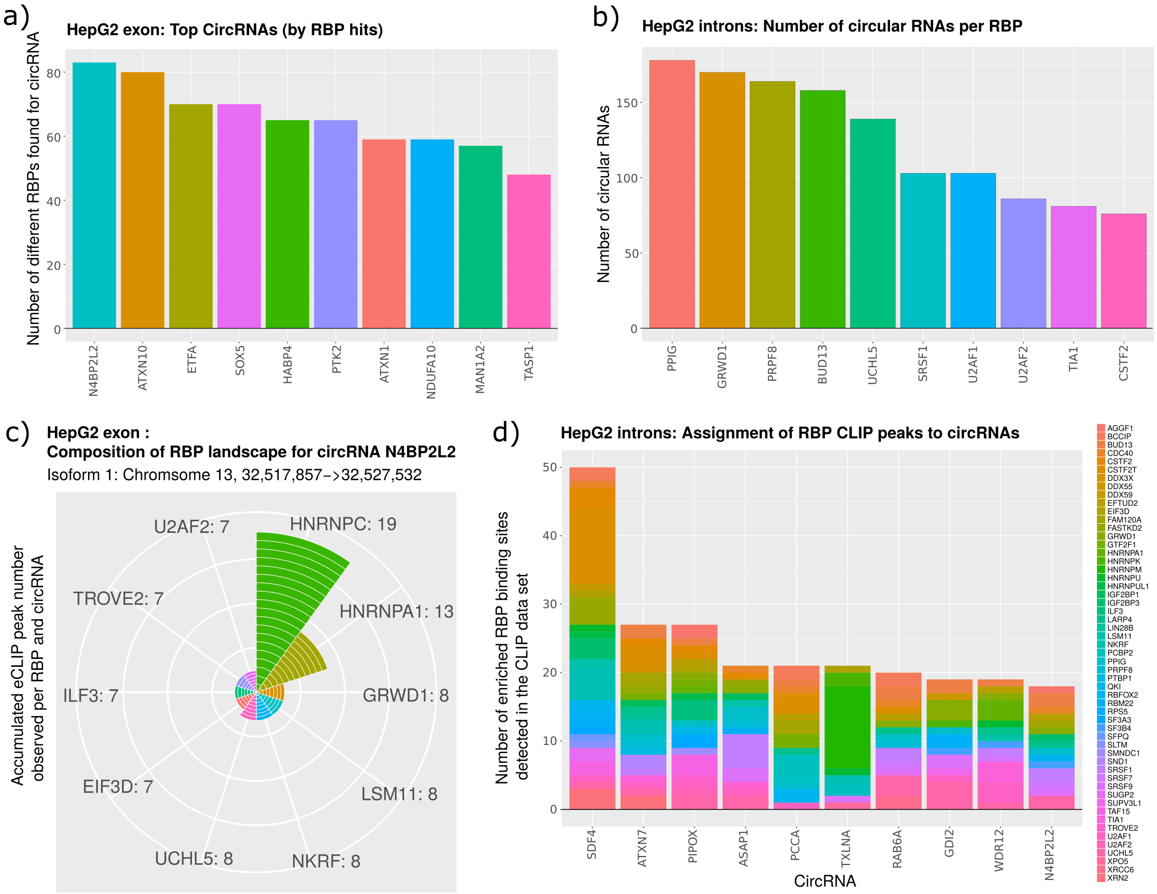
Supplementary Figure 3: Graphical representation of results generated by the `exon`, `circtest`, and `reconstruct` module. **a)** ENSEMBL genome annotation (build 95) for the human genome in blue showing protein coding transcripts and circRNA candidates that show a significant ($p < 0.05$) enrichment in the RNase R-trated samples compared to the untreated samples (green). **b)** Reconstructed circRNA structure by the reconstruction module. Shown are the identical structures for K562 (green) and HepG2 (purple). **c)** Graphical representation of results produced by the `exon` module. Shown are the fold-changes for exons that show differential RNase R-resistance between experimental conditions (brown: up, blue: down).

circtools `exon`

The `exon` module complements the `reconstruction` module in that within this module, the focus is set to test particular exons of circRNA candidates for resistance to RNAseR treatment. Hence, to further extend the researcher's knowledge of the circRNA structure, results from the `detect` module are combined with exon read counts from StringTie runs. Herein, only multi-exon genes are considered. The results encompass 3,377 circRNA-related exons for K562, which show differential RNAseR resistance ($p < 0.05$), while for HepG2 1,659 exons show differential RNAseR resistance (Supplementary Fig.1 e). The exon analysis also confirms the results of the example circRNA circN4BP2L2-1 `reconstruction` module, showing a positive Log_2 fold change for the 4 exons predicted by the reconstruction algorithm (Supplementary Fig.2 c).

circtools `enrich`

As one of the last steps of the workflow the significantly enriched candidate circRNAs produced by the `circtest` module were used as input for the `enrich` module together with eCLIP data provided through the ENCODE project (Van Nostrand *et al.*, 2016) for the two cell lines and yielded 1,667 and 1,012 circRNAs that show a significant ($p < 0.05$) enrichment for RBP peaks in exons for K562 and HepG2 (Supplementary Fig.1 f).

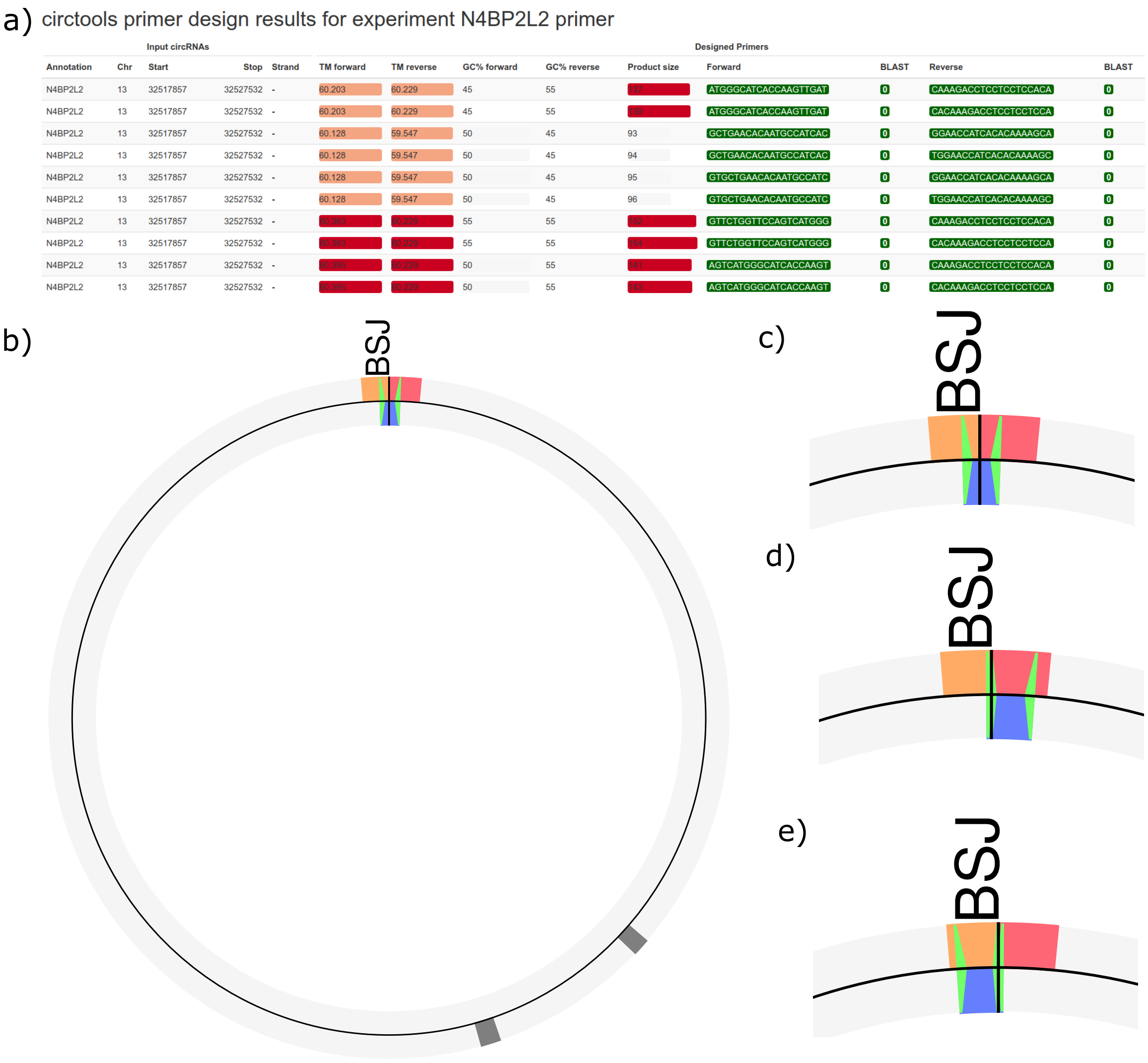


Supplementary Figure 4: Visualization of results generated by the enrichment module. **a)** Top 10 circRNAs enriched ($p < 0.05$) for eCLIP peaks for the HepG2 data set. **b)** Top 10 RBPs with enrichment within the flanking introns (max. $+/-$ 2kb) of the significantly enriched circRNA candidates of the HepG2 cell line **c)** Detailed view of the RBP eCLIP peaks enriched in the exons of isoform 1 of circN4BP2L2 **d)** Combined view of circRNAs enriched for RBP eCLIP peaks in the flanking introns and the landscape of the RBPs that are enriched within the flanking intronic regions.

Going more into detail, in HepG2 the circRNA circN4BP2L2 shows more than 80 RBPs that show a significant enrichment of eCLIP peaks within the exons, while circATXN10 follows closely (Supplementary Fig.4 a). When focusing the enrichment analysis on introns instead of exons (the `enrich` module can be configured to use any annotated feature for enrichment analyses) our data shows, that the number of significantly enriched circRNAs per RBP varies. Several RBPs (PPIG, GRWD1, PRPF8, BUD13, and UCHL5) show elevated counts compared to other RBPs (Supplementary Fig.4 b). The `enrich` module, like most other modules allows to view results on a genome-wide and per-circRNA level. Detailed inspection of the landscape of enriched RBPs for circRNA circN4BP2L2 show that specifically the heterogeneous nuclear ribonucleoproteins (hnRNPs) HNRNPC and HNRNPA1 display a high number of significant eCLIP peaks compared to other RBPs also enriched on this circRNA (Supplementary Fig.4 c). When changing back to the genome-wide view and setting the focus on the raw number of eCLIP peaks per circRNA, in HepG2 we can observe that circSDF4 accumulates nearly twice as much peaks as the second circRNA in the list, circATXN7 (Supplementary Fig.4 d). This type of plot also is able to show different preferences for RBP binding sites, like a strong prevalence for hnRNPs in TXLNA (Supplementary Fig.4 d).

circtools `primex`

In the last stage of the `circtools` workflow researchers have the possibility to design specific primer pairs for selected circRNAs of interest (Supplementary Fig.1 g). The `primex` module only requires a circRNA ID, genomic position or host gene name in order to automatically design optimal primer pairs that are circRNA specific, do not amplify the linear host gene, and therefore allow in combination with qRT-PCR to test and verify *in silico* predictions. The module generates an easy to navigate HTML page that can be displayed in any recent browser (Supplementary Fig.5 a). The table contains the most important information such as primer sequences, melting temperature, product size, or GC content. Additionally, during the design phase primers



circtools primer design results for experiment N4BP2L2 primer

Input circRNAs					Designed Primers								
Annotation	Chr	Start	Stop	Strand	TM forward	TM reverse	GC% forward	GC% reverse	Product size	Forward	BLAST	Reverse	BLAST
N4BP2L2	13	32517857	32527532	-	60.203	60.229	45	55	137	ATGGGCATCACCAAGTTGAT	0	CAAAGACCTCCTCCTCCACA	0
N4BP2L2	13	32517857	32527532	-	60.203	60.229	45	55	139	ATGGGCATCACCAAGTTGAT	0	CACAAAGACCTCCTCCTCCA	0
N4BP2L2	13	32517857	32527532	-	60.128	59.547	50	45	93	GCTGAACACAATGCCATCAC	0	GGAACCATCACACAAAAGCA	0
N4BP2L2	13	32517857	32527532	-	60.128	59.547	50	45	94	GCTGAACACAATGCCATCAC	0	TGGAACCATCACACAAAAGC	0
N4BP2L2	13	32517857	32527532	-	60.128	59.547	50	45	95	GTGCTGAACACAATGCCATC	0	GGAACCATCACACAAAAGCA	0
N4BP2L2	13	32517857	32527532	-	60.128	59.547	50	45	96	GTGCTGAACACAATGCCATC	0	TGGAACCATCACACAAAAGC	0
N4BP2L2	13	32517857	32527532	-	60.363	60.229	55	55	152	GTTCTGGTTCCAGTCATGGG	0	CAAAGACCTCCTCCTCCACA	0
N4BP2L2	13	32517857	32527532	-	60.363	60.229	55	55	154	GTTCTGGTTCCAGTCATGGG	0	CACAAAGACCTCCTCCTCCA	0
N4BP2L2	13	32517857	32527532	-	60.395	60.229	50	55	141	AGTCATGGGCATCACCAAGT	0	CAAAGACCTCCTCCTCCACA	0
N4BP2L2	13	32517857	32527532	-	60.395	60.229	50	55	143	AGTCATGGGCATCACCAAGT	0	CACAAAGACCTCCTCCTCCA	0

Supplementary Figure 5: Helpful visualization output produced by the primer design module. **a)** Table of all designed primers. Columns contain information about position, PCR product size, melting temperature and GC content for forward and reverse primer as well as BLAST results for possible cross-hybridization of the designed primers. **b)** To-scale representation of the circRNA based on the supplied genome annotation. Annotated exons are shown in thick gray, the first and last exon are colored in red and orange respectively. Forward and reverse primer are depicted in green, the final PCR product is shown in blue. **c)** Default primer design mode, positioning both primers in the flanking exons. **d)** Forward primer is placed over the BSJ **e)** Reverse primer is placed over the BSJ.

can automatically be checked for specificity via an online BLAST search. The HTML page also contains a full graphical representation of the circRNA, including known exons from the genome annotation (Supplementary Fig.5 b). In default mode, the `primex` module generates primer pairs that have the forward and reverse primer each located within the fused exons, thus the PCR product spans the BSJ (Supplementary Fig.4 c). However, for specific use cases it might be desirable to have either the forward or reverse primer located directly on the BSJ (Supplementary Fig.5 d) & e).

References

Cheng, J., Metge, F., and Dieterich, C. (2016). Specific identification and quantification of circular RNAs from sequencing data. *Bioinformatics*, pages 1–2.

Dobin, A., Davis, C. a., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, **29**(1), 15–21.

Jakobi, T., Czaja-Hasse, L. F., Reinhardt, R., and Dieterich, C. (2016). Profiling and validation of the circular rna repertoire in adult murine hearts. *Genomics, Proteomics & Bioinformatics*, **14**(4), 216–223.

Jeck, W. R., Sorrentino, J. a., Wang, K., Slevin, M. K., Burd, C. E., Liu, J., Marzluff, W. F., and Sharpless, N. E. (2013). Circular RNAs are abundant, conserved, and associated with ALU repeats. *RNA (New York, N.Y.)*, **19**(2), 141–157.

Metge, F., Czaja-Hasse, L. F., Reinhardt, R., and Dieterich, C. (2017). FUCHS—towards full circular RNA characterization using RNAseq. *PeerJ*, **5**, e2934.

Pertea, M., Pertea, G. M., Antonescu, C. M., Chang, T. C., Mendell, J. T., and Salzberg, S. L. (2015). StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology*, **33**(3), 290–295.

Robinson, J. T., Thorvaldsdóttir, H., Winckler, W., Guttman, M., Lander, E. S., Getz, G., and Mesirov, J. P. (2011). Integrative genomics viewer. *Nature Biotechnology*, **29**, 24.

Roehr, J. T., Dieterich, C., and Reinert, K. (2017). Flexbar 3.0 - SIMD and multicore parallelization. *Bioinformatics*, **33**(18), 2941–2942.

The ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**(7414), 57–74.

Van Nostrand, E. L., Pratt, G. A., Shishkin, A. A., Gelboin-Burkhart, C., Fang, M. Y., Sundararaman, B., Blue, S. M., Nguyen, T. B., Surka, C., Elkins, K., Stanton, R., Rigo, F., Guttman, M., and Yeo, G. W. (2016). Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). *Nature methods*, **13**(November 2015), 1–9.