Supplementary Materials for

"Accurate assembly of circular RNAs with TERRACE"

Tasfia Zahin[1], Qian Shi[1,†], Xiaofei Carl Zang[2,†], and Mingfu Shao[1,2,*]

[1]Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA

[2]Huck Institutes of the Life Sciences, The Pennsylvania State University, University Park, PA 16802, USA

Contents

1 Supplemental Methods **2**

2 Supplemental Note **7**

3 Supplemental Figures **10**

4 Supplemental Tables **18**

5 Supplemental References **25**

[†]contributed equally to this work

[*]to whom correspondence should be addressed; email: mxs2589@psu.edu

Supplemental Methods

Identification of back-spliced reads

The back-spliced reads (see Supplemental Fig. S1) are pivotal in detecting circRNAs. TERRACE identifies back-spliced reads from two sources: *chimerically aligned* reads in the input alignment, and by a new, light-weight, junction-targeted mapping algorithm. A chimerically aligned read is a special class of reads where different portions of it are aligned to different locations of the reference genome. These reads are indicative of structural variation, including BSJs. In the BAM format, one of its alignments is recorded as *primary* and others as *supplementary* alignments. TERRACE looks for chimerically aligned reads with only one supplementary alignment. Let *R1* and *R2* be the two ends of a paired-end read, where we assume *R1* is chimerically aligned, with its primary and supplementary alignment being denoted as *R1.primary* and *R1.supple* respectively. A pattern often exists in the CIGAR strings if R_1 contains a BSJ. An example is given in Supplemental Fig. S1, where the CIGAR strings of *R1.primary* and *R1.supple* are `30H70M` and `30M70S`, respectively. The 30 matched base pairs in the supplementary alignment complement the 30 unaligned portion (*hard clipped*, denoted by an 'H') in the primary alignment while the 70 matched base pairs in the primary alignment complement the 70 *soft clipped* portion in the supplementary alignment. Such a complementary relationship strongly indicates a BSJ. TERRACE collects chimerically aligned reads satisfying this relationship as back-spliced reads.

TERRACE also implements a new method to identify more back-spliced reads whose chimeric property are not captured by the aligner. First, splicing positions are extracted from read junctions (represented by an 'N' in the CIGAR; a junction specifies two splicing positions). Additional splicing positions are also identified from the annotated transcripts given a reference transcriptome is provided. The collected splicing positions will be used as supporting evidence for BSJs. Next, the reads that have soft clips at either end greater than a threshold (a parameter of TERRACE with a default value of 15) are considered candidates for back-spliced reads. The sequence of the soft clipped region, denoted as S, will be remapped to the reference genome at a splicing position to identify a significant match. More specifically, for each candidate splicing position that may form a BSJ with the soft clipped region (depending on the relative locations of *R1* and *R2*), we

extract a sequence of the same length as S from the reference genome, denoted as T, and calculate the Jaccard index of the two sets of kmers in S and T, where by default $k = 10$. If the Jaccard is greater than a threshold (0.9 by default) and there does not exist another such T, TERRACE will use it and create a supplementary alignment record for S by mapping it to T. Now we have a new back-spliced read that can be treated as the same way as those identified from chimerically aligned reads.

Transforming assembly to bridging

A back-spliced read is presumably expressed from a circRNA; we aim for assembling its original circRNA for each back-spliced read. Recall that a back-spliced read R with ends *R1* and *R2* consists of three segments *R1.primary*, *R2*, and *R1.supple*, assuming *R1* contains the BSJ. We already know that these three fragments must be part of the original circRNA, and that the circRNA uses the BSJ to form its circular structure. What we still miss is how the three segments are connected in the circRNA, as the three segments may be aligned to distant portions of the reference genome. We refer to the task of closing the two gaps among the three segments in a back-spliced read as "bridging": assembling the full-length circRNA now becomes bridging. See Supplemental Fig. S2(b).

To formalize the bridging task, we introduce the underlying data structure: splice graph. See Supplemental Fig. S2(a). Splice graph has been instrumental in studying alternative splicing and in assembling (linear) transcripts. It is a weighted directed graph, denoted as $G = (V, E, w)$, that organizes the splicing and coverage information in the read alignment of a gene locus. To construct G, junctions from reads and annotated transcripts (if provided) are collected and their splicing positions will be used to partition the reference genome into (partial) exons and introns, in which the partial exons will be the vertices V of G. A directed edge $e \in E$ is placed between vertex u and vertex v if there exists a read that spans u and v; the weight $w(e)$ of edge e will be the number of such reads. A source vertex s is added and connected to any vertex u with in-degree of 0 using weight $w(s, u) = \sum_{v:(u,v)\in E} w(u, v)$; a sink vertex t is also added and any vertex v with out-degree of 0 will be connected to t with weight $w(v, t) = \sum_{u:(u,v)\in E} w(u, v)$.

The three fragments of a back-spliced read can be represented as three paths in the splice graph G. The bridging task now involves finding two *bridging paths* in G that connect the three known

paths. Solving this results in a single path that threads the 3 fragments, which forms the circRNA together with the BSJ. Due to alternative splicing and sequencing/alignment errors, multiple possible bridging paths exist. We therefore need a characterization for better bridging paths and an efficient algorithm to calculate them.

Formulation and algorithm for bridging

Let $A = (a_1, a_2, \cdots, a_i)$, $B = (b_1, b_2, \cdots, b_j)$, and $C = (c_1, c_2, \cdots, c_k)$ be the 3 paths in G corresponding to the 3 fragments of a back-spliced read. We aim to find the "best" paths in G from a_i to b_1 and from b_j to c_1. We adopt a definition we proposed in reconstructing the entire fragment of paired-end RNA-seq reads Zhang et al. (2022); Li and Shao (2023). The idea was to seek a path whose "bottleneck weight" is maximized, which is effective for selecting the path with the strongest support and for excluding false paths due to errors which often contain edges with a small weight. Formally, we define the *score* of a path p as the smallest weight over all edges in p. The formulation is to find a path p_1 from a_i to b_1 and a path p_2 from b_j to c_1 such that the score of p_1 and that of p_2 are maximized. The optimal p_1 and p_2 can be calculated independently (we assume paths A, B, and C are vertex-disjoint; otherwise, they can be either bridged trivially or bridging is not possible in which case this read will be discarded). An efficient dynamic programming algorithm can be designed to find optimal p_1 and p_2. Please refer to Zhang et al. (2022) for details. The programming algorithm can be extended to produce suboptimal paths as candidates (by default TERRACE calculates top 10 optimal paths for each back-spliced read), which will be combined with additional information for selection.

Selection of candidate paths

Let P be the set of candidate full-length circular paths for a back-spliced read. We apply some heuristic procedures to filter false-positive paths. If a path $p \in P$ contains a vertex (partial exon) in which a region of length larger than a threshold (by default 10 base pairs) is not covered by any read then p will be removed from P. For every pair of paths $p, q \in P$, if an intron of p is fully covered by an exon of q, then q will be removed. This procedure helps to filter out paths with anticipated intron retentions. If there exists a path $p \in P$ with bottleneck weight higher than a chosen threshold c (by default $c = 1$), all paths in P with bottleneck weight smaller than or equal

to c are discarded. This procedure aims to remove less reliable paths when more reliable ones exist. We use P_1 to denote the set of survived paths.

If a reference transcriptome is provided, TERRACE will then identify more candidate full-length circular paths from it. See Supplemental Fig. S3. We define an annotated (linear) transcript q is *compatible* with a back-spliced read R if both the BSJ and the splicing positions in the three fragments of R match the splicing positions of q. Unique compatible paths bounded by the BSJ will be collected as another set P_2 of candidate full-length circular paths for R. If $P_1 \cap P_2 \neq \emptyset$, the path in $P_1 \cap P_2$ with maximized bottleneck weight will be picked; if $P_1 \cap P_2 = \emptyset$ and $P_1 \neq \emptyset$, we pick the path in P_1 with maximized bottleneck weight regardless of P_2 (i.e., we give higher priority to paths inferred from the read alignment rather than from reference annotation). If $P_1 = \emptyset$ and P_2 contains a single path, we pick that path from P_2. If P_2 contains multiple paths, ambiguity exists, and hence we discard the read. A read is also discarded if $P_1 \cup P_2 = \emptyset$. The selected full-length circular path is then transformed to a fully annotated circRNA by borrowing genomic coordinates from the reference genome.

The assembled circRNA then goes through a series of quality check. For example, if the circRNA has a single exon greater than 2000bp or a multi exon greater than 1000bp, it is discarded because of possible intron retention. A circRNA is also discarded if the number of exons in the path is greater than 15 since it is most likely to be a false positive arising from spurious small junctions. Note that multiple back-spliced reads can produce identical circRNA. We merge identical circRNAs to a single instance and the number of back-spliced reads generating this circRNA is recorded as its *abundance*. We observe that circRNAs may differ by a few base pairs (50bp by default) at their BSJs but share the same intron chain. In such case, the circRNA with higher abundance is retained. To further investigate the effect of this merging parameter on the results, we conduct additional experiments by varying the threshold from 0 to 100 and generate precision-recall curves. Supplemental Figure S6 shows that the change in precision or recall due to the variation is insignificant. Therefore, we conclude that a very low percentage of the circRNAs have such few base pairs difference at their BSJs.

Scoring assembled circular RNAs

Assigning a confidence score to assembled circRNAs is desirable to ensure that those with higher scores are more likely to be true. Traditionally, abundance has served this purpose in transcript assembly as it shows a high correlation to the correctness of the assembled transcripts. We investigate whether a machine-learning approach could yield a more accurate scoring function. To this end, we extract 13 features to characterize each assembled circRNA, ranging from its abundance to the (average) length of the soft clips of back-spliced reads. Please refer to Supplemental Note for a detailed description of all features.

We use a random forest model trained on a single tissue sample (brain) and tested it on other samples. The default loss function from the *python scikit-learn* package was used during the training process. We run TERRACE (both with and without annotation) to generate a combined feature file from the brain sample. Each entry within the feature file corresponds to distinct features extracted from the output circRNA list produced by TERRACE. Using the ground truth, we assigned labels to each entry within the feature file indicating whether the circRNA is a true one. The labeled feature file is then fed to the random forest model for training. For testing a sample, we used the feature file of the sample without any label and fed it to the pre-trained model. The model generates a list of score or probabilities that represent the reliability of the assembled circRNAs.

We aim to train a model that is capable of generalizing across different tissues. This is challenging due to the variability between samples and the limited number of instances available. To fortify its stability, we incorporate the number of reference transcripts present in each instance (gene locus) as features, and use the assembled circRNAs by TERRACE with and without reference annotations to train the model. This approach is proven beneficial in generalizing, especially when the test set is markedly different from the training set or when the dataset is small (such as skeletal muscle). Additionally, this enables a single model to be applicable on circRNAs assembled both with and without annotations (instead of training two models).

Supplemental Note

Description of random forest features

We elaborate the features used to train the scoring function described in Methods and provide our insights on why they can be informative. The features are characterizing an assembled circRNA x.

1. Coverage. This is the number of back-spliced reads that produce x. Intuitively, a circRNA supported by a higher number of back-spliced reads is more likely to be correct.

2. Count of additional back-spliced reads that produce x. By additional, we mean those back-spliced reads identified by TERRACE but missed by the aligner. The argument for including this as a feature is similar to the intuition in 1, i.e., a higher abundance of reads is an evidence for real circRNAs.

3. Sum of soft clip lengths of the back-spliced reads that produce x. Back-spliced read with a longer soft clip is more likely to be correctly aligned to the correct junction. Hence, circular RNAs characterized by longer soft clips are more likely to be genuine.

4. Sum of bridging path scores of reads that produce x. Recall that the score of a path represents its bottleneck weight. Paths with higher scores receive support from more reads, and may be an important factor in distinguishing between true and false instances.

5. Sum of the count of full-length candidate paths from back-spliced reads that produce x. By default, TERRACE considers the top 10 bridging paths and selects a set of them using some filtering criteria (Supplemental Methods: Selection of candidate paths) to be further analyzed. The greater the number of paths in this selected set, the more likely it is to deviate from choosing the correct path.

6. Count of bridging path type of back-spliced reads producing x. The path type refers to whether the selected path is inferred from read alignments or reference transcripts and if the path length is within the range of insert size. When selecting a bridging path, we normally would want to assign a higher priority to paths inferred from read alignment than from the reference annotation, aiming to construct novel circRNAs. However, if the set of paths inferred from reads is empty, sometimes a path from the annotation (if provided) may help

recover a correct circRNA specially when the coverage within a region is low. These features may provide guidance to the machine learning model to make accurate decisions.

7. Number of exons in x. In certain high-coverage areas of specific samples, numerous false circRNAs with a large number of exons are detected due to the presence of many small splicing sites. Hence, this information can be an important factor.

8. Total length of exons in x. We observe that some circRNAs have extended exon lengths, which are misleading and primarily caused by intron retentions. Therefore, considering the total exon length as a feature could prove valuable in making decisions.

9. Maximum length of exons in x. Intuition similar to 8.

10. Minimum length of exons in x. Intuition similar to 8.

11. Total number of reads in the region where x is identified. We observe a few instances where a false circRNA is supported by many reads, primarily in some high coverage regions of certain samples. Involving the number of reads (both ordinary and back-spliced) as a feature may help to normalize this bias.

12. Total number of reference transcripts in the region where x is identified. The splicing positions from reference transcripts (when annotation provided) influences the identification of additional chimeric reads and adds to the set of paths to be considered for bridging. Therefore, the number of annotated transcripts in a region may serve as a useful feature for learning a better score.

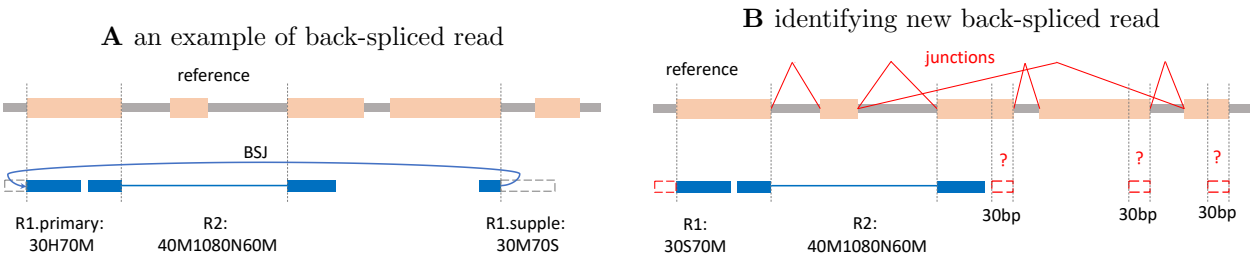
Comparison of runtime and memory usage

Table S1 shows the CPU time (user time plus kernel time) of various tools on the real dataset. CIRCexplorer2 has the fastest execution time which is expected given it utilizes a more compact annotation file than the raw version. TERRACE is the second fastest, surpassing both CircAST and CIRI-full by a large margin. Overall, TERRACE delivers vastly superior accuracy within a reasonable processing time.

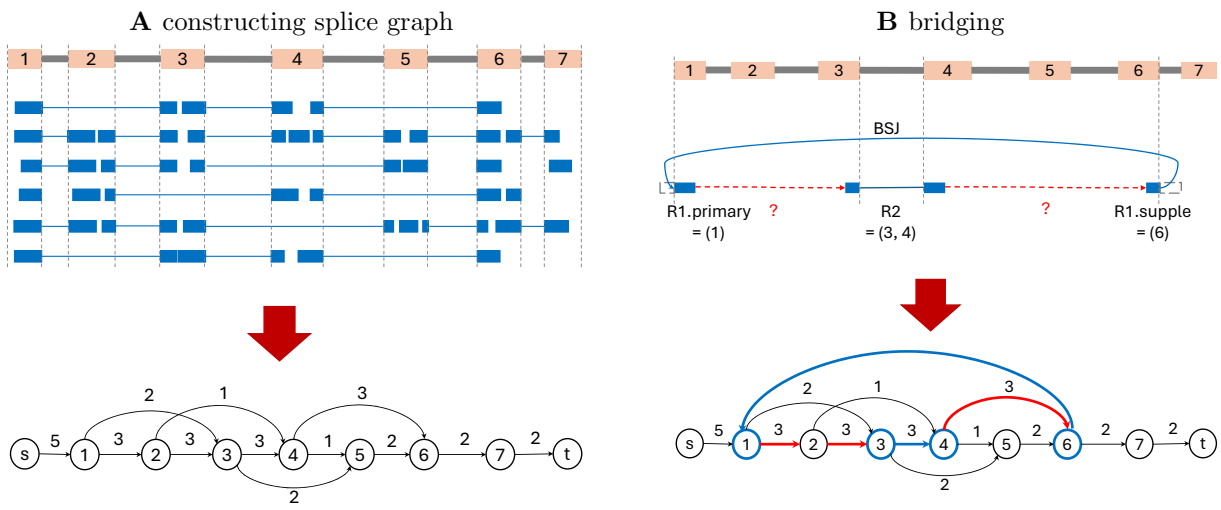
Table S2 shows the peak memory usage of various tools on the real dataset. While TERRACE

may not excel in peak memory usage, it still operates within an acceptable range and outperforms CIRI-full significantly. It is important to highlight that the both running time and peak memory usage values of TERRACE are very similar regardless of whether an annotation is provided (i.e., the use of annotation does not significantly impact its computational efficiency).

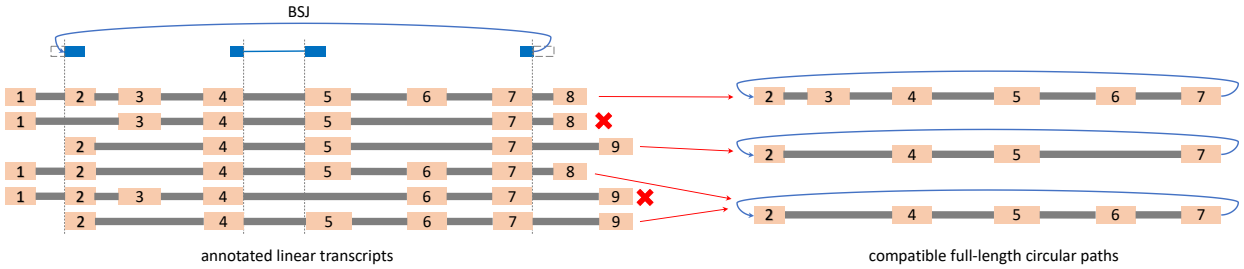
Supplemental Figures



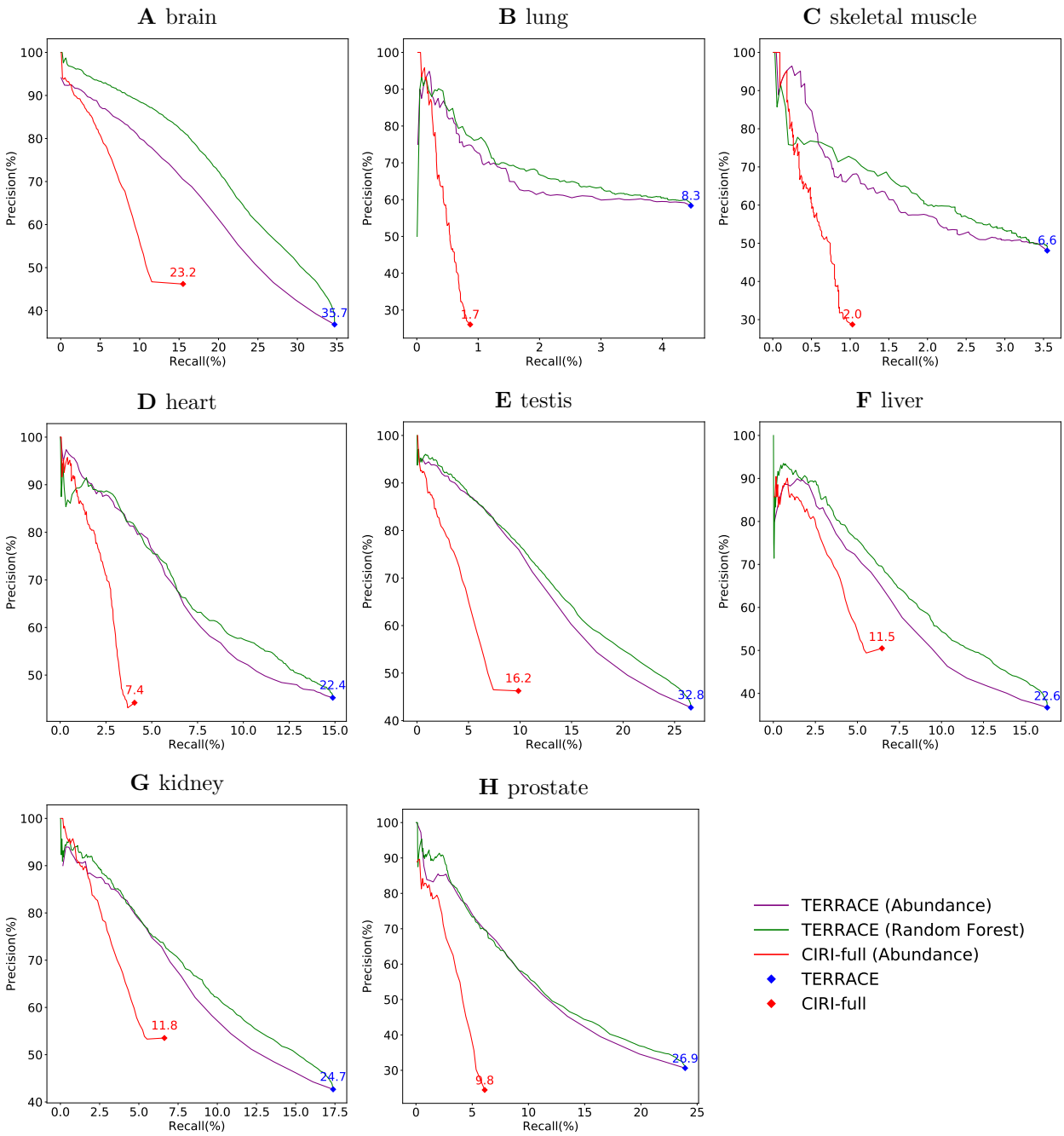
Supplemental Figure S1: **A**, the CIGAR strings of the 3 segments in a back-spliced read. **B**, the soft clipped sequence (dashed box in red) will be compared with the reference sequence next to a splicing position of the same length.



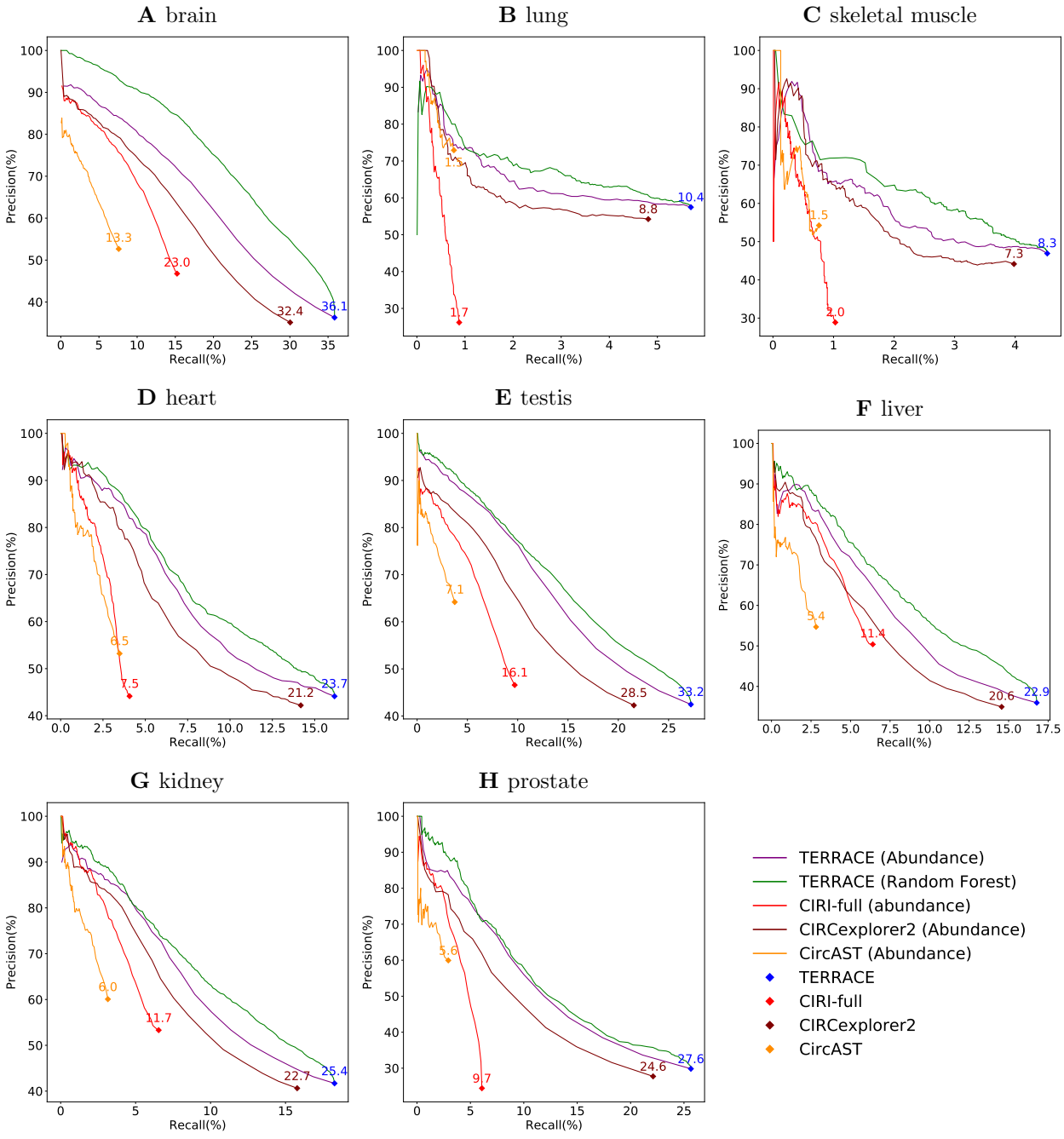
Supplemental Figure S2: **A**, constructing splice graph from reads alignment. **B**, the 3 fragments namely *R1.primary*, *R2*, and *R1.supple* of a back-spliced read are represented as paths in the splice graph, which are (1), (3,4), and (6), respectively. The two optimal bridging paths, which maximize the "bottleneck" weight, are marked in red. The resulting full-length circular path for this back-spliced read is $1 \to 2 \to 3 \to 4 \to 6 \to 1$.



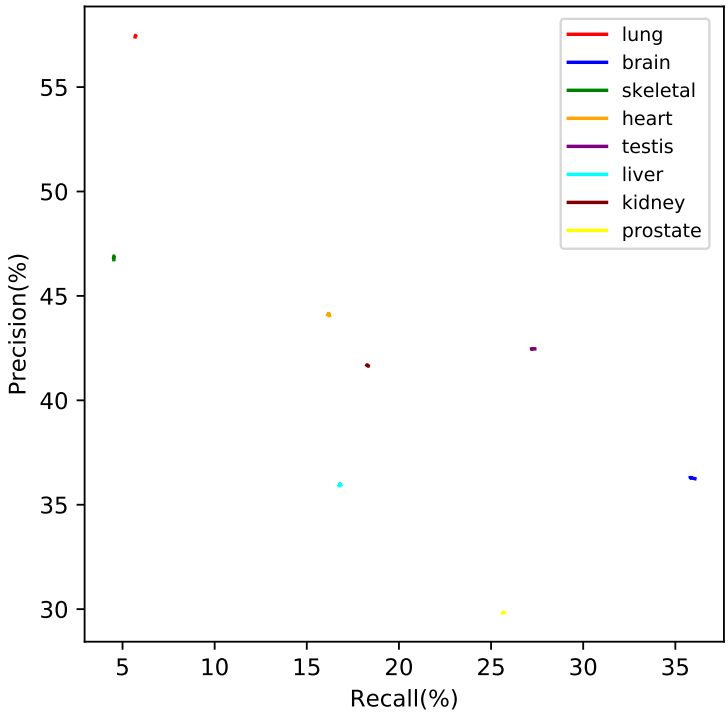
Supplemental Figure S3: Identifying compatible full-length circular paths for a back-spliced read from annotated transcripts.



Supplemental Figure S4: Comparison of precision-recall curves of methods without annotation. Fscores (%) are indicated on top of data points.

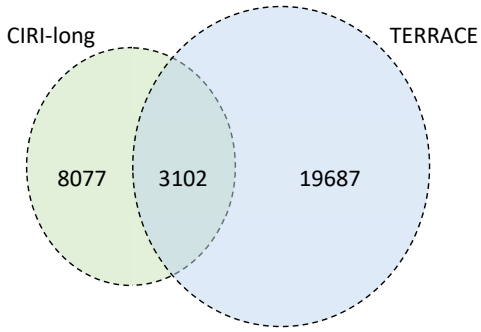


Supplemental Figure S5: Comparison of precision-recall curves of methods with annotation. Fscores (%) are indicated on top of data points.

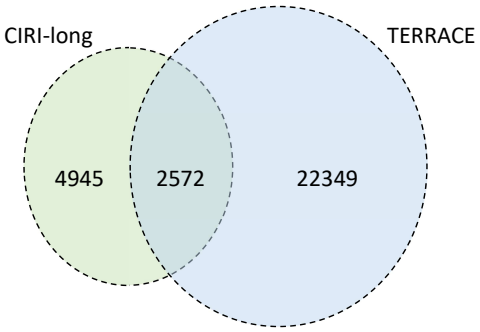


Supplemental Figure S6: Precision-recall curves for varying merging parameter from 0 to 100 on the human tissue datasets show negligible difference in accuracy.

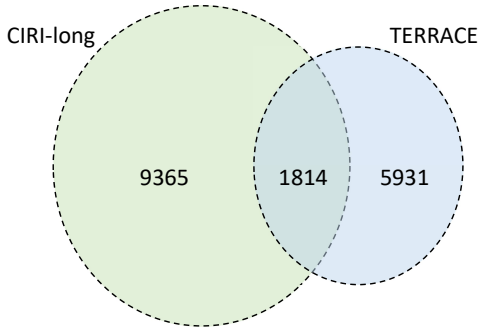
A Illumina RNaseR rep1 vs Long SMARTer H- Atail rep1



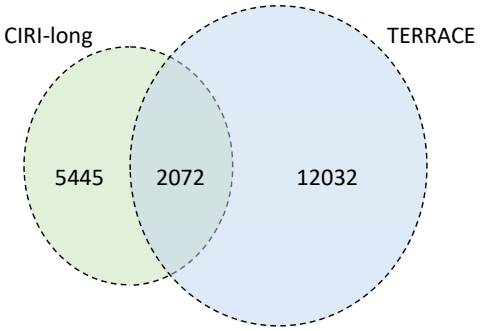
B Illumina RNaseR rep2 vs Long SMARTer H- Atail rep2



C Illumina Total rep1 vs Long SMARTer H- Atail rep1

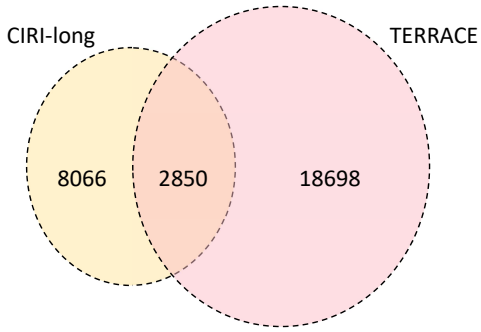


D Illumina Total rep2 vs Long SMARTer H- Atail rep2

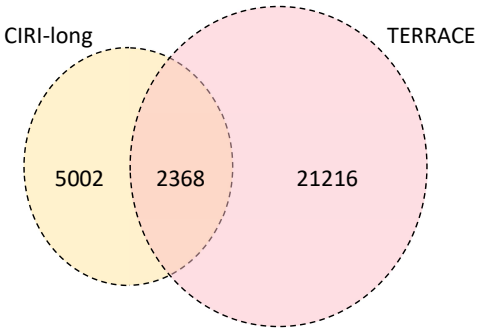


Supplemental Figure S7: Number of overlapping circRNAs detected by CIRI-long and TERRACE on mouse brain samples with annotation.

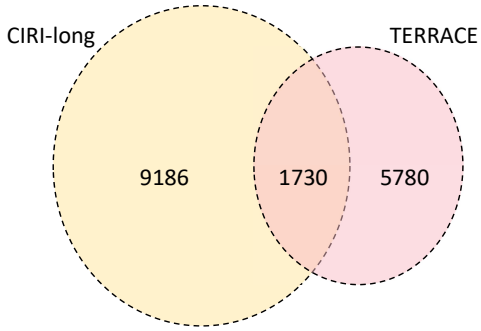
A Illumina RNaseR rep1 vs Long SMARTer H- Atail rep1



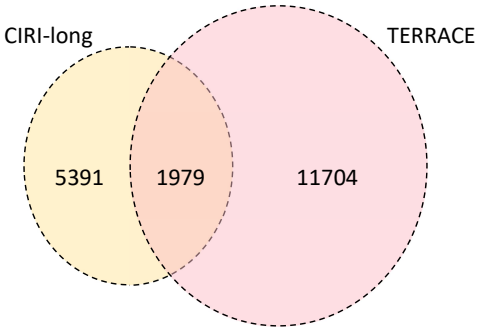
B Illumina RNaseR rep2 vs Long SMARTer H- Atail rep2



C Illumina Total rep1 vs Long SMARTer H- Atail rep1



D Illumina Total rep2 vs Long SMARTer H- Atail rep2



Supplemental Figure S8: Number of overlapping circRNAs detected by CIRI-long and TERRACE on mouse brain samples without annotation.

Supplemental Tables

Supplemental Table S1: CPU time (in minutes) for different tools on various datasets.

sample	methods w/o annotation		methods with annotation			
	CIRI-full	TERRACE	CIRCexplorer2	CircAST	CIRI-full	TERRACE
lung	265	29	0.14	249	266	26
brain	471	42	0.21	1234	489	45
skeletal	334	40	0.10	260	337	50
heart	380	32	0.10	488	389	35
testis	427	39	0.14	922	472	40
liver	330	31	0.08	423	345	31
kidney	352	31	0.10	554	370	32
prostate	253	52	0.24	229	307	57
average	352	37	0.14	545	372	40

Supplemental Table S2: Peak memory usage (in GB) for different tools on various datasets.

sample	methods w/o annotation		methods with annotation			
	CIRI-full	TERRACE	CIRCexplorer2	CircAST	CIRI-full	TERRACE
lung	10.1	16.9	0.14	0.07	11.0	16.7
brain	80.4	11.2	0.16	0.11	81.1	11.4
skeletal	22.2	19.8	0.14	0.10	22.8	20.2
heart	30.8	10.9	0.14	0.61	31.4	11.1
testis	74.8	5.5	0.15	0.09	76.2	5.6
liver	271.9	17.7	0.14	0.08	31.7	17.9
kidney	32.0	10.3	0.14	0.08	33.1	10.1
prostate	21.7	23.4	0.15	0.08	22.4	23.6
average	68.0	14.5	0.15	0.15	38.7	14.6

Supplemental Table S3: Comparison of pAUC (%, w/o annotation). Δ% represents the percentage improvement of TERRACE over other methods.

sample	TERRACE vs. CIRI-full constrained by *recall*			TERRACE vs. CIRI-full constrained by *precision*		
	TERRACE	CIRI-full	Δ%	TERRACE	CIRI-full	Δ%
lung	65.2	52.3	24.6	66.8	14.3	365.9
brain	1380.1	1047.8	31.7	969.7	339.0	185.9
skeletal	75.7	62.5	21.0	59.9	17.3	246.0
heart	347.6	294.4	18.0	334.2	113.0	195.5
testis	851.9	645.4	31.9	618.4	196.4	214.9
liver	527.1	450.4	17.0	255.9	136.8	87.0
kidney	553.9	483.3	14.6	267.9	133.8	100.1
prostate	495.7	364.1	36.1	598.0	161.4	270.3

Supplemental Table S4: Comparison of pAUC (%, constrained by *recall*, with annotation). Δ% represents the percentage improvement of TERRACE over other methods.

sample	TERRACE vs. CIRI-full			TERRACE vs. CIRCexplorer2			TERRACE vs. CircAST		
	TERRACE	CIRI-full	Δ%	TERRACE	CIRCexplorer2	Δ%	TERRACE	CircAST	Δ%
lung	70.9	56.5	25.3	332.2	297.7	11.5	62.1	65.5	-5.1
brain	1363.2	1098.3	24.1	2418.8	1884.8	28.3	721.1	513.5	40.4
skeletal	72.7	61.6	18.1	257.7	225.3	14.3	52.1	52.0	0.2
heart	359.9	303.3	18.6	995.2	886.9	12.2	313.5	259.1	20.9
testis	843.1	685.8	22.9	1598.2	1378.1	15.9	345.3	282.7	22.1
liver	524.3	466.8	12.3	962.1	804.6	19.5	252.0	196.7	28.1
kidney	564.8	498.6	13.2	1113.0	988.9	12.5	284.0	239.8	18.4
prostate	517.3	404.5	27.8	1305.8	1093.3	19.4	272.1	203.1	33.9

Supplemental Table S5: Comparison of pAUC (%, constrained by *precision*, with annotation). Δ% represents the percentage improvement of TERRACE over other methods.

sample	TERRACE vs. CIRI-full			TERRACE vs. CIRCexplorer2			TERRACE vs. CircAST		
	TERRACE	CIRI-full	Δ%	TERRACE	CIRCexplorer2	Δ%	TERRACE	CircAST	Δ%
lung	84.9	15.0	464.5	63.6	38.9	63.5	11.1	7.7	42.4
brain	996.4	400.5	148.7	1427.5	766.4	86.2	703.7	119.6	488.0
skeletal	78.2	17.5	345.3	78.2	44.4	76.1	57.7	14.2	305.4
heart	395.7	128.4	208.2	395.7	265.4	49.0	260.6	77.9	234.2
testis	631.5	242.3	160.5	742.0	418.1	77.4	275.0	44.8	513.3
liver	257.0	147.8	73.9	476.1	288.7	64.9	212.6	43.3	390.7
kidney	289.4	152.3	89.9	485.4	333.1	45.7	205.0	53.2	284.9
prostate	663.4	219.5	202.1	666.7	409.4	62.8	179.9	28.1	540.2

Supplemental Table S6: Results for varying the parameters of CIRI-simulator, w/o annotation.

read length	circular coverage	linear coverage	TERRACE %recall	TERRACE %precision	CIRI-full %recall	CIRI-full %precision
100	10	10	80.17	97.18	58.45	91.74
75	10	10	78.33	97.03	41.41	86.73
100	5	10	64.09	96.95	34.06	83.48
100	15	10	82.97	96.67	67.7	94.19
100	10	5	79.75	97.28	56.78	91.69
100	10	15	80.07	97.2	58.78	91.83

Supplemental Table S7: Results for varying the parameters of CIRI-simulator, with annotation.

read length	circular coverage	linear coverage	TERRACE		CIRI-full		CIRCexplorer2		CircAST	
			%recall	%precision	%recall	%precision	%recall	%precision	%recall	%precision
100	10	10	83.79	95.89	58.44	91.73	76.91	87.99	1.29	95.45
75	10	10	82.74	95.62	41.41	86.74	75.35	87.8	0.89	93.58
100	5	10	67.95	95.84	34.05	83.47	61.86	88.15	0.02	100
100	15	10	87.47	95.21	67.71	94.21	81.07	87.54	8.08	96.36
100	10	5	83.6	95.86	56.74	91.66	76.85	87.76	1.13	96.84
100	10	15	83.83	95.83	58.8	91.89	76.88	87.85	1.42	95.39

Supplemental Table S8: Number of paired-end reads in the human tisue samples, number of circRNAs produced from long-reads in the isoCirc paper (which we use as ground truth for evaluation), number of circRNAs assembled and correctly identified by TERRACE, CIRI-full, and CIRCexplorer2.

sample	#reads	#circRNAs	w/o annotation				with annotation			
			TERRACE		CIRI-full		TERRACE		CIRCexplorer2	
			#detected	#correct	#detected	#correct	#detected	#correct	#detected	#correct
lung	87M	18136	1388	810	606	158	1798	1033	1608	872
brain	82M	35801	33785	12428	12024	5553	35365	12835	30611	10754
skeletal	93M	10908	805	387	390	112	1053	494	983	434
heart	79M	11223	3692	1670	1032	456	4113	1815	3770	1591
testis	90M	42633	26509	11333	9070	4195	27329	11603	21740	9188
liver	87M	11978	5314	1951	1533	774	5588	2010	4989	1744
kidney	93M	22521	9176	3915	2791	1494	9869	4115	8747	3554
prostate	83M	8114	6342	1942	2029	496	6973	2081	6469	1794

References

Li X and Shao M. 2023. On de novo bridging paired-end RNA-seq data. In *Proc. 14th ACM Conf. Bioinformatics, Computational Biolofy, and Health Informatics (ACM-BCB'23)*, 41, p. 5.

Zhang Q, Shi Q, and Shao M. 2022. Accurate assembly of multi-end rna-seq data with Scallop2. *Nature computational science* **2**: 148–152.