Supporting Information

An RNA Polymerase Ribozyme that Synthesizes its Own Ancestor

Katrina F. Tjhung, Maxim N. Shokhirev, David P. Horning, and Gerald F. Joyce

The Salk Institute

10010 North Torrey Pines Road

La Jolla, CA 92037, USA

E-mail: dhorning@salk.edu, gjoyce@salk.edu

www.pnas.org/cgi/doi/10.1073/pnas.1914282117

SI Methods

In Vitro Transcription. Fragments of the class I ligase were produced by in vitro transcription of double-stranded DNA using T7 RNA polymerase (see Table S4 for sequences). Fragments longer than 30 nucleotides were prepared as described previously (1). For fragments shorter than 30 nucleotides, in vitro transcription was performed under conditions that maximized the yield of correct-length products (2). RNA was transcribed from 50 nM DNA template that had been annealed with 60 nM synthetic oligodeoxynucleotide that contains the second strand of the T7 RNA polymerase promoter. In vitro transcription was carried out in a reaction mixture containing 15 U/μL T7 RNA polymerase, 0.002 U/μL inorganic pyrophosphatase, 5 mM each NTP, 9 mM $MgCl_2$, 1 mM spermidine, 5 mM DTT, 50 μg/mL bovine serum albumin, 0.01% Triton X-100, 80 mg/mL PEG-8000, and 40 mM Tris (pH 8.1), which was incubated at 37 °C for 2 h, followed by addition of 0.1 U/μL TurboDNase and incubation at 37 °C for 1 h. All reactions were quenched by the addition of excess EDTA relative to $MgCl_2$, then the products were purified by PAGE and subsequent ethanol precipitation.

SI References

1. D. P. Horning, G. F. Joyce, Amplification of RNA by an RNA polymerase ribozyme. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 9786–9791 (2016).
2. J. F. Milligan, D. R. Groebe, G. W. Witherell, O. C. Uhlenbeck, Oligoribonucleotide synthesis using T7 RNA polymerase and synthetic DNA templates. *Nucleic Acids Res.* **15**, 8783–8798 (1987).

SI Figure Legends

Fig. S1. Time course of RNA ligation catalyzed by various forms of the class I ligase, all prepared by in vitro transcription using T7 RNA polymerase. The ligase was provided as a contiguous strand (black circles), 3 separate fragments (white circles), or 3 separate fragments with added primer regions at both ends of each fragment (black squares). Multiple-turnover, first-order rate constants are 0.33, 0.050, and 0.0062 min^{-1}, respectively. Reaction conditions: 1 μM ligase (either contiguous or each fragment), 20 μM 5′-substrate (S2), 80 μM 3′-substrate with attached template (S3), 60 mM $MgCl_2$, 200 mM KCl, and 0.6 mM EDTA at pH 8.3 and 23 °C.

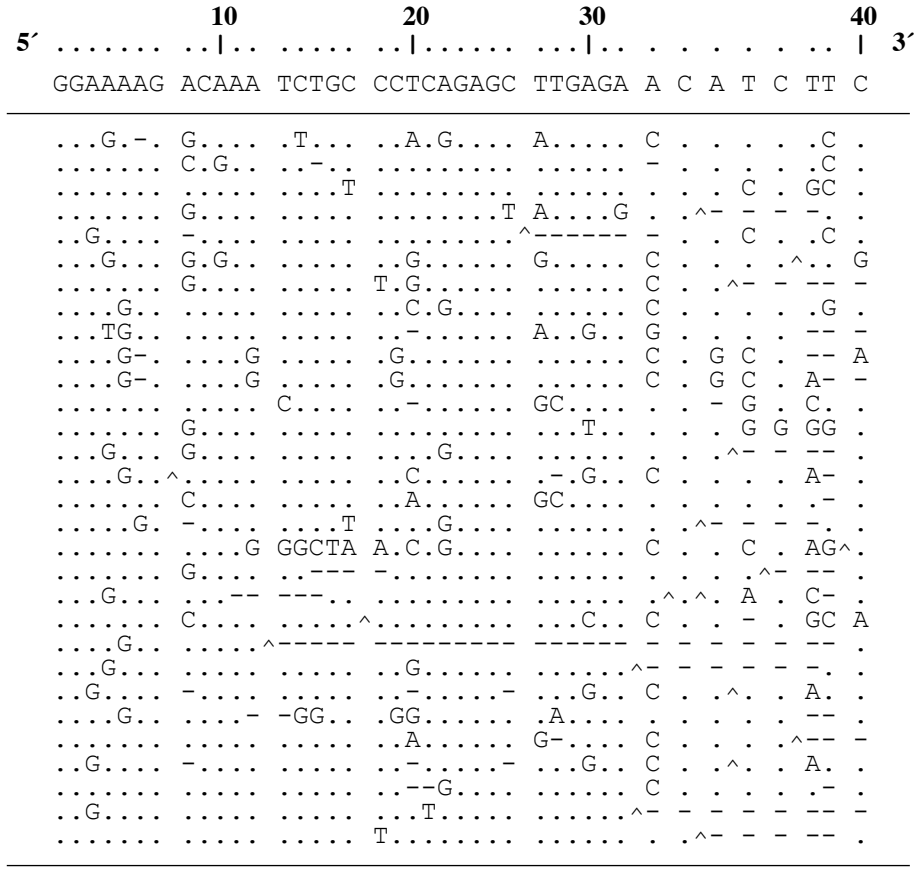
Fig. S2. Time course of RNA ligation catalyzed by the 3-fragment form of the class I ligase that had been synthesized by the 38-6 polymerase over 72 h (black circles). The uncatalyzed, RNA-templated reaction was measured under the same conditions (white circles). First-order rate constants are $6.1 \pm 0.2 \times 10^{-5}$ and $0.86 \pm 0.08 \times 10^{-5}$ h^{-1}, respectively. Values are the average of 3 replicates, with standard deviation. Reaction conditions: ±1 μM ligase fragments, 20 μM 5′-substrate (S2), 80 μM 3′-substrate (S3), 60 mM $MgCl_2$, 200 mM KCl, and 0.6 mM EDTA at pH 8.3 and 23 °C.

Table S1. Fidelity of the 24-3 and 38-6 polymerases

Pol	Expected	Observed (%)					
		A	G	C	U	Del	Ins
24-3	A	**91.7**	6.9	0.0	0.0	1.4	0.0
	G	0.0	**100.0**	0.0	0.0	0.0	0.0
	C	0.0	0.0	**98.6**	1.4	0.0	0.0
	U	0.9	0.9	0.0	**98.1**	0.0	0.0
38-6	A	**98.6**	0.0	0.0	0.0	1.4	0.0
	G	0.0	**97.3**	0.5	0.0	0.0	2.2
	C	0.0	0.0	**94.4**	5.6	0.0	0.0
	U	0.9	2.8	0.9	**94.4**	0.9	0.0

Operating on the favorable template 3′-ACGCUUCGCAC-5′, the average fidelities of 24-3 and 38-6 were 97.1% and 96.2%, respectively, based on 36 reads each and calculated as the geometric mean of the fidelities for each templating nucleobase, including deletions. Insertions were treated as a single mutation event at the position immediately upstream of the insertion.

Table S2. Sequences of cloned individuals obtained from synthesis of fragment 1 of the class I ligase by the 38-6 polymerase



Sequences of 30 individuals, aligned in reference to the correct sequence of fragment 1. The flanking primer regions are not shown. Dots indicate nucleotides that match the correct sequence, dashes indicate sites of deletion, and carat marks indicate sites of insertion of one or more nucleotides.

Table S3. Fidelity of the 24-3 and 38-6 polymerases

Pol	Expected	Observed (%)					
		A	G	C	U	Del	Ins
24-3	A	**88.0**	6.7	2.7	0.8	1.8	0.2
	G	0.5	**97.5**	0.7	0.7	0.6	0.1
	C	0.4	0.5	**93.2**	5.6	0.4	0.1
	U	1.5	1.9	5.4	**90.1**	1.1	0.1
38-6	A	**88.5**	7.0	1.2	0.4	3.0	0.2
	G	0.3	**98.5**	0.4	0.2	0.6	0.1
	C	0.3	0.7	**89.7**	8.8	0.6	0.1
	U	0.7	1.1	2.6	**94.7**	1.0	0.1

Operating on the template 3′-CUGCAUGACUACUCCGGCUUUCCGG-CUUUUCGC-5′, which encodes the hammerhead ribozyme, the average fidelities of 24-3 and 38-6 were 92.1 and 92.8%, respectively, based on deep sequencing analysis and calculated as the geometric mean of the fidelities for each templating nucleobase, including deletions. Insertions were treated as a single mutation event at the position immediately upstream of the insertion. Wobble mutations are underlined.

Table S4. Sequences of RNA and DNA molecules used in this study

Type	Name	R or DNA	Source	Sequence (5′→3′)
PCR primer	Fwd1	DNA	com	GGACTAATACGACTCACTATTAGTCATTGCCGCAC
	Rev1	DNA	com	GTCAGCCATGTGTTG
Ribozyme	24-3 pol	RNA	ivt	AGUCAUUGCCGCACGAAAGACAAAUCUGCCCUCAGAG CUUGAGAACAUCUUCGGAUGCAGAGGAGGCAGCCUUC GGUGGAACGAUCGUGCCACCGUUCUCAACACGUACCC GAACGAAAAAGACCUGACAAAAAGGCGUUGUUAGAC ACGCCCAGGUGCCAUACCCAACACAUGGCUGAC
	38-6 pol	RNA	ivt	AGUCAUUGCCGCACAAAGACAAAUCUCCCCUCAGAGC UUGAGAACAUCUACGGAUGCAGAGGAGGGGGCCUUC GGUGGAUCAAUUGUGCACCACCGUUCUCAACACGUAC CCGAACAUAAAAAGACCUGACAAAAAGGCGAUGUUA GACACGCACAGGUGCCAUACCCAACACAUGGCUGAC
	hammer-head	RNA	ivt	CACUCCACACGACGUACUGAUGAGGCCGAAAGGCCGA AAAGCG
	class I ligase	RNA	ivt	GGAAAAGACAAAUCUGCCCUCAGAGCUUGAGAACAUC UUCGGAUGCAGGGGAGGCAGCCCCCGGUGGCUUUAAC GCCAACGUUCUCAACAAUAGUGA
	F1	RNA	ivt	GGAUGCUACAUGGGAAAAGACAAAUCUGCCCUCAGAG CUUGAGAACAUCUUCGACAUUCGUGUC
	F2	RNA	ivt	GGUCGAAUGAUCGGAUGCAGGGGAGGCAGCCCCCGGU GGCGCGCAAUAGUUGGU
	F3	RNA	ivt	GGCAUAUCCAGCGCGCCAACGUUCUCAACAAUAGUGA CGUACGAAUCGU
	F1 (w/o primers)	RNA	ivt	GGAAAAGACAAAUCUGCCCUCAGAGCUUGAGAACAUC UUC
	F2 (w/o primers)	RNA	ivt	GGAUGCAGGGGAGGCAGCCCCCGGUGGCGC
	F3 (w/o primers)	RNA	ivt	GCGCCAACGUUCUCAACAAUAGUGA
Extension primer	P1a	RNA	syn	hexynyl-$(PEG)_4$-CGCUUAUACGUC-PEG-biotin-PEG-pcl-PEG-FAM-CACUCCACAC (hammerhead selection)
	P1b	RNA	syn	FAM-biotin-CACUCCACAC (hammerhead synthesis)
	P2	RNA	syn	FAM-biotin-GCGGAUUUAGCUCAG (tRNA synthesis)
	P3	RNA	syn	FAM-biotin-GGAAAAGACAAAUCUGCCCU (class I ligase synthesis)
	P4	RNA	syn	FAM-biotin-GGAUGCUACAUG (F1 synthesis)
	P5	RNA	syn	FAM-biotin-GGUCGAAUGAUC (F2 synthesis)
	P6	RNA	syn	FAM-biotin-GGCAUAUCCAGC (F3 synthesis)
	P7	RNA	syn	FAM-biotin-GACACGAAUGUC (F1′ synthesis)
	P8	RNA	syn	FAM-biotin-ACCAACUAUUGC (F2′ synthesis)
	P9	RNA	syn	FAM-biotin-ACGAUUCGUACG (F3′ synthesis)

Template	Tem1	RNA	ivt	GACAAUGACAAAAAACGCUUUUCGGCCUUUCGGCCUC AUCAGUACGUCGUGUGGAGUG (hammerhead synthesis)
	Tem2	RNA	ivt	GACAAUGACAAAAAAUGGUGCGAAUUCUGUGGAUCG AACACAGGACCUCCAGAUCUUCAGUCUGGCGCUCUCC CAACUGAGCUAAAUCCGC (tRNA synthesis)
	Tem3	RNA	ivt	GACAAUGACAAAAAAUCACUAUUGUUGAGAACGUUG GCGUUAAAGCCACCGGGGGCUGCCUCCCCUGCAUCCG AAGAUGUUCUCAAGCUCUGAGGGCAGAUUUGUCUUU UCC (class I ligase synthesis)
	Tem4	RNA	ivt	GACAAUGACAAAAAAGACACGAAUGUCGAAGAUGUU CUCAAGCUCUGAGGGCAGAUUUGTCUUUUCCCAUGUA GCAUCC (F1 synthesis)
	Tem5	RNA	ivt	GACAAUGACAAAAAAACCAACUAUUGCGCGCCACCGG GGGCUGCCUCCCCUGCAUCCGAUCAUUCGACC (F2 synthesis)
	Tem6	RNA	ivt	GACAAUGACAAAAAAACGAUUCGUACGUCACUAUUG UUGAGAACGUUGGCGCGCUGGAUAUGCC (F3 synthesis)
	Tem7	RNA	ivt	GACAAUGACAAAAAAGGAUGCUACAUGGGAAAAGAC AAAUCUGCCCUCAGAGCUUGAGAACAUCUUCGACAUU CGUGUC (F1′ synthesis)
	Tem8	RNA	ivt	GACAAUGACAAAAAAGGUCGAAUGAUCGGAUGCAGG GGAGGCAGCCCCCGGUGGCGCGCAAUAGUUGGU (F2′ synthesis)
	Tem9	RNA	ivt	GACAAUGACAAAAAAGGCAUAUCCAGCGCGCCAACGU UCUCAACAAUAGUGACGUACGAAUCGU (F3′ synthesis)
Substrate	S1	RNA	syn	Cy5-CGCUUAUACGUC-FAM (hammerhead reaction)
	S2	R/DNA	syn	Cy5-d(AAA)-r(CCAGUC) (ligase reaction)
	S3	RNA	syn	pppGGAACAUUAUACGACUGGCACCAU (ligase reaction)
Sequencing	Fwd2	DNA	com	GGAGCAAAACGAGG
	Fwd3	DNA	com	GGGGGGATGCTACATG
	Fwd4	DNA	com	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGCTACA GGGCACTCCACAC
	Rev2	DNA	com	ATTGATGGTGCCTACAG
	Rev3	DNA	com	GCCTACAGTTTGACACGAATGTC
	Rev4	DNA	com	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGATTG ATGGTGCCTACAG

The molecules were synthesized in-house (syn), purchased from IDT (com), or prepared by in vitro transcription (ivt). The T7 RNA polymerase promoter sequence is underlined. Sequences in red indicate the 5′ tag used on polymerase ribozymes and templates to improve processivity. Sequences in blue are primer binding sites on templates used for polymerase extension assays. PEG, polyethylene glycol spacer; pcl, photocleavable linker; FAM, 6-fluorescein label; Cy5, cyanine 5-methine label, ppp, 5′-triphosphate added by chemical synthesis.

Figure S1

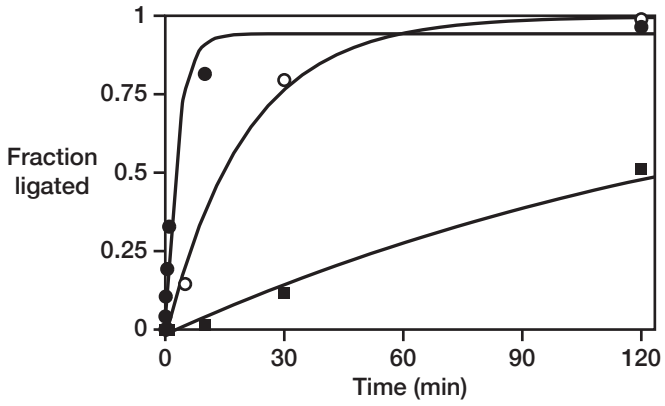
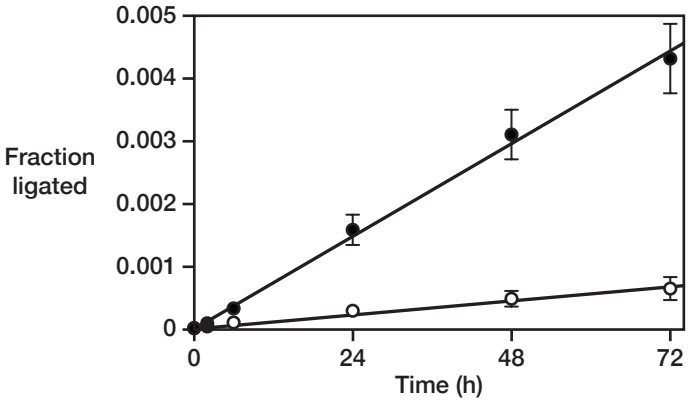


Figure S2



Java script ParseZymeReads

```
public class ParseZymeReads {

	private char[] template;
	private int[] map;
	private int[][] counts;
	private int end;
	private int length = -1;

	public ParseZymeReads(String[] args) {
		SuperScanner ss = new SuperScanner(args[0]);
		template = ss.getLine().split("  >  ")[0].toCharArray();
		ss.getLine(); //skip the alignment line
		end=Integer.parseInt(args[1]);

		if(args.length > 2)
			length=Integer.parseInt(args[2])-1;
		//0=match 1=mismatch A, 2=mismatch C, 3=mismatch G, 4=mismatch T, 5=deletion 6=insertion
7=insertion1 8=insertion2 ...26=insertion20
		counts=new int[end][27];
		map = new int[template.length];
		int index = 0;
		map[0]=0;
		for(int i = 1; i < template.length; i++)
		{
			if(template[i]=='-')
				map[i]=index;
			else
			{
				index++;
				map[i]=index;
			}
		}
		int count =0;
		while(ss.hasMore())
		{
			String r = ss.getLine().split("  >  ")[0];
			processRead(r);
			count++;
			if(count%100000==0){
				System.err.println(count);
			}
		}
		//now let's output the results
		String header = "Position";
		if(length>-1)
			header=length+"Position";
		System.out.println(header+"\tMatches\tMismatch A\tMismatch C\tMismatch G\tMismatch
T\tDeletions\tInsertions\tIns1\tIns2\tIns3\tIns4\tIns5\tIns6\tIns7\tIns8\tIns9\tIns10\tIns11\tIns12\tIns13\tIn
s14\tIns15\tIns16\tIns17\tIns18\tIns19\tIns20");
		for(int i = 0; i < counts.length; i++)
		{
```

```
            System.out.print(i+1);
            for (int j = 1; j < counts[i].length; j++)
            {
                System.out.print("\t"+counts[i][j-1]);
            }
            System.out.println();
        }
    }

    /**
     * Tabulates the position-specific stats of the read. If length is set, then read must have that length to
be counted.
     * @param read
     */
    private void processRead(String read)
    {
        read=formatLeadingWhiteSpace(read);
        char[] c = read.toCharArray();
        if(length > -1)
        {
            int readLength= end;
            if(read.indexOf(' ') < map.length && read.indexOf(' ')>-1)
                readLength=map[read.indexOf(' ')-1];
            if(readLength != length)
                return;
        }
        int insertSize=0;
        for(int i = 0 ; i < Math.min(c.length,template.length); i++)
        {
            char t = template[i];
             char r = c[i];
            if(r == ' '){
                break; // the read ended so let's stop counting.
            }
            if(c[i]!='-')
            {
                if(t == r){
                    counts[map[i]][0]++; //match
                    if(insertSize>0)
                    {
                        counts[map[i-1]][6]++;
                        counts[map[i-1]][insertSize+6]++;
                        insertSize=0;
                    }
                }
                else
                {
                    if(t=='-') //insertion
                    {
                        insertSize++;
                    }
                    else
                    {
                        if(r=='A')
```

```
                        counts[map[i]][1]++;//mismatch A
                    else if(r=='C')
                        counts[map[i]][2]++;//mismatch C
                    else if(r=='G')
                        counts[map[i]][3]++;//mismatch G
                    else if(r=='T')
                        counts[map[i]][4]++;//mismatch T
                    if(insertSize>0)
                    {
                        counts[map[i-1]][6]++;
                        counts[map[i-1]][insertSize+6]++;
                        insertSize=0;
                    }
                }
            }
        }
        else //a dash in the read
        {
            if(t != r)//deletion
            {
                counts[map[i]][5]++;
                if(insertSize>0)  //an insert that ends with a deletion
                {
                    counts[map[i-1]][6]++;
                    counts[map[i-1]][insertSize+6]++;
                    insertSize=0;
                }
            }
        }
    }
}

/**
 * Preprocess the reads to remove leading whitespace
 * @param read
 * @return
 */
private String formatLeadingWhiteSpace(String read) {
    // TODO Auto-generated method stub
    char[] chars = read.toCharArray();
    for(int i = 0; i < chars.length; i++)
    {
        if(chars[i]==' ')
            chars[i]='-';
        else break;
    }
    return new String(chars);
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    if(args.length > 1)
        new ParseZymeReads(args);
    else
```

```
        {
            System.err.println("Usage: ParseZymeReads breseqBamToAln TemplateLength [length of
reads to quantify]");
            System.err.println("Will create a tab-separated table of matches, mismatches, insertions,
deletions, and insertion lengths for each position of the template.\n"
                    + "If a third argument is provided, it will only tabulate reads of the specified length.
Output is to standard output.");
        }
    }

}
```

Java script SuperScanner

```
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.zip.GZIPInputStream;

public class SuperScanner {

    BufferedReader s = null;

    public SuperScanner(String file)
    {
        try{
            InputStream fileStream = new FileInputStream(file);
            if(file.endsWith(".gz"))
                s = new BufferedReader(new InputStreamReader(new GZIPInputStream(fileStream)));
            else
                s = new BufferedReader(new InputStreamReader(fileStream));
        }catch(IOException e)
        {
            e.printStackTrace();
        }
    }

    public void close()
    {
        try {
            s.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public boolean hasMore()
    {
        try {
```

```
            return s.ready();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return false;
    }

    public String getLine()
    {
        try {
                return s.readLine();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            return null;
        }
    }
}
```