PNAS
www.pnas.org

Supplementary Information for

Title: ERVmap analysis reveals genome-wide transcription of human endogenous retroviruses

Authors: Maria Tokuyama[1], Yong Kong[1], Eric Song[1], Teshika Jayewickreme[1], Insoo Kang[2], and Akiko Iwasaki[1,3]*

Author Affiliations:

[1]Department of Immunobiology, Yale School of Medicine, New Haven, CT 06520, USA.

[2]Department of Internal Medicine, Yale University School of Medicine, New Haven, CT 06520, USA

[3]Howard Hughes Medical Institute, Chevy Chase, MD 20815, USA.

*Corresponding Author: Akiko Iwasaki, Howard Hughes Medical Institute, Department of Immunobiology, Yale School of Medicine, 300 Cedar Street, New Haven, CT 06519, USA, Tel: 203.785.2919. akiko.iwasaki@yale.edu.

This PDF file includes:

Supplementary Information Text

Heading

Subhead. Materials

ChIP sequencing datasets:

The following big.wig files were obtained through ENCODE:

H3K4me3			
GM12878	ENCFF003DXG	ENCFF583MDQ	ENCFF776DPQ
A549	ENCFF152LRB		
HelaS3	ENCFF489CIY	ENCFF913SUG	
MCF7	ENCFF530LJW	ENCFF797IUA	
K562	ENCFF525ZRM	ENCFF712XRE	ENCFF847JMY
HepG2	ENCFF736LHE	ENCFF746CXV	ENCFF777EVS
SKNSH	ENCFF208OZM	ENCFF555ZEF	
H3K27Ac			
GM12878	ENCFF340JIF		
Hela S3	ENCFF194XTD		
MCF-7	ENCFF515VXR	ENCFF986ZEW	
K562	ENCFF779QTH		
HepG2	ENCFF764VYK		
SK-N-SH	ENCFF151FXB	ENCFF143WKK	
H3K9me3			
HelaS2	ENCFF891XLY		
MCF7	ENCFF191LDZ	ENCFF754TEC	
K562	ENCFF812HRW		
HepG2	ENCFF485BCI		
SK-N-SH	ENCFF919ISB	ENCFF932EUO	
H3K27me3			

GM12878	ENCFF039JOT	ENCFF313LYI	
HelaS3	ENCFF614HNF		
MCF7	ENCFF495QYP	ENCFF081UQC	
K562	ENCFF928NWQ	ENCFF914VFE	
HepG2	ENCFF419FUZ	ENCFF598TWA	
SK-N-SH	ENCFF224TVD		

All of the data we chose are ChIP-sequencing files that contained fold over control for 2 replicates. The sequencing files were uploaded onto IGV software for visualization of tracks at each ERV locus.

RT-qPCR primers:

Primers used to amplify ERVs in cell lines are as follows:

Name	Sequence (5' -> 3')
Fwd:1114	GTG ATG AAG TTG TTC CTG TTT AG
Rev:1114	CAT ACA GTT GAG TGG GAG TAA A
Fwd:4627	GGG AAC TGA AGC CTG TTA TC
Rev:4627	TTC TGG ATG GGT AGG ATA GG
Fwd:ERVK3-1	CTG CCA TTG TCC TCT TCT TAT
Rev:ERVK3-1	TCC TCC CTC TCC TGT TTA TC
Fwd:3409	CAG AAG CGG GTA GAT GAA ATA G
Rev:3409	CTT ATG TAG CCC TCC CAA ATC
Fwd:4452	GAG CAA TCC CGC CAA TAA
Rev:4452	GAC TGC CCA AAG GAA TCT AC
Fwd:5346	GAC AGG CAG TAA ATA CCC ATT A
Rev:5346	GGA GTA GGA CAG AGT GTA GAA
Fwd:4585	TCC TCA ATA CCT GCC TCT ATA A
Rev:4585	GGA TGA AGG GTG CAA AGA A
Fwd:ERVH13q33.3	CCA AAC TGC CAC TCT TAA CT
Rev:ERVH13q33.3	GGG AGG ATC TAG GAC ATC TAA T

Fwd:4503	TCA GTC AGC CCT TGT TTA TTC
Rev:4503	CAA GTG CCA CTC CAG TTA TT
Fwd:4700	CCT TCC ATC ACT CTT CCT AAT G
Rev:4700	GAC CCT GAA ATA GGT GGA TAA A
Fwd:2187	GAG CCA GGG TAA ATG TCT TC
Rev:2187	GAG CAT TCC AGC TTC GTA TTA
Fwd:4059	CCC ATG GAA AGA GGT GTA TTT
Rev:4059	GGG CCT TTA AGA GGT GAT TAG
Fwd:2003	CCA GTC AGG TAT AGT ACG AGA T
Rev:2003	CAA CTC CAG AGG TTG GTA TAA G
Fwd:5361	GAG ACC AAA GTG CCG ATA AG
Rev:5361	GGA GTC TGG ACA TCT GGA ATA
Fwd:4699	CCT TCC ATC ACT CTT CCT AAT G
Rev:4699	GAC CCT GAA ATA GGT GGA TAA A
Fwd:1043	CCA ACT GTC CCT AAC CCT TAA A
Rev:1043	GGG TCA GAG GAA TAG CAA AGA A
Fwd:2744	GAC CAG ACC GTT AGA AGA ATA AG
Rev:2744	GCC CTT GGT TTC CTG AAT AA
Fwd:ERV3-1	CTA CTA CCA GTG CCT GAT TTA C
Rev:ERV3-1	CGC AGG AGG AAG AGA AAT TAT
Fwd:768	GAC ATG GGT AGT GAG GAT GAA AG
Rev:768	GAG GCA GAA ATG GGC ATA AGA
Fwd:3340	CTG GGT TCT GTG TCT CTT TG
Rev:3340	GAC ATC AGA TGA GCA GGA TAA G

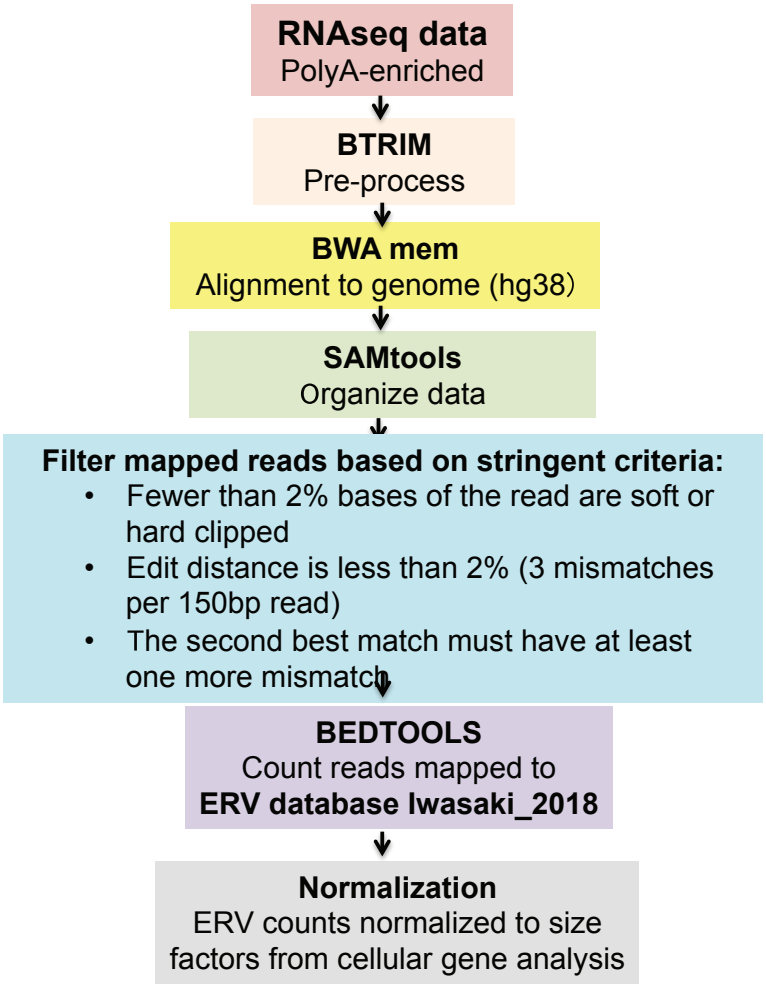


Fig. S1. Flowchart of ERVmap. ERVmap was developed by first compiling a list of proviral ERVs into a database of 3220 ERVs. Then a computational pipeline was developed to filter mapped reads using stringent criteria in order to increase accuracy in reads mapped to repetitive sequences. Reads that were mapped to ERVs in our database were extracted and normalized to size factors obtained from cellular gene analysis.

```
#!/usr/bin/env perl

#
# erv: map to genome, for SE data
#

use warnings;
use strict;
use File::Basename;
use Cwd;
use Getopt::Long;
use File::Type;

my $home = "~/scripts";


my $cell = "sample";          # sample name
my $workdir = ".";           # working dir "$cell"
my $outdir = "tmp";           # dir under $cell

my $stage  = 0;
my $stage2 = 0;

my $fastq;            # sequence file

my $length = 40;
my $score_offset = 33;
my $score = 25;
my $pat = "$~/bin/trim/illumina_adapter.txt";
my $cat = '';


my $test = 0;

print map("\t$_", @ARGV), "\n";


GetOptions ("test"      => \$test,
            "cell=s"    => \$cell,
            "workdir=s" => \$workdir,
            "outdir=s"  => \$outdir,

            "stage=i"   => \$stage,
            "stage2=i"  => \$stage2,

            "length=i"  => \$length,
```

```
            "score_offset=i"  => \$score_offset,
            "pat=s"     => \$pat,
            "cat"       => \$cat,
            "score=i"   => \$score,

            "fastq=s"  => \$fastq,   ### absolute path

            );
die unless ($fastq && $stage > 0 && $stage2 > 0);


my $genome = "~/genome/genome";


### check if fastq file is zipped
my $zipped = 0;
my $ftype = File::Type->new->checktype_filename($fastq);
print STDERR "$fastq: $ftype\n";
if ($ftype =~ /zip/) {
   $zipped = 1;
}


### btrim
my $btrim = "~/bin/trim/btrim/btrim3";
my $adaptor = "~/bin/trim/illumina_adapter.txt";
my $btrimout = "btrim_g_se.out";
my $btrim_cmd;
if ($score_offset == 33) {
   if ($zipped) {
         $btrim_cmd = "/bin/bash -c '$btrim -l $length -w 10 -a 25 -p $adaptor -3 -P -o
$btrimout -t <(gunzip -c $fastq) -C > btrim.log 2> btrim.log'";
   } else {
         $btrim_cmd = "/bin/bash -c '$btrim -l $length -w 10 -a 25 -p $adaptor -3 -P -o
$btrimout -t <(cat $fastq) -C > btrim.log 2> btrim.log'";
   }
} else {
   if ($zipped) {
         $btrim_cmd = "/bin/bash -c '$btrim -i -l $length -w 10 -a 25 -p $adaptor -3 -P -o
$btrimout -t <(gunzip -c $fastq) -C > btrim.log 2> btrim.log'";
   } else {
         $btrim_cmd = "/bin/bash -c '$btrim -i -l $length -w 10 -a 25 -p $adaptor -3 -P -o
$btrimout -t <(cat $fastq) -C > btrim.log 2> btrim.log'";
   }
}
```

```
### bwa
my $bwa = "bra";
my $samtools = "samtools";
my $filter = "$~/scripts/parse_bam.pl";
my $bwabam = "bwa.bam";
#my $bwa_cmd = "/bin/bash -c '$bwa mem -t 8 -p $genome ${btrimout} | $samtools
view -Sh -F4 - | tee >($samtools view -Shb - | $samtools sort - -o bwa_unfiltered.bam) |
$filter | $samtools view -bSh - > $bwabam'";
my $bwa_cmd = "/bin/bash -c '$bwa mem -t 8 -p $genome ${btrimout} | $samtools view
-Sh -F4 - | $filter | $samtools view -bSh - > $bwabam'";

### sort
my $bwabam_sorted = "bwa_sorted.bam";
my $sort_cmd = "$samtools sort $bwabam -o $bwabam_sorted";

# index
my $bamindex_cmd = "$samtools index ${bwabam_sorted}";

### count
my $bedtools = "~/bin/bedtools";
my $bed = "~/genome/ERVmap_v2_all_sorted.bed";
my $genomefile = "~/genome/GRCh38.genome_file.txt";
my $cntfile = "herv_coverage_GRCh38_g_pe2.txt";
my $count_cmd = "$bedtools coverage -b ${bwabam_sorted} -a $bed -counts -sorted -g
$genomefile > $cntfile";

chdir($workdir);
my $wdir = "$cell";
my $cmd = "mkdir -p $wdir";
&cmd($cmd);
chdir($wdir);


# btrim
if ($stage <= 1 && $stage2 >= 1) {
   &cmd($btrim_cmd);
}

# bwa
if ($stage <= 2 && $stage2 >= 2) {
   &cmd($bwa_cmd);
   unlink($btrimout) if (-s $bwabam > 10);
}


# sort
```

```
if ($stage <= 3 && $stage2 >= 3) {
   &cmd($sort_cmd);
   &cmd($bamindex_cmd);
   unlink($bwabam);
}

# count
if ($stage <= 4 && $stage2 >= 4) {
   &cmd($count_cmd);
}


sub cmd {
   my ($c, ) = @_;

   print "In ", cwd(), ":\n";
   print "$c\n";
   system($c) unless ($test);
}
```

```
#!/usr/bin/env perl

#
# parse bam from bwa
#

use strict;
use warnings;

use Getopt::Long;
my $test;
my $fix;
GetOptions ("test" => \$test,
            "fix" => \$fix,
            );

while (my $li = <>) {
   chomp($li);
   my @t = split(/\t/, $li);
#    print "$t[2]\t$t[10]\n";

   if ($li =~ /^\@/) {
         print "$li\n" ;
         next;
```

```
    }

    next if (soft_clipping(\@t));
    my $seq = $t[9];
    my $seql = length($seq);
    #print "$seq\n";

    my ($nm, $as, $xs, ) = (1000, 0, 0,);

    for (my $i=10; $i<@t; $i++) {
        if ($t[$i] =~ /^AS:i:(\d+)/) {       # Alignment score
            $as = $1;
        } elsif ($t[$i] =~ /^XS:i:(\d+)/) {  # Suboptimal alignment score
            $xs = $1;
        } elsif ($t[$i] =~ /^NM:i:(\d+)/) {
            $nm = $1;
        }
    }

    print join("\t",
               "NM-AS-XS", $nm, $as, $xs, $as - $xs), "\n" if ($test);

    if ($fix) {
        if ($nm < 3 && ($as - $xs >= 5)) {
            print "$li\n";
        }
    } else {
        my $nmperc = $nm / $seql;
        #my $asxsperc = ($as - $xs) / $seql;
        if ($nmperc < 0.02 && ($as - $xs >= 5)) {
            print "$li\n";
        }
    }
}

sub soft_clipping {
    my ($t, ) = @_;

    my $c = $t->[5];
    my @l = split(/[0-9]+/, $c);
    shift(@l);  # 1st is empty
    my @d = split(/[a-zA-Z]+/, $c);
    die "$#l != $#d" if (scalar @l != scalar @d);

    if ($test) {
        print "\t$c\n";
```

```
        print "\t->\t", map(">$_< ", @d), "\n";
        print "\t=>\t", map(">$_< ", @l), "\n";
    }

    my ($tot, $s, $h, ) = (0, 0, 0,);
    for (my $i=0; $i<@l; $i++) {
        $tot += $d[$i];
        $h += $d[$i] if ($l[$i] eq "H");
        $s += $d[$i] if ($l[$i] eq "S");
    }
    print join("\t", "\t",  $tot, $s, $h, ),"\n" if ($test);

    my $perc = ($h+$s) / $tot;

    my $r;
    if ($fix) {
        $r = ($h+$s>=3) ? 1 : 0;
    } else {
        $r = ($perc>=0.02) ? 1 : 0;
    }
    return $r;
}
```

Fig. S2. ERVmap core scripts.

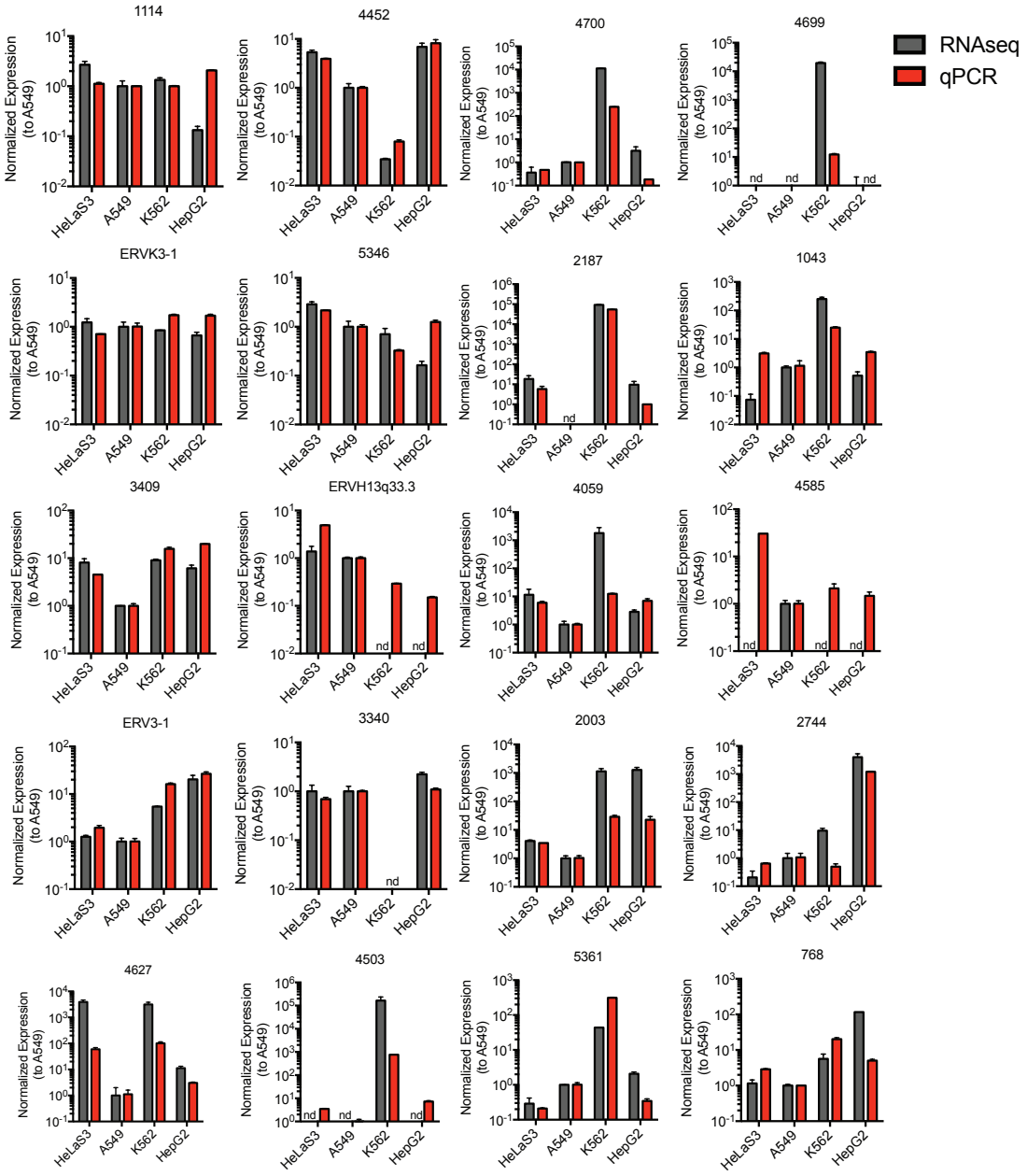
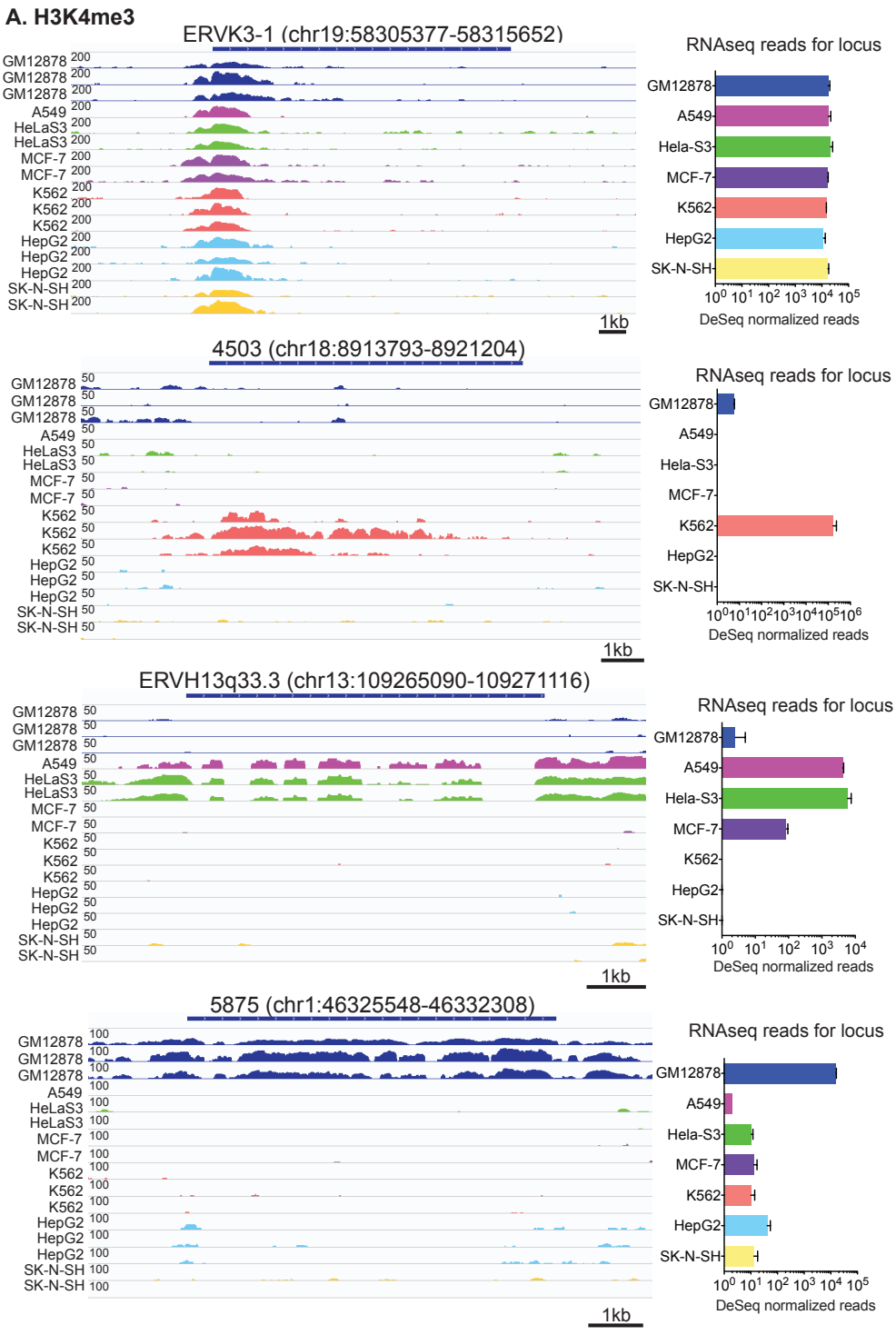


Figure S3. Confirmation of RNA sequencing data using qPCR. 20 ERVs that were either highly expressed by all cell types or showed cell type-specific expression patterns were confirmed by qPCR for the indicated cell types. The graphs show SEM of technical replicates. nd, not detected.


A. H3K4me3
ERVK3-1 (chr19:58305377-58315652)
RNAseq reads for locus
GM12878
A549
HeLaS3
HeLaS3
MCF-7
MCF-7
K562
K562
K562
HepG2
HepG2
HepG2
SK-N-SH
SK-N-SH
200
1kb
Hela-S3
DeSeq normalized reads
4503 (chr18:8913793-8921204)
50
ERVH13q33.3 (chr13:109265090-109271116)
5875 (chr1:46325548-46332308)
100

B. H3K27Ac


ERVK3-1 (chr19:58305377-58315652)
GM12878
HeLaS3
MCF-7
MCF-7
K562
HepG2
SK-N-SH
SK-N-SH
RNAseq reads for locus
Hela-S3
DeSeq normalized reads
4503 (chr18:8913793-8921204)
ERVH13q33.3 (chr13:109265090-109271116)
5875 (chr1:46325548-46332308)


C. H3K9me3
ERVK3-1 (chr19:58305377-58315652)
HeLaS3
MCF-7
MCF-7
K562
HepG2
SK-N-SH
SK-N-SH
RNAseq reads for locus
Hela-S3
MCF-7
K562
HepG2
SK-N-SH
DeSeq normalized reads
4503 (chr18:8913793-8921204)
RNAseq reads for locus
DeSeq normalized reads
ERVH13q33.3 (chr13:109265090-109271116)
RNAseq reads for locus
DeSeq normalized reads
5875 (chr1:46325548-46332308)
RNAseq reads for locus
DeSeq normalized reads

Figure S4. Histone modifications at highly transcribed ERV loci. ChIP-seq data for each cell line were obtained through ENCODE, and IGV software was used to visualize the tracks for H3K4me3 **(A)**, H3K27Ac **(B)**, H3K9me3 **(C)**, and H3K27me3 **(D)** at ERVK3-1, 4503, ERVH13q33.3, and 5875 loci. The level of ERV expression for each ERV per cell type is depicted as a bar graph.

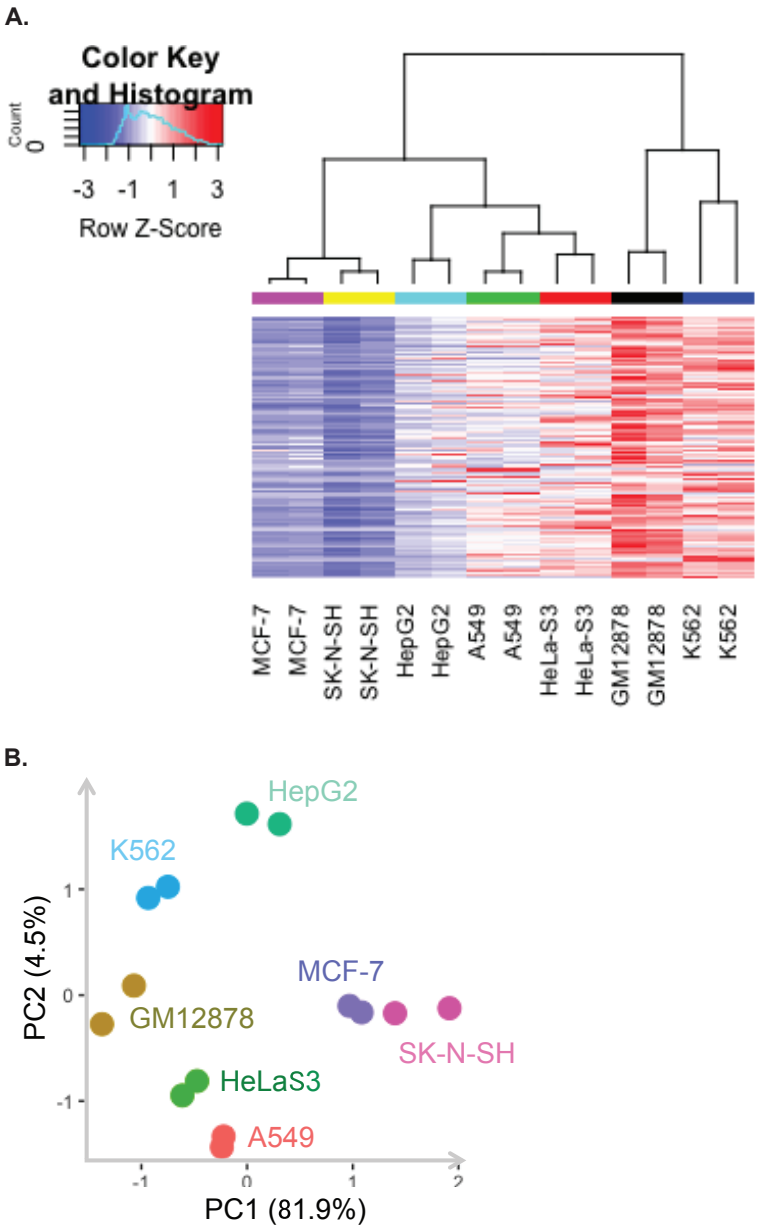


Figure S5. RepEnrich analysis of ERV expression in cell lines. A) Heatmap of all RepEnrich reads for 550 LTR families for each cell line. B) Principle component analysis using the same set of data as (A).

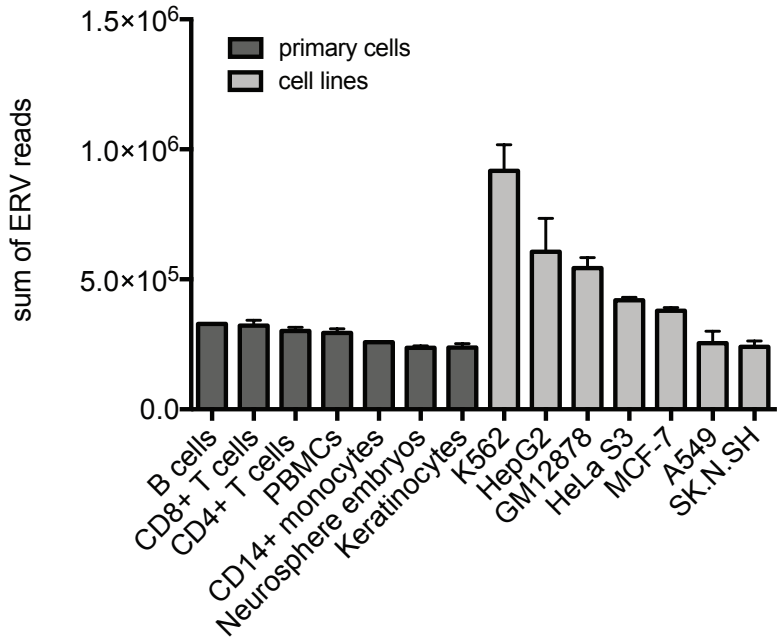


Figure S6. ERV reads compared across all cell types. The sum of all ERV reads for all of the cell line and primary cell samples were plotted on the same graph. Cell types with multiple datasets were averaged. Error bars represent SEM.

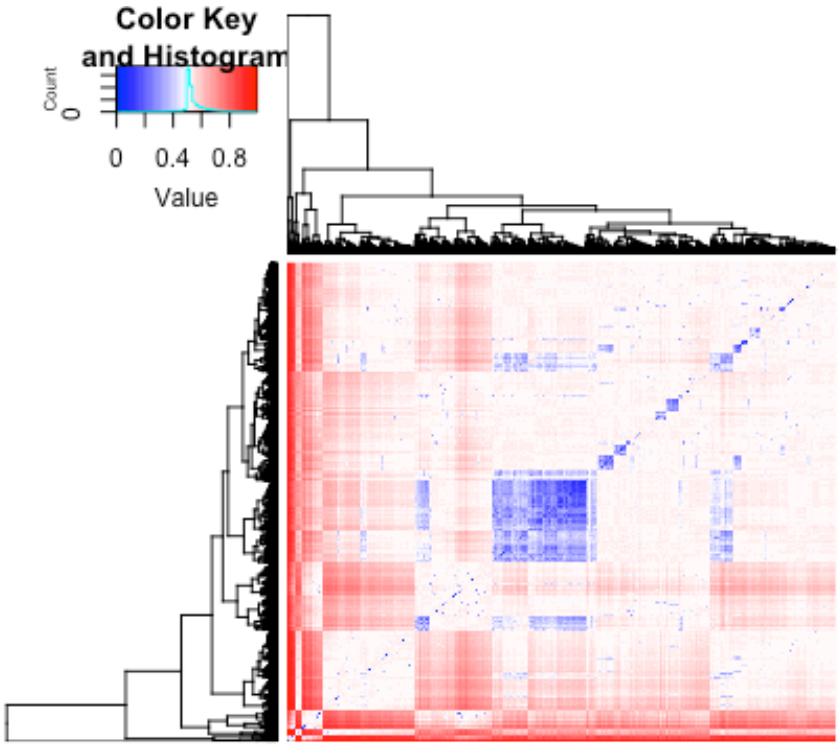


Figure S7. Distance matrix of ERVs in ERVmap database. Pair-wise edit distance (Levenshtein distance) between all 3220 ERVs in ERVmap were calculated based on their DNA sequences. Heatmap was generated using the edit distance normalized to the length of each ERV locus. 0 (blue) corresponds to identical sequence pairs and 1 (red) corresponds to no matching sequence between sequence pairs.

Table S1. Summary of LTR elements in Repeatmasker database

Repeatmasker database, hg38

	# LTR elements
Family	10
Subfamily	595
Total # of copies	**742621**

Table S2: Demographic information for SLE patients and healthy donors

SLE Donor	Age	Sex	Race
1	43	female	White
2	60	female	White
3	29	female	Hispanic
4	36	female	Black
5	52	female	Black
6	48	female	Hispanic
7	41	female	Hispanic
8	47	female	Black
9	39	female	Hispanic
10	26	female	Black
11	49	female	Black
12	48	female	Hispanic
13	39	female	Black
14	29	female	Hispanic
15	28	female	Black
16	50	female	Black
17	50	female	Black
18	27	female	Hispanic
19	40	female	Black
20	33	female	Hispanic

Healthy Donor	Age	Sex	Race
1	29	female	Black
2	31	female	White
3	34	female	Hispanic
4	41	female	Hispanic
5	22	female	White
6	68	female	White

Table S3: Comparison of ERV databases

ERV annotation	Author/Institution	Description	Number of entries	Ref.
Repbase	Genetic Information Research Institute	-Database of consensus sequences for repetitive or mobile DNA elements for eukaryotic species. -No genomic location associated with sequences. -Largely non-autonomous LTR elements for ERVs.	-48860 total entries -333 human ERV sequences	(1)
Repeatmasker	Institute for Systems Biology	-Program that annotates repetitive elements in the genome based on Repbase and Dfam databases. -Provides exact genomic location. -Mostly non-autonomous LTR elements.	-771,683 entries for hg38	(2)
DFAM	University of Montana Institute for Systems Biology	-Database of repetitive DNA elements for eukarytic genomes. -Mostly non-autonomous LTR elements.	-4150 total entries -540 human LTR elements	(3)
HERVd	Ondrej Moravcik Jan Paces Dan Elleder Institute of Molecular Genetics of the ASCR, Czech Republic	-Interactive database for human repetitive and mobile DNA elements. -Based on Repbase/Repeatmasker annotations. -Mostly non-autonomous LTR elements.	-5,317,386 total entries -725,763 ERV entries	(4)
HERVgDB4	Francois Mallet Hospice Civils de Lyon bioMerieux Centre Hospitalier Lyon Sud	-Database of ERVs, active LINE-1 elements, lncRNA, infectious ERVs and human genes used for the development of HERV-V3 hybridization assay. -ERVs annotated based on protype ERV sequences. -Also includes ERV elements annotated in Dfam. -No information regarding genomic location or sequence information on each ERV element provided.	-428,301 ERV elements	(5)

Reference viral database (RVDB)	Arifa S. Khan U.S. FDA	-Databases of viral, virus-related, virus-like nucleotide sequences. -Based on sequences in GenBank. -No genomic location for ERVs. -Many are partial, fragmented or ERV-like sequences. -Unclear how many are unique sequences.	-561,676 total entries -955 ERV sequences	(6)
Human LTR-retrotransposon genome browser	Andrew Garazha & Anton Buzdin Cellgenetics, Russia	-A web browser to visualize LTR elements that are bound by various transcription factors (TF) based on chromatin immunoprecipitation sequencing data on ENCODE. -LTR element annotation is based on Repbase/Repeatmasker annotations. -Mostly non-autonomous LTR elements.	-717,612 ERV/LTR elements	(7)

References:

1. Bao W, Kojima KK, Kohany O (2015) Repbase Update, a database of repetitive elements in eukaryotic genomes. *Mobile DNA*:1–6.
2. Smit AFA, Hubley R, Green P (2013) RepeatMasker Open-4.0. Available at www.repeatmasker.org. Accessed Nov. 13, 2018.
3. Hubley R, et al. (2016) The Dfam database of repetitive DNA families. *Nucleic Acids Research* 44(D1):D81–D89.
4. Paces J, Pavlícek A, Paces V (2002) HERVd: database of human endogenous retroviruses. *Nucleic Acids Research* 30(1):205–206.
5. Becker J, et al. (2017) A comprehensive hybridization model allows whole HERV transcriptome profiling using high density microarray. 1–14.
6. Goodacre N, Aljanahi A, Nandakumar S, Mikailov M, Khan AS (2018) A reference viral database (RVDB) to enhance bioinformatics analysis of high-throughput sequencing for novel virus detection. mSphere 3:e00069-18.
7. Garazha A, et al. (2015) New bioinformatic tool for quick identification of functionally relevant endogenous retroviral inserts in human genome. *Cell Cycle* 14(9):1476–1484.