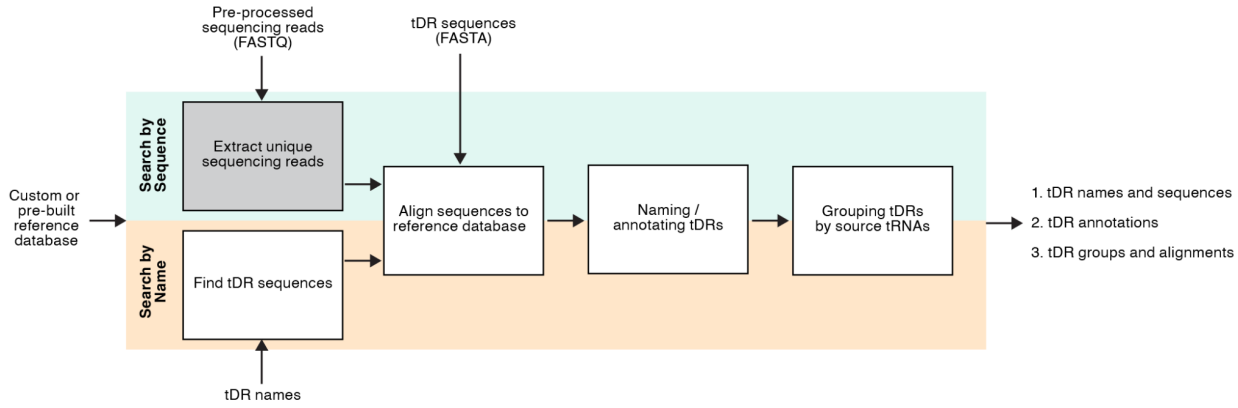
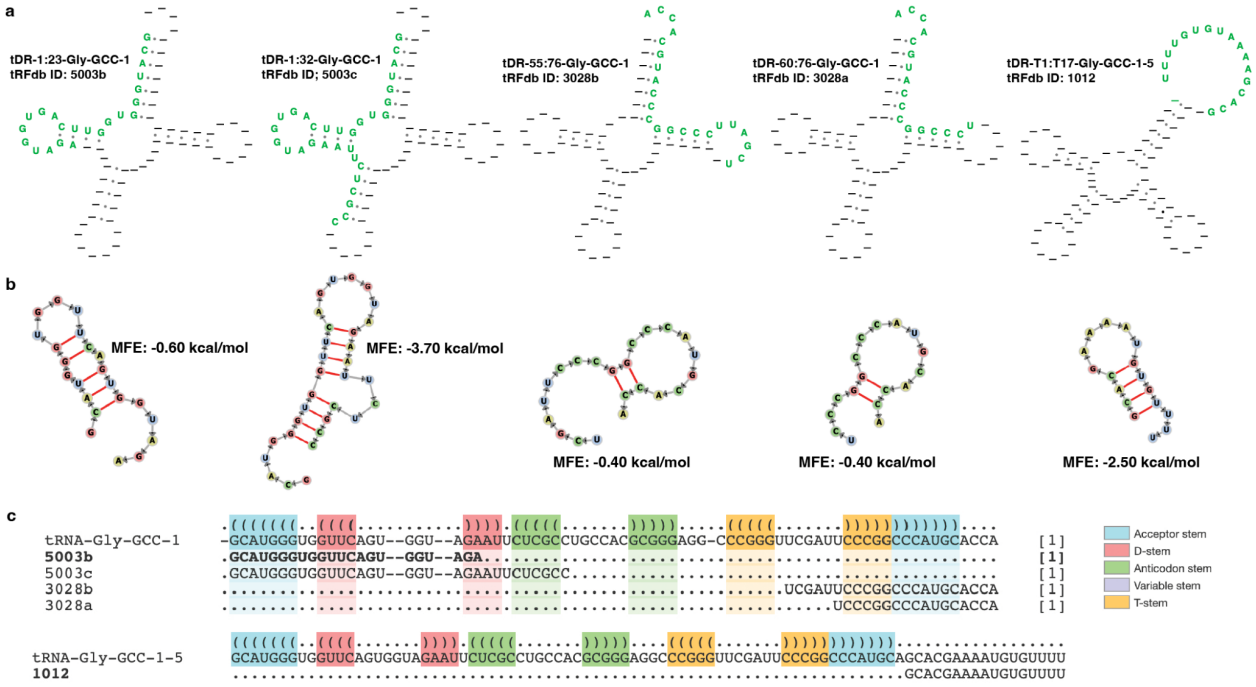
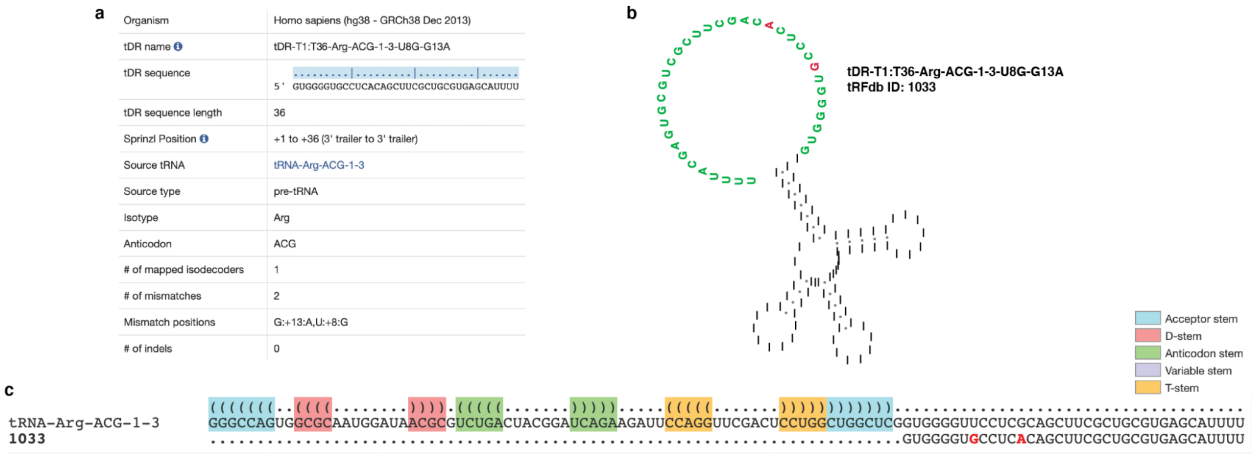
Supplementary Figures



Supplementary Fig. 1 Overview of tDRnamer workflow. Both the web server and downloadable software versions take sequences and generate tDR names according to the new naming convention (Fig. 1), create annotations, and perform group alignments of related sequences (green). Alternatively, the system can take tDRnamer-generated names, and provide the corresponding tDR sequences with annotations and alignments (orange). The gray box represents an extra processing step designed to handle RNA-seq data that may contain many identical sequences (only available in the downloaded software). See Supplementary Methods for full details.



Supplementary Fig. 2 A family of five tDRs derived from human tRNA-Gly-GCC-1. **a.** tDRs (two from 5′ end of the source tRNA, two from 3′ end, and one from 3′ trailer) are superimposed onto the secondary structure of tRNA-Gly-GCC-1. **b**. Minimum free energy secondary structures are predicted for tDRs. **c**. Four of the tDRs (tRFdb IDs: 5003b, 5003c, 3028a, and 3028b) are aligned with the mature tRNA-Gly-GCC-1 transcript. tDR-T1:T17-Gly-GCC-1-5 (tRFdb ID: 1012) is aligned separately to the precursor tRNA-Gly-GCC-1-5. All images are generated by the web server version of tDRnamer.



Supplementary Fig. 3 tDR-T1:T36-Arg-ACG-1-3-U8G-G13A with two single nucleotide variations from 3′ trailer of $tRNA^{Arg(ACG)}$. **a.** tDR details are shown in the screenshot of tDRnamer results. **b**. tDR is superimposed onto the secondary structure of tRNA-Arg-ACG-1-3. **c**. tDR is aligned with the source tRNA-Arg-ACG-1-3 gene, showing known substitution mutations (dbSNP IDs: rs4982711 and rs34026312) marked in red. All images are generated by the web server version of tDRnamer.

Supplementary Tables

Supplementary Table 1 tDRnamer feature comparison in standalone and web server versions. Both versions employ the same processing routines for naming tDR sequences and searching tDR sequences by name. Feature differences in usage are listed.

Features	**Standalone version**	**Web server version**
Use environment	Command-line in Linux/Unix platform	Web browser
Reference database	Download pre-built databases for model organisms or build user-defined databases with create_tDRnamer_db	Use available pre-built databases for model organisms (can request new databases to be added)
Input for tDR naming	Pre-processed small RNA sequencing reads in FASTQ file or tDR sequences in FASTA file	tDR sequences in FASTA format (maximum 1,000 with default search mode and 100 with maximum sensitivity search mode)
Input for tDR sequence search	Text file with tDR names	tDR names (maximum 500 with default search mode and 50 with maximum sensitivity search mode)
tDR sequence output	FASTA file	FASTA file and web page visualization
tDR naming and annotation output	Tab-delimited file	Tab-delimited file and web page visualization with links to source tRNAs in GtRNAdb
tDR search mode	Default or maximum sensitivity (--max)	Default or maximum sensitivity
tDR alignment output	Stockholm files	Image visualization and downloadable files (PNG and postscript)
tDR secondary structure prediction	Not available	Minimum free energy (MFE) secondary structure image visualization and downloadable files (PNG and postscript)
tDR group output	Text file with alignments	Text file with alignments and web page visualization with links to tDR group members and source tRNAs in GtRNAdb

Supplementary Table 2 Re-annotation and comparison of tDR names derived from human $tRNA^{Pro}$ in tRFdb. Prefixes of License Plates in parentheses were assigned automatically when using the web server tool and depend on the existence of tDRs in MINTbase.

tRFdb ID	tDR position	tDR names and synonyms by tDRnamer	License Plates nomenclature
5011a	1 to 15	tDR-1:15-Pro-AGG-1-M8 tDR-1:15-Pro-AGG-2-M8 tDR-1:15-Pro-AGG-3-M8 tDR-1:15-Pro-CGG-1-M8 tDR-1:15-Pro-CGG-2-M8 tDR-1:15-Pro-TGG-1-M8 tDR-1:15-Pro-TGG-2-M8 tDR-1:15-Pro-TGG-3-M8	(nlr-)15-6978WP
5012a	1 to 18	tDR-1:18-Pro-AGG-1-M7 tDR-1:18-Pro-AGG-2-M7 tDR-1:18-Pro-AGG-3-M7 tDR-1:18-Pro-CGG-1-M7 tDR-1:18-Pro-CGG-2-M7 tDR-1:18-Pro-TGG-2-M7 tDR-1:18-Pro-TGG-3-M7	(tRF-)17-6978WP4
5013b	1 to 22	tDR-1:22-Pro-AGG-1-M6 tDR-1:22-Pro-AGG-2-M6 tDR-1:22-Pro-CGG-1-M6 tDR-1:22-Pro-CGG-2-M6 tDR-1:22-Pro-TGG-2-M6 tDR-1:22-Pro-TGG-3-M6	(tRF-)21-6978WPRLE
5012b	1 to 24	tDR-1:24-Pro-AGG-1-M6 tDR-1:24-Pro-AGG-2-M6 tDR-1:24-Pro-CGG-1-M6 tDR-1:24-Pro-CGG-2-M6 tDR-1:24-Pro-TGG-2-M6 tDR-1:24-Pro-TGG-3-M6	(tRF-)23-6978WPRL0L
5013c	1 to 31	tDR-1:31-Pro-AGG-1-M5 tDR-1:31-Pro-AGG-2-M5 tDR-1:31-Pro-CGG-1-M5 tDR-1:31-Pro-CGG-2-M5 tDR-1:31-Pro-TGG-3-M5	(tRF-)30-6978WPRLXN4V
3002a	59 to 76	tDR-59:76-Pro-AGG-1-M8 tDR-59:76-Pro-AGG-2-M8 tDR-59:76-Pro-AGG-3-M8 tDR-59:76-Pro-CGG-1-M8 tDR-59:76-Pro-CGG-2-M8 tDR-59:76-Pro-TGG-1-M8	(tRF-)18-HR6HFRD2

tRFdb ID	tDR position	tDR names and synonyms by tDRnamer	License Plates nomenclature
		tDR-59:76-Pro-TGG-2-M8 tDR-59:76-Pro-TGG-3-M8	
3002b	55 to 76	tDR-55:76-Pro-AGG-1-M8 tDR-55:76-Pro-AGG-2-M8 tDR-55:76-Pro-AGG-3-M8 tDR-55:76-Pro-CGG-1-M8 tDR-55:76-Pro-CGG-2-M8 tDR-55:76-Pro-TGG-1-M8 tDR-55:76-Pro-TGG-2-M8 tDR-55:76-Pro-TGG-3-M8	(tRF-)22-8B8SOUPR2

Supplementary Table 3 Files included in tDRnamer reference database. “dbname” in the file name represents the database name set as input option.

File	Description
dbname-tRNAgenome.*	FASTA file of tRNA sequences with Bowtie2 indexes and BLAST database
dbname-trnaalign.stk	Alignments of mature tRNA sequences in Stockholm file format
dbname-trnaconvert.stk	Alignments of mature tRNA sequences in Stockholm file format
dbname-trnaloci.stk	Alignment of tRNA gene sequences in Stockholm file format
dbname-trnatable.txt	Tab-delimited file with tRNA transcripts and tRNA genes map
dbname-maturetRNAs.fa	FASTA file of mature tRNA sequences
dbname-maturetRNAs.bed	tRNA transcripts in BED file format
dbname-tRNAloci.fa	FASTA file of tRNA gene sequences
dbname-trnaloci.bed	tRNA genes in BED file format
dbname-filtered-tRNAs.out	tRNAscan-SE output file format with filtered tRNA genes used for database creation
dbname-dbinfo.txt	Database creation information
dbname-create_tDRnamer_db.log	Database creation log file

Supplementary Table 4 Output files generated by tDRnamer. “prefix” in the file name represents the output file prefix set by the --output option. “tDR” as prefix of file name represents the tDR name of the corresponding transcript.

Output File	**Description**
prefix-tDR.fa	FASTA file with tDR names and sequences
prefix-tDR-info.txt	Tab-delimited file with tDR annotations including tDR names and sequences, source tRNAs, Sprinzl positions of tDRs relative to source tRNAs, original tRNA isotype and anticodon, sequence variation counts, and group ID
prefix-tDR-groups.txt	Text file containing queried tDRs that are grouped together. Alignments of group members and source tRNAs are arranged in Stockholm format that includes primary sequence and secondary structure information.
prefix-found-seq.fa	FASTA file with tDR sequences when searching by tDR names. This file is generated during the initial round of sequence search before the annotation process.
prefix-tDRs.stk	Alignments of identified tDRs derived from mature tRNAs with reference tRNA sequences in Stockholm file format
prefix-pre-tDRs.stk	Alignments of identified tDRs derived from precursor tRNAs with reference tRNA sequences in Stockholm file format. This file is only generated when the sequence source is set as eukaryotes.
prefix-unique-seq.fa	FASTA file with unique sequences in provided FASTQ file. This is only generated when a FASTQ file is provided as input.
prefix-filtered-seq.fa	FASTA file with sequences after filtering by length constraint and error checking. This is only generated when naming tDRs.
prefix-reformatted-seq.fa	FASTA file with sequences after converting RNA sequences to DNA sequences if applicable
prefix-filtered-names.txt	List of tDR names after filtering by error checking. This is only generated when searching by tDR names.
prefix-tDR-list.txt	Intermediate file generated during tDR annotation process
prefix-clusters.txt	Intermediate file generated during tDR group process
prefix-pre-clusters.txt	Intermediate file generated during tDR group process. This file is only generated when the sequence source is set as eukaryotes.
prefix-tDRs.sam	Intermediate alignment file generated during tDR searching/annotation process with maximum sensitivity option

Output File	Description
prefix-find-tdrs.log	Intermediate log file for searching tDR sequences by names
prefix_tDRnamer.log	Log file of tDRnamer run
tDR-align.png	PNG file of tDR superimposed on source tRNA
tDR-align.ps	Postscript file of tDR superimposed on source tRNA
tDR-MFE.png	Secondary structure prediction of tDR based on minimum free energy in PNG file
tDR-MFE.eps	Secondary structure prediction of tDR based on minimum free energy in postscript file

Supplementary Methods

tDRnamer software design and requirements

tDRnamer is a software package that generates standardized transcript names and annotations for tRNA-derived RNAs (tDRs) and searches tDR sequences by name (Supplementary Fig. 1, Supplementary Table 1). The standalone version runs natively as a command-line tool within the Unix/Linux environment, and can also be used via a Docker container or Conda environment in a broader context. The software was developed using Perl and Python for the framework and data processing. The package also uses external bioinformatics tools and data sources as described below in more detail. Researchers who do not have the resources or expertise to work with a command-line tool can use the web server version, developed using HTML, Javascript, and CGI Perl for the execution of the tDRnamer processing engine. The command-line software was designed to be used with transcript sequences in FASTA file format as well as sequencing data in FASTQ format generated by the Illumina sequencing platform. Due to the relatively large file size and much longer processing time for sequencing data, the web server version is limited to support only transcript sequences as input. For the same reason, we recommend that researchers use tDRnamer on a computing server with multi-core processors instead of personal computers when processing sequencing data.

Reference database generation

tDRnamer requires the use of a reference database that is specifically built with mature and precursor tRNAs of a desired genome. A tool called create_tDRnamer_db is

included in the tDRnamer package with which researchers can create the database via tRNAscan-SE 2.0[1] annotations and sequences downloaded from the Genomic tRNA Database[2], and the genome sequence of the organism. As a default feature, the tool will exclude pseudogenes and tRNA genes with unknown isotype in the database generation process. Moreover, only the high confidence tRNA set, and genes with consistent isotype model and anticodon, prediction score at least 50 bits, and isotype model score at least 80 bits are included for large eukaryotes to avoid confusion and likely inaccurate associations with tRNA-derived repetitive elements which very rarely produce stable mature tRNAs. tRNA genes removed by this filtering step are outputted in the tRNAscan-SE[1] output file format (Supplementary Table 3). Researchers can choose to skip this filtering step by using the --skipfilter option or specify alternative score cutoffs with --score and --isoscore options to fit their needs.

The database building tool generates a Bowtie2[3] index and a BLAST[4] database (Supplementary Table 3) that consist of unique mature tRNA and precursor tRNA sequences based on the provided data for default and maximum sensitivity (--max) search modes respectively. The mature tRNA sequences are created with 3′ CCA tails, removal of introns, and addition of the histidine tRNA-specific post-transcriptional 5′ G base at position -1. Unlike eukaryotic tRNAs that always have the 3′ CCA added enzymatically, some bacterial and archaeal tRNAs have genetically encoded 3′ CCA[5] tails. Therefore, the tool will assess the existence of 3′ CCA in each tRNA gene when the sequence source (--source option) is set for bacteria or archaea, and only make the addition as appropriate. As an additional step, 20 “N” bases flanking each side of the tRNA sequences are added to allow for extra bases appearing off the end of sequenced transcripts, such as potential “CCACCA” ends. Precursor tRNA sequences are created with 50 nt flanking regions to accommodate tDRs that are derived from 5′ leaders or 3′ trailers. When the intergenic region that flanks a tRNA gene is shorter than 50 nt, such as those transcribed in polycistronic operons, only the annotated intergenic region will be included. The reference database also contains the alignments of both mature and precursor tRNA sequences in Stockholm[6] file format for determining the derived location of tDRs (Supplementary Table 3). These alignments that include both primary sequence and secondary structure information are generated using the cmalign program from Infernal v1.1[6] with options “--nonbanded --notrunc -g” and clade-specific tRNA covariance models corresponding to the specified sequence source. For alignments of precursor tRNAs, covariance models trained with eukaryotic, bacterial, or archaeal tRNA genes from tRNAscan-SE 2.0[1] are used. Because tRNAscan-SE covariance models were trained with tRNA genes (some which include introns), the training data were converted to mature tRNA sequences as described above for generating new covariance models using cmbuild from Infernal[6], with options “--hand --wnone --enone”.

Reference databases for the following model organisms in Eukaryota, Bacteria, and Archaea have been pre-built for the web server version and are available for

download: *Arabidopsis thaliana* (TAIR10), *Caenorhabditis elegans* (WBcel235/ce11), *Drosophila melanogaster* (Release 6 plus ISO1 MT/dm6), *Homo sapiens* (GRCh37/hg19 and GRCh38/hg38), *Mus musculus* (GRCm38/mm10 and GRCm39/mm39), *Rattus norvegicus* (Rnor_6.0/rn6 and mRatBN7.2/rn7), *Saccharomyces cerevisiae* S288c, *Schizosaccharomyces pombe* 972h-, *Bacillus subtilis* 168, *Escherichia coli* K-12 MG1655, *Helicobacter pylori* J99, *Mycobacterium tuberculosis* H37Rv, *Haloferax volcanii* DS2, *Methanosarcina barkeri* str Fusaro, *Methanococcus maripaludis* S2, and *Thermococcus kodakarensis* KOD1. Reference databases for additional species may also be requested for use on the web server resource.

Processing small RNA sequencing reads

When searching by sequence (option --mode as “seq”), tDRnamer automatically determines the input data type by file extension. Small RNA sequencing reads should be contained in a FASTQ file called *.fq or *.fastq, or in Gzip compressed format. The sequencing reads are expected to have been pre-processed, with sequencing adapters and low-quality reads removed, before executing tDRnamer. If paired-end reads are used, the two read ends must be merged into pseudo-single-end reads. Although tDRnamer does not include the pre-processing step, trimadapters.py in the companion tRNA-sequencing data analysis pipeline, tRAX (http://trna.ucsc.edu/tRAX/), can be used for this task. To identify possible tDRs, unique sequencing reads are extracted from the input data as the first step of tDRnamer. Only those reads that have at least ten copies are selected by default to eliminate background noise and potential sequencing errors, however a different threshold can be specified with --minread option. A sequence file in FASTA format is generated with the identical read count as the suffix of each sequence name (Supplementary Table 4) and can be further processed using the same routine as using a FASTA file for tDRnamer’s input data.

Identifying tDRs by alignments to reference database

With the default search mode, sequences provided in a FASTA file or as unique sequencing reads extracted from a FASTQ file are aligned to the pre-built reference database using Bowtie2[3] in the very-sensitive mode which ignores quality scores and allows a maximum of 100 alignments per sequence (--very-sensitive --ignore-quals --np 5 -k 100). Because RNA modifications and editing may exist in tDRs, tDRnamer does not use a hard mismatch threshold to filter alignments. Instead, only all best mappings (highest alignment score – AS tag in SAM file) of the sequence alignments are included. In most cases, query sequences with 90% or greater identity relative to the reference are identified. However, if multiple variations are in close proximity, or located near the sequence ends, the alignment quality may drop too far and result in detection failure. Pairwise global alignment between query sequences and reference tRNAs using the

Needleman-Wunsch algorithm[7] can eliminate this shortcoming, yet the long computation time for computing every pairwise alignment makes this method impractical. Therefore, we designed a higher sensitivity search mode (--max option) than the default Bowtie2 search mode for researchers who are missing expected matches to query sequences. We estimated that the run time of this search mode may take six times or longer than the default mode. The mismatch with the source tRNAs can be specified as a threshold for identified tDRs; by default, the –maxmismatch option is set to 10%, which finds most expected matches, yet this parameter can be set to as high as 20%. The initial step of this enhanced sensitivity mode uses NCBI BLAST+[4] to identify source tRNA candidates from a pre-built BLAST database with relaxed sequence similarity search parameters (blastn -word_size 4 -perc_identity 70 -task blastn-short -evalue 5). Due to the minimum word size requirements of BLASTN, tDRs that are shorter than 16 nucleotides and have more than one variation relative to the reference sequences are usually excluded from the search results. For query sequences that are detected by BLAST, but do not have perfect matches to reference sequences, the top target matches are re-aligned globally using EMBOSS[8] (needleall -gapopen 15 -gapextend 2.0) to avoid accidental truncation of hits. The global alignments with variations fewer than the specified threshold (--maxmismatch option) and those perfect match alignments identified with BLAST are formatted in SAM file format to proceed with the tDR naming and annotation process.

When the sequence source (--source option) is set to eukaryotes, sequences aligned to mature tRNAs and pre-tRNAs are both assessed. Otherwise, only those aligned to mature tRNAs are assessed. If a sequence aligns equally well to a mature tRNA and pre-tRNA, the mature tRNA alignment is preferred and the pre-tRNA alignment is excluded. Using this method, sequences that contain a region flanking the tRNA loci in the genome will be considered to be part of the pre-tRNAs, while sequences that contain no flanking sequence will be considered as derived from mature tRNAs. Sequences that are successfully aligned to tRNAs are further processed as identified tDRs.

Naming and annotating tDRs

To calculate the Sprinzl[9] positions and secondary structure regions of tDRs, they are structurally aligned to the reference database of tRNAs. Nucleotide differences including substitutions, insertions, and deletions between tDRs and the matching source tRNAs are determined from the alignments. Using the source tRNA identity information, and the starting and ending Sprinzl[9] positions where the tDR overlaps the source tRNA, tDRnamer then generates the standardized name for each individual tDR (Fig.1a). Existing nucleotide differences can be optionally included as the last element of the tDR name by specifying the --var option. If the tDR is derived from multiple source tRNAs, synonyms following the same naming convention are created based on all the identified

source tRNAs. In addition to outputting a tDR sequence file in FASTA format, a tab-delimited file containing the tDR names, annotations, and sequences is generated, as well as a file containing the alignments of tDRs with source tRNAs in Stockholm[6] format (Supplementary Table 4). Furthermore, the tDRnamer web server version uses NAVIEW[10] to superimpose tDR sequence onto the secondary structure image of its source tRNA for visualization and file download in PNG and postscript formats (Fig. 1b-d, Supplementary Table 4). tDR secondary structure based on minimum free energy is computed using RNAFold[11] with images rendered by Forna[12] and VARNA[13] (Supplementary Fig. 3, Supplementary Table 4).

Grouping tDRs by source tRNAs

If tDRs identified in the same tDRnamer analysis run are derived from the same source tRNAs, they are grouped together to visualize all tDR products processed from the same parent molecule(s). tDRs that are derived from mature tRNAs are grouped separately from those derived from pre-tRNAs. Alignment of tDRs with their corresponding source tRNAs is based on both primary sequence and secondary structure. Numeric IDs are assigned to the source tRNAs belonging to each group in sequential order of the tRNA transcript name or GtRNAdb gene symbol. tDRs in the alignments are marked with the numeric IDs corresponding to their source tRNA(s). A text file is generated as output which contains the tDR groups, each with the member tDR list, the source tRNAs, the tRNA isotype, nucleotide substitutions (mismatches) if they exist, and the tDR alignments in Stockholm[6] format (Supplementary Table 4).

Finding tDR sequences by name

tDR names provided as queries to find their corresponding sequences are split into the defined components according to the naming convention (Fig. 1a) and checked for validity. The source tRNA name and Sprinzl[9] positions extracted from each valid tDR name are used to search for the corresponding sequence in the reference database. If nucleotide differences between tDR and its source tRNA are encoded in the name, the tDR sequence retrieved from the database will be modified to include the changes. A sequence file in FASTA format is generated as output (Supplementary Table 4). tDR sequences found in the search process are then processed using the same method as searching by sequences described above to obtain the same full annotation and grouping output files (Supplementary Fig. 2, Supplementary Table 3).

Re-annotation of tDRs in tRFdb

tDR data including sequences, mapped positions, and source tRNAs were downloaded from tRFdb[14] in CSV file format. Sequences were converted into a FASTA file for each genome to be analyzed. tDRnamer reference databases were generated with default options using tRNA annotations from GtRNAdb[2] release 19 for human assembly

GRCh38/hg38, mouse assembly GRCm39/mm39, *Caenorhabditis elegans* assembly WBcel235/ce11, *Drosophila melanogaster* assembly BDGP Rel. 6/dm6, *Schizosaccharomyces pombe* 972h-, *Xenopus tropicalis* assembly Xenopus_tropicalis_v9.1/xenTro9, zebrafish assembly GRCz11/danRer11, and *Rhodobacter sphaeroides* ATCC 17025. tDR sequences in each genome were then analyzed and annotated using tDRnamer with --minlen option set as 13 to allow inclusion of shorter transcripts.

References

1. Chan, P. P., Lin, B. Y., Mak, A. J. & Lowe, T. M. tRNAscan-SE 2.0: improved detection and functional classification of transfer RNA genes. *Nucleic Acids Res.* **49**, 9077–9096 (2021).
2. Chan, P. P. & Lowe, T. M. GtRNAdb 2.0: an expanded database of transfer RNA genes identified in complete and draft genomes. *Nucleic Acids Res* **44**, D184-9 (2016).
3. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat Methods* **9**, 357–9 (2012).
4. Camacho, C. *et al.* BLAST+: architecture and applications. *BMC Bioinformatics* **10**, 421 (2009).
5. Redko, Y., Li de la Sierra-Gallay, I. & Condon, C. When all's zed and done: the structure and function of RNase Z in prokaryotes. *Nat. Rev. Microbiol.* **5**, 278–286 (2007).
6. Nawrocki, E. P. & Eddy, S. R. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* **29**, 2933–5 (2013).
7. Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443–453 (1970).
8. Rice, P., Longden, I. & Bleasby, A. EMBOSS: The European Molecular Biology Open Software Suite. *Trends Genet.* **16**, 276–277 (2000).
9. Steinberg, S., Misch, A. & Sprinzl, M. Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res* **21**, 3011–5 (1993).
10. Bruccoleri, R. E. & Heinrich, G. An improved algorithm for nucleic acid secondary structure display. *Comp Appl Biosci* **4**, 167–173 (1998).
11. Lorenz, R. *et al.* ViennaRNA Package 2.0. *Algorithms Mol Biol* **6**, 26 (2011).
12. Kerpedjiev, P., Hammer, S. & Hofacker, I. L. Forna (force-directed RNA): Simple and effective online RNA secondary structure diagrams. *Bioinformatics* **31**, 3377–3379 (2015).
13. Darty, K., Denise, A. & Ponty, Y. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics* **25**, 1974–1975 (2009).
14. Kumar, P., Mudunuri, S. B., Anaya, J. & Dutta, A. tRFdb: a database for transfer RNA fragments. *Nucleic Acids Res.* **43**, D141–D145 (2015).