Mobile DNA

RESEARCH

Open Access

Reproducible evaluation of transposable element detectors with McClintock 2 guides accurate inference of Ty insertion patterns in yeast

Jingxuan Chen[1], Preston J. Basting[1], Shunhua Han[1], David J. Garfinkel[2] and Casey M. Bergman[1,3*]

Abstract

Background Many computational methods have been developed to detect non-reference transposable element (TE) insertions using short-read whole genome sequencing data. The diversity and complexity of such methods often present challenges to new users seeking to reproducibly install, execute, or evaluate multiple TE insertion detectors.

Results We previously developed the McClintock meta-pipeline to facilitate the installation, execution, and evaluation of six first-generation short-read TE detectors. Here, we report a completely re-implemented version of McClintock written in Python using Snakemake and Conda that improves its installation, error handling, speed, stability, and extensibility. McClintock 2 now includes 12 short-read TE detectors, auxiliary pre-processing and analysis modules, interactive HTML reports, and a simulation framework to reproducibly evaluate the accuracy of component TE detectors. When applied to the model microbial eukaryote *Saccharomyces cerevisiae*, we find substantial variation in the ability of McClintock 2 components to identify the precise locations of non-reference TE insertions, with RelocaTE2 showing the highest recall and precision in simulated data. We find that RelocaTE2, TEMP, TEMP2 and TEBreak provide consistent estimates of ~50 non-reference TE insertions per strain and that Ty2 has the highest number of non-reference TE insertions in a species-wide panel of ~1000 yeast genomes. Finally, we show that best-in-class predictors for yeast applied to resequencing data have sufficient resolution to reveal a dyad pattern of integration in nucleosome-bound regions upstream of yeast tRNA genes for Ty1, Ty2, and Ty4, allowing us to extend knowledge about fine-scale target preferences revealed previously for experimentally-induced Ty1 insertions to spontaneous insertions for other *copia*-superfamily retrotransposons in yeast.

Conclusion McClintock (https://github.com/bergmanlab/mcclintock/) provides a user-friendly pipeline for the identification of TEs in short-read WGS data using multiple TE detectors, which should benefit researchers studying TE insertion variation in a wide range of different organisms. Application of the improved McClintock system to simulated and empirical yeast genome data reveals best-in-class methods and novel biological insights for one of the most widely-studied model eukaryotes and provides a paradigm for evaluating and selecting non-reference TE detectors in other species.

Keywords Bioinformatics, High-throughput sequencing, *Saccharomyces cerevisiae*, Transposable elements, Variant calling

*Correspondence:
Casey M. Bergman
cbergman@uga.edu
Full list of author information is available at the end of the article

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

Background

Transposable elements (TEs) are mobile, repetitive DNA sequences that occupy large fractions of most eukaryotic genomes and can cause mutations of large effect [1]. Since their discovery, a variety of cytological, molecular and genomic techniques have been developed to identify TE insertions in genomes at varying levels of resolution [2–8]. In principle, the most comprehensive method for detecting new TE insertions in a genome is whole genome sequencing (WGS) followed by *de novo* genome assembly and systematic genome-wide annotation of TEs. However, the difficulty of accurately assembling repetitive DNA sequences using short reads [9, 10] and the challenges of generating high-quality long-read genome sequencing data hinder using assembly-based methods for genome-wide detection of TE insertions, especially for large population samples [11].

Currently, the most widely-used approach for detection of new TE insertions involves mapping unassembled short-read WGS data to a reference genome and inferring the location and family using split-read or read-pair information [8]. Over the last decade, many bioinformatic methods have been developed for the reference-based detection of TE insertions using short-read WGS data (https://tehub.org/en/resources/repeat_tools) [12]. Most of these methods were initially developed for a specific organism with its unique TE composition, and thus have different design goals and features. The general performance of short-read TE detectors is typically unknown because of the "self-assessment trap" and limited number of methods compared in primary studies [13]. A few independent performance evaluation studies have been conducted comparing subsets of short-read TE detectors, but these studies differ in their evaluation frameworks, selection of tools, and focal organisms [14–16]. Crucially, no study to date has provided a reproducible evaluation of multiple TE detectors, making it difficult to validate published results or extend findings beyond the context of the initial evaluation study. Despite these issues, one emerging theme from current evaluation studies is that considerable variation in performance exists among different short-read TE detectors, across organisms and evaluation frameworks [8, 14–16]. As such, there is not yet any clear basis to select the best short-read TE detector for a new organism, and thus researchers must base tool choice by extrapolation from performance on other taxa or employ multiple methods to ensure robust biological conclusions [17–19].

Previously, we developed a meta-pipeline called McClintock [15] to facilitate the installation and execution of six first-generation short-read TE detectors (ngs_te_mapper [20], PoPoolationTE [21], RelocaTE [22], RetroSeq [23], TE-locate [24], and TEMP [25]). The original McClintock system automated installation of these six "component" TE detection methods, provided a common interface to run all components, reduced the number of shared input files, and generated a standard set of output files [15]. Since its initial development, the McClintock system has been used to support detection of TE insertions and enable biological discoveries in a variety of organisms and biological contexts [15–19, 26–44] and to facilitate comparative evaluation of multiple TE detectors [15, 16, 45].

A long-term aim of the McClintock project was to develop a flexible framework to incorporate additional TE detectors as they were published, allowing researchers to run the most relevant set of methods and compare performance on their genomes of interest. While the original pipeline made it easier to install and run multiple TE detectors and compare their output, its initial design had several limitations that made it difficult to achieve this long-term aim. Most importantly, the original McClintock pipeline was implemented in Bash with minimal use of functions or classes to encapsulate, abstract, or modularize the code, and most variables were defined in the global scope. These shortcomings made it difficult to modify and expand the original codebase without compromising existing functionality, and thus hindered the addition of new TE detectors to the original framework.

The original McClintock system also had several limitations related to its installation. Component TE detectors were installed automatically by McClintock, however the software dependencies for each component had to be installed manually, which required substantial effort on the part of the user. Moreover, all components in the original McClintock system ran in a single computing environment, which necessitated a mutually-compatible set of software dependencies to be installed. This fragile configuration also caused compromises in the versions of software dependencies that were used and, in some cases, locked the original system to increasingly out-of-date versions of software dependencies.

Even after successful installation, the original McClintock system had a number of limitations regarding its usability. Notably, error handling was largely absent from the original codebase. Thus, when failures occurred during the execution of component TE detectors or their software dependencies, McClintock would continue to run, often producing a cascade of error messages from downstream processes that made it difficult for users to know how and why the pipeline failed. Similarly, the original pipeline had strict input file formatting requirements (e.g., requiring unzipped fastq files), but which would provide no warnings or messages if the input files did not comply to required specifications. Moreover, the original McClintock system hard-coded some input and post-processing parameters that were not easily modified by users but impacted the performance of component methods. The original McClintock pipeline also output TE insertion predictions only in Browser Extensible Data (BED) format (https://samtools.github.io/hts-specs/BEDv1.pdf), rather than the Variant Call Format (VCF) [46] that is standard for reporting genetic variation relative to a reference genome. Finally, the original pipeline also did not produce plots or tables to facilitate comparison of predictions across component methods.

To address these limitations, we completely re-implemented McClintock in Python leveraging Conda (https://github.com/conda/conda) and Snakemake [47] to improve the installation, error handling, speed, stability, and extensibility of the pipeline. We also incorporated six new TE detectors (ngs_te_mapper2 [35], PoPoolationTE2 [48], RelocaTE2 [49], TEBreak [50], TEFLoN [51], and TEMP2 [52]), doubling the total number of components available in the new McClintock system. In addition, the updated McClintock system now includes new modules to automate read trimming, provide estimates of TE family abundance from depth of coverage, and produce an integrated HTML report summarizing predictions made by all components. Importantly, McClintock now also provides a new reproducible simulation framework that allows users to evaluate the performance of component TE detectors in a flexible manner across different organismal contexts. In this report, we describe the structure and function of the updated McClintock system, highlight its improvements and new features, evaluate component TE detector performance using simulated and empirical yeast genome data, and use the updated McClintock system to provide new biological insights into the pattern of TE insertion in *S. cerevisiae*.

Implementation

Re-implementation of McClintock in Python using Conda and Snakemake

Here we describe the major features distinguishing the re-implemented McClintock 2 system from the original version reported in Nelson et al. [15] (Table 1). We initially sought to further develop the original Bash-based McClintock system reported in Nelson et al. [15] by improving its installation using the Conda package and environment management system (https://github.com/conda/conda). The use of Conda allows the creation of distinct environments (both for McClintock itself and for the component methods wrapped in McClintock) that can be reproducibly generated on different computing systems without the need for root privileges. This modification to the original McClintock system permitted external software dependencies to be automatically installed separately for each component, and each component to be executed in its own isolated environment. While solving some of the problems with the installation and versioning of external dependencies in the original system, this development work led to an endpoint because it did not solve most of the software engineering and usability limitations described above. Therefore, the final version of the Bash-based McClintock system extended to use Conda (referred to in this report as McClintock 1) was retired.

We next completely rewrote the McClintock system in Python 3 using Conda and Snakemake [47], with the initial aim of recapitulating the general functionality in McClintock 1. The new McClintock 2 codebase is heavily modularized and leverages the native error handling functions in Python. Snakemake was used to simplify and fully automate installation of all components and their software dependencies (either from Conda channels like Bioconda [53] or directly from project repositories), and to automatically create independent Conda

Table 1 Major features that distinguish the original McClintock version reported in Nelson et al. [15] and the McClintock 2 system reported in the current study

	Nelson et al. [15]	McClintock 2
McClintock implementation	Bash	Python
Workflow management	N.A.	Snakemake
Dependency installation	Manual	Conda (automated)
# Component methods	6	12 (with extensibility)
Read trimming and QC	N.A.	trimgalore module
TE copy number estimation	N.A.	coverage module
Output format	.BED	.VCF and .BED
Summary report	.CSV	.CSV and .HTML (interactive)
Reproducible simulation system	N.A.	Built-in (with documentation)

N.A. indicates that the feature is not available environments for each component. Additionally, because the "meta-pipeline" architecture of McClintock is naturally suited to a rule-based workflow, we encapsulated of each component as an individual rule in a Snakefile that is executed by Snakemake inside a specific Conda environment whose dependencies do not conflict with any other component/rule.

Using the Snakemake workflow manager to underpin the McClintock 2 meta-pipeline provided several advantages. Since the Snakemake engine determines the rules that are run to create the requested outputs, no additional coding logic is required in McClintock itself to manage the complexities of which components were run and in which order, or how files shared by multiple components are created and used (e.g., reference genomes, trimmed read files, BAM files, etc.). Importantly, this virtue of Snakemake applies regardless of how many components are executed at run time or incorporated into McClintock 2 in the future, permitting flexible execution of various components by users and easy addition of new components by developers. Finally, the Snakemake engine can run multiple independent rules simultaneously if enough central processing unit (CPU) cores are available, potentially allowing for multiple rules to be run in parallel on multi-core systems. In the re-implemented system, if $n>1$ cores are requested, McClintock 2 by default now allocates a maximum of $n/2$ (for an even number of cores) or $(n-1)/2$ (for an odd number of cores) to each rule which allows multiple rules to be executed in parallel. Optionally, users can specify the "–serial" flag which prevents multiple rules from being run in parallel and allows rules that use multiple cores to have access to maximal number of cores requested.

After integrating the original six components (ngs_te_mapper [20], PoPoolationTE [21], RelocaTE [22], RetroSeq [23], TE-locate [24], and TEMP [25]) into the McClintock 2 system, we cross-validated its accuracy versus the original McClintock 1 system using simulated genomes generated by our new reproducible simulation platform (see below and Supplemental Text). We then incorporated six more recently-developed TE detectors that fulfilled our original inclusion criteria [15] into the re-engineered McClintock system: ngs_te_mapper2 [35], PoPoolationTE2 [48], RelocaTE2 [49], TEBreak [50], TEFLoN [51], and TEMP2 [52].

In addition to providing a full re-implementation and additional TE detectors, McClintock 2 has new functionality not present in the original system (Table 1). First, McClintock 2 now has a preprocessing module to perform read trimming and quality control (QC) using TrimGalore (https://github.com/FelixKrueger/TrimGalore) and produce a read QC report using MultiQC [54]. If the read trimming option is set (either optionally in conjunction with user-specified components, or in a default run that executes all 12 components), then trimmed reads are used as input for other components in McClintock 2. Second, McClintock 2 has a new "coverage module" that can generate estimates of TE copy number by computing relative depth-of-coverage for each query TE sequence normalized by depth-of-coverage in non-repetitive regions of the genome. The coverage module also produces plots for each query that allow users to inspect variation in read depth across TE sequences, which may be caused by the existence of multiple TE sub-families that differ by structural variants (e.g., [17, 55]). Third, McClintock 2 reports non-reference TE insertion variants predicted by each component in VCF format [46] in addition to BED files containing standardized information about reference and non-reference TE insertions. Finally, McClintock 2 is able to generate an interactive HTML report for each run that allows users to view, sort, and filter results for each component and TE family (Fig. 1).

A reproducible simulation system for evaluating McClintock component performance

In addition to improvements in the meta-pipeline itself, McClintock 2 now also provides code to automatically simulate TE insertions in any user-supplied genome and generate synthetic WGS datasets that can be used as input to evaluate component TE detector performance. Previously, we developed a "single synthetic insertion" framework for evaluating the performance of McClintock components to detect non-reference TE insertions and used this framework to evaluate the performance of the six TE detectors in McClintock 1 [15]. This evaluation framework created a synthetic genome comprised of a single non-reference TE and its corresponding target site duplication (TSD) inserted into in an otherwise-unmodified reference genome, then simulated a corresponding paired-end WGS dataset that could be used as input for McClintock. The Bash code for this simulation framework was not released as part of the original McClintock 1 system, did not run inside a controlled environment, was hard-coded to simulate TE insertion preferences for only one species (*S. cerevisiae*), only evaluated only one WGS fold-coverage (100×), and did not generate standard performance metrics like precision and recall.

To overcome these limitations, we re-implemented a flexible and fully-reproducible version of this single synthetic insertion simulation framework in Python 3 using Conda and Snakemake. The new simulation framework in McClintock 2 allows customization for different organisms, and consists of three major parts that run under controlled Conda environments:

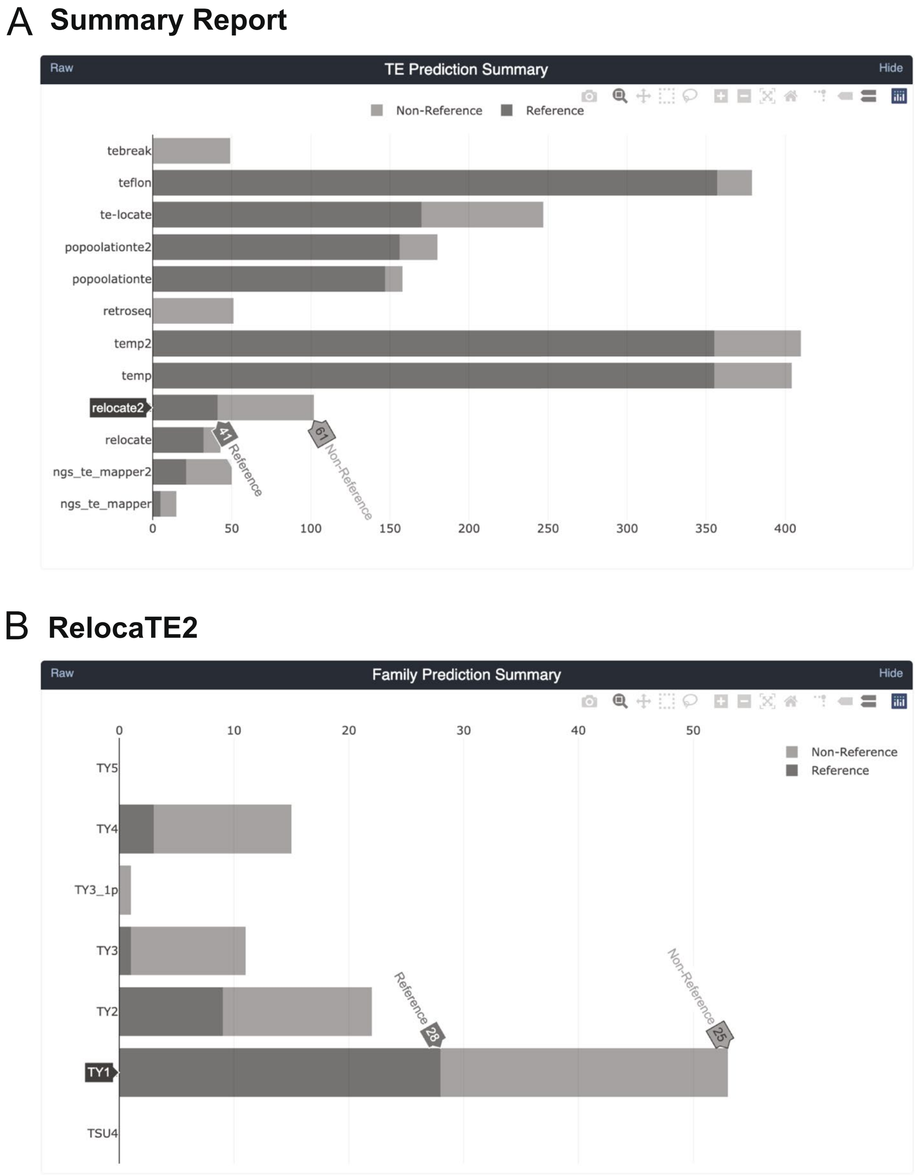

Fig. 1 Sample screenshots from the new interactive HTML report in McClintock 2. The HTML report generates summary information for the McClintock run including interactive bar plots for: (**A**) the number of reference, non-reference, and total number of predictions made across all TE families by all 12 component methods; and (**B**) the number of reference, non-reference, and total number of predictions made for a specific component method (e.g., RelocaTE2). Barplots from the report shown were generated by a complete McClintock run (commit SHA d2b819a) applied to Illumina 101-bp paired-end sequences for *S. cerevisiae* strain YJM1460 (SRA: SRR800842) down-sampled to 50× fold-coverage

(i) A Python 3 script that creates a configuration file with information about the simulation experiment to be run in JavaScript Object Notation (JSON) format. Required inputs to the configuration script include user-supplied files containing the reference genome, TE library, and permissible locations for TE insertions. The configuration script also allows users to modify the number of replicates in the experiment, properties of the synthetic genomes (e.g., TSD length and strand of the TE insertion), properties of the simulated WGS datasets (e.g., read length, single or paired end reads, insertion sizes for paired-end reads, read error rates, fold-coverage), the read simulator used to generate WGS datasets (ART [56] or wgsim [57]), and computing resources for each cluster job (e.g., number of threads, amount of memory).

(ii) A primary Snakemake workflow that reads input parameters from the JSON file created by the configuration script, then submits and manages jobs on a high-performance computing (HPC) cluster. Currently our simulation framework is designed for use with cluster systems running the SLURM workload manager, but it can be adapted to other HPC systems using alternative Snakemake cluster profiles (https://github.com/Snakemake-Profiles/). Each submitted job runs a Python script that creates a new synthetic genome with one additional insertion, simulates a corresponding WGS dataset from this synthetic genome, then runs the McClintock pipeline on the simulated WGS dataset. Replicates for each combination of parameters are submitted as individual cluster jobs to maximize available cluster resources, and any potential failed jobs are automatically re-submitted by the workflow to minimize effort needed to monitor the simulation experiment.

(iii) A secondary Snakemake workflow that is automatically executed after all cluster jobs submitted by the primary workflow are completed, which summarizes results and generates preliminary plots from all replicates of the overall simulation experiment. This workflow runs several scripts to calculate performance metrics (e.g., mean numbers of non-reference TE predictions, precision, recall) across replicates, generate precision/recall curves as a function of fold-coverage, and generate positional accuracy and predicted TSD length plots for each fold-coverage level. Performance metrics are automatically generated by the simulation system and calculated as follows. Non-reference TE predictions with the identical family, chromosome, and strand, with start and end locations falling within an N bp window from the synthetic insertion (N=0, 5, 100, 300, or 500) are labeled as "within-N" predictions. Within-0 bp predictions have the identical start and end coordinates as the synthetic insertion in that simulated sample and are considered "exact" predictions. The mean number of reference and non-reference TE predictions (overall and at various window sizes) per synthetic genome are then calculated across all replicates. To calculate precision and recall, non-reference TE predictions within-N bp from the insertion site that match the TE family and strand of the synthetic insertion are regarded as true positives (TP), while all other predictions are regarded as false positives (FP). If more than one non-reference prediction for the correct TE family and strand was made within-N bp from the synthetic insertion, only one was considered a TP and all others are considered FPs. If a replicate had zero non-reference TE predictions within-N bp from the synthetic insertion, it was regarded as a false-negative (FN). For each N bp window size, recall was calculated as TP/(TP+FN) and precision was calculated as TP/(TP+FP).

Materials and methods

Evaluation of McClintock component performance on simulated data

We conducted a series of four simulation experiments to validate aspects of the McClintock re-implementation, demonstrate the utility of our new reproducible simulation framework, and evaluate the ability of McClintock components to predict non-reference TE insertions in *S. cerevisiae*. The different simulation experiments varied in the version of McClintock used, which component methods were run, their placement of synthetic insertions, the read simulator used to generate WGS datasets, and performance metrics used (See Additional file 1 for details). Simulations 1 and 2 were designed to cross-validate the McClintock 2 meta-pipeline and simulation system at a fixed coverage (100×), while Simulations 3 and 4 were designed to evaluate component method performance at range of fold-coverages (3×, 6×, 12×, 25×, 50×, 100×). In all cases, synthetic genomes were created with the new reproducible simulation framework available in McClintock 2 (see Implementation above), the UCSC sacCer2 version of the *S. cerevisiae* S288c reference genome (to allow cross-validation with results in Nelson et al. [15]), and 5-bp TSDs for all Ty families [15, 58–61]. All simulations and McClintock jobs run on simulated data used the following input files (provided in https://github.com/bergmanlab/mcclintock/blob/master/test/): the UCSC sacCer2 version of the *S. cerevisiae* S288c reference genome (sacCer2.fasta, commit SHA 53a5e8f); reference TE annotations (reference_TE_locations.gff, commit SHA 571ef0e), taxonomy files (sac_cer_te_families.tsv, commit SHA 571ef0e), and canonical sequences for *S. cerevisiae* Ty elements (sac_cer_TE_seqs.fasta, commit SHA d649713) from [62]. Quantitative results from all simulations can be found in Additional file 2 (average numbers of TEs) and Additional file 3 (recall and precision). UpSet plots [63] visualizing overlaps among component methods for true positive predictions at different window sizes and fold-coverages for Simulations 3 and 4 can be found in Additional file 4.

Evaluation of McClintock CPU efficiency and run time on empirical data

To test whether Snakemake's ability to run multiple rules simultaneously could improve run-time performance in the new McClintock system, we applied the original McClintock 1 (commit SHA 714fe6d) and the new McClintock 2 (commit SHA f766e73) systems to a 101-bp paired-end Illumina HiSeq 2000 dataset for *S. cerevisiae* strain YJM1460 [64] that was down-sampled to 50× and 100× coverage using seqtk (v1.3) [65]. To compare run times for McClintock 1 and McClintock 2, we invoked both versions of the system only with the original six components present in McClintock 1 ("-m ngs_te_mapper,relocate,temp,retroseq,popoolationte,te-locate"). To evaluate the impact on run time caused by addition of the six new components in McClintock 2, we also invoked McClintock 2 using all 12 components ("-m ngs_te_mapper,ngs_te_mapper2,relocate, relocate2,temp,temp2,retroseq,popoolationte, popoolationte2,te-locate,teflon,tebreak"). In both cases, we varied the number of cores requested (1, 5, 10, 15, 20, 25, 30) but kept the amount of RAM constant (20 Gb). For all runs, we used unzipped, untrimmed fastq reads (as required by McClintock 1) and the following input files (provided in https://github.com/bergmanlab/mcclintock/blob/master/test/): the UCSC sacCer2 version of the *S. cerevisiae* S288c reference genome (sacCer2.fasta, commit SHA 53a5e8f); reference TE annotations (reference_TE_locations.gff, commit SHA 571ef0e), taxonomy files (sac_cer_te_families.tsv, commit SHA 571ef0e), and canonical sequences for *S. cerevisiae* Ty elements (sac_cer_TE_seqs.fasta, commit SHA d649713) from [62]. Trials were run serially on the same cluster node (an AMD EPYC processor with a total of 32 cores and 256GB RAM, operating system Linux 3.10.0-1160.36.2.el7.x86_64). The average and standard deviation of CPU efficiency and run times was calculated from five replicates of each setting (defined as the combination of number of cores and McClintock run options). All possible settings were run as a batch for each replicate to control for differences in cluster node performance over the duration of the experiment.

Analysis of TE insertions in a diverse panel of wild and domesticated yeast strains

To demonstrate the utility of the McClintock 2 system for large-scale mining of empirical short-read WGS data, we applied our pipeline to Illumina paired-end datasets for 1,011 *S. cerevisiae* isolates [64, 66] sequenced to variable depths (from 50× to 900× fold-coverage). Illumina WGS data were downloaded from NCBI Sequence Read Archive and converted to fastq format using SRA toolkit (v2.10.8) [67]. To minimize the influence of variable coverage and reduce computing resources, if the coverage of the original dataset was greater than 50×, we down-sampled it to 50× fold-coverage using seqtk (v1.3) [65]. The full McClintock 2 pipeline using all 12 components plus the coverage and trimgalore module ("-m ngs_te_mapper,ngs_te_mapper2,relocate, relocate2,temp,temp2,retroseq,popoolationte,popoolationte2,te-locate,teflon,tebreak,coverage,trimgalore") (commit SHA 7aa5298) was applied to this yeast resequencing dataset using the following input files (provided in https://github.com/bergmanlab/mcclintock/blob/master/test/): the UCSC sacCer2 version of the *S. cerevisiae* S288c reference genome (sacCer2.fasta, commit SHA 53a5e8f); reference TE annotations (reference_TE_locations.gff, commit SHA 571ef0e), taxonomy files (sac_cer_te_families.tsv, commit SHA 571ef0e), and canonical sequences for *S. cerevisiae* Ty elements (sac_cer_TE_seqs.fasta, commit SHA d649713) from [62]. Multi-sample VCF files for non-reference TEs predicted by McClintock 2 component methods the 1,011 *S. cerevisiae* isolates can be found in Additional file 5.

We identified non-reference TE insertions in the vicinity of tRNA genes using BEDtools (v2.30.0, "window -u -sw -l 1000 -r 500") [68]. BEDtools ("closest -D b") was used to calculate the genomic distance between the start position of each non-reference TE insertion and its closest tRNA transcription start site (TSS). To analyze the relationship between the distribution of non-reference TE insertions and nucleosome occupancy upstream of tRNA genes, we mapped micrococcal nuclease digestion with deep sequencing (MNase-seq) data from [69] to the sacCer2 version of the *S. cerevisiae* S288c reference genome using Bowtie (v1.2.3 [70]) and generated genome-wide nucleosome occupancy profiles using NUCwave [71]. Whole genome nucleosome occupancy profiles in .wig format were converted to .bw format using wigToBigWig (v377) [72], and then bwtool (v20170428, "aggregate 2000:500") [73] was used to calculate nucleosome occupancy in the region 2 kb upstream to 500 bp downstream of all tRNA TSSs.

Data analysis and visualization

Data were analyzed and visualized in R (v3.6.3) using the ggplot2 (v3.3.3) [74] and GGally (v2.1.1, https://ggobi.github.io/ggally/) packages. UpSet plots were generated by comparing overlaps among true-positive predictions across component methods using UpSetR (v1.4.0) [75]. Kernel density plots showing distributions of Ty insertion sites around tRNA genes were created with the ggplot2 function "geom_density" with Gaussian kernel and 0.4× default bandwidth. In contrast to Nelson et al. [15], predictions from TEMP [25] in the current study were not partitioned into those with split-read versus read-pair evidence. Scripts to generate results in this study can be found in Additional file 6.

Results and discussion

Validation of McClintock 2 implementation and simulation system

Here we present a Python-based re-implemention and extension of the McClintock TE detection meta-pipeline, as well as a reproducible simulation system to test the performance of component TE detectors integrated into McClintock. The major features that distinguish the new McClintock 2 system from the original version reported in Nelson et al. [15] are described in the Implementation section above and are summarized in Table 1. To validate the McClintock 2 implementation and simulation system, we conducted a series of *in silico* experiments testing the ability of the six original components to detect a single synthetic non-reference TE insertion introduced into an otherwise unmodified *S. cerevisiae* reference genome (see Additional file 1 for details). Contrasts between different simulations allowed us to test whether various improvements in the new McClintock 2 system could yield results that replicate those previously published in Nelson et al. [15] including: (i) the Python-based implementation of the simulation system, (ii) the Python-based implementation of the meta-pipeline, and (iii) the random insertion model in the single insertion simulation framework. Results from these simulations (Fig. S1) allowed us to validate that the McClintock 2 re-implementation yields results that broadly replicate those previously published in Nelson et al. [15].

Parallelization by Snakemake in McClintock 2 improves multi-core job run times

McClintock 2 uses Snakemake as a workflow manager, which allows parallelization of multiple rules when enough CPU cores are available. To investigate potential improvements in computational efficiency due to Snakemake job management in McClintock 2, we executed McClintock 1 and McClintock 2 (using either the six original or all 12 components) with a variable number of cores under controlled computing environments. For this analysis, we used a 101-bp paired-end Illumina dataset from the *S. cerevisiae* domesticated palm-wine strain Y12 (YJM1460; SRA: SRR800842) downsampled to either 50× or 100× (see Materials and Methods for details). In general, we observe a decreasing trend in CPU efficiency for all McClintock run configurations as the number of CPU cores increases (Fig. 2A). However, when more than one core is used, we found that McClintock 2 has better CPU efficiency than McClintock 1 for the six original components, and that CPU efficiency is highest for all numbers of cores used when executing all 12 McClintock 2 components. Improved CPU efficiency for McClintock 2 is observed at both 50× and 100× coverage.

When comparing run times for the six original components at both 50× and 100× coverage (Fig. 2B), we found that McClintock 2 was slightly slower (~10%) than McClintock 1 when using only a single core, however McClintock 2 finished faster than McClintock 1 when five or more cores were available. Run times for McClintock 2 decreased up to 15 CPU cores when executing either the original six or all 12 components, while no major benefits in run time were observed beyond 5 CPU cores for McClintock 1. Interestingly, when 15 or more cores are allocated, executing McClintock 2 with all 12 components is faster than running McClintock 1 using only six components. Together, these results indicate that parallelization afforded by using Snakemake in McClintock 2 results in better utilization of computing resources in multi-core computing systems relative to McClintock 1.

Evaluation of McClintock 2 component TE detectors on simulated yeast data

McClintock 2 now includes six additional component methods for detecting non-reference TE insertions using short-read WGS data that are not available in McClintock 1: ngs_te_mapper2 [35], PoPoolationTE2 [48], RelocaTE2 [49], TEBreak [50], TEFLoN [51], and TEMP2 [52]). Four of the new components (ngs_te_mapper2, PoPoolationTE2, RelocaTE2, TEMP2) are "second-generation" versions of tools previously incorporated in McClintock 1, and the other two (TEBreak and TEFLoN) represent new TE detection strategies not represented in McClintock 1.

To evaluate the relative performance of these 12 TE detection strategies, we used the new reproducible single insertion simulation approach available in McClintock 2. The evaluation of McClintock 2 component methods performed here differs from the approach originally used in Nelson et al. [15] in several ways (see Additional file 1 for details). First, we used a more biologically realistic model to randomly select the family and location of simulated non-reference TE insertions over a range of genomic positions. Second, we investigated the effects of component performance over a range of fold-coverages, which provides practical guidance for users to optimize data generation. Third, the summary statistics used in Nelson et al. [15] are not standard evaluation metrics, and therefore difficult to compare with other evaluation studies (e.g., [14, 16]). Thus, here we used the standard evaluation metrics of recall and precision derived from the number of true-positive, false-positive, and false-negative predictions at various window sizes (see Implementation for details). Finally, here we evaluated performance using simulations that model yeast TE target preferences (Simulation 3) as in Nelson et al. [15], as well as random

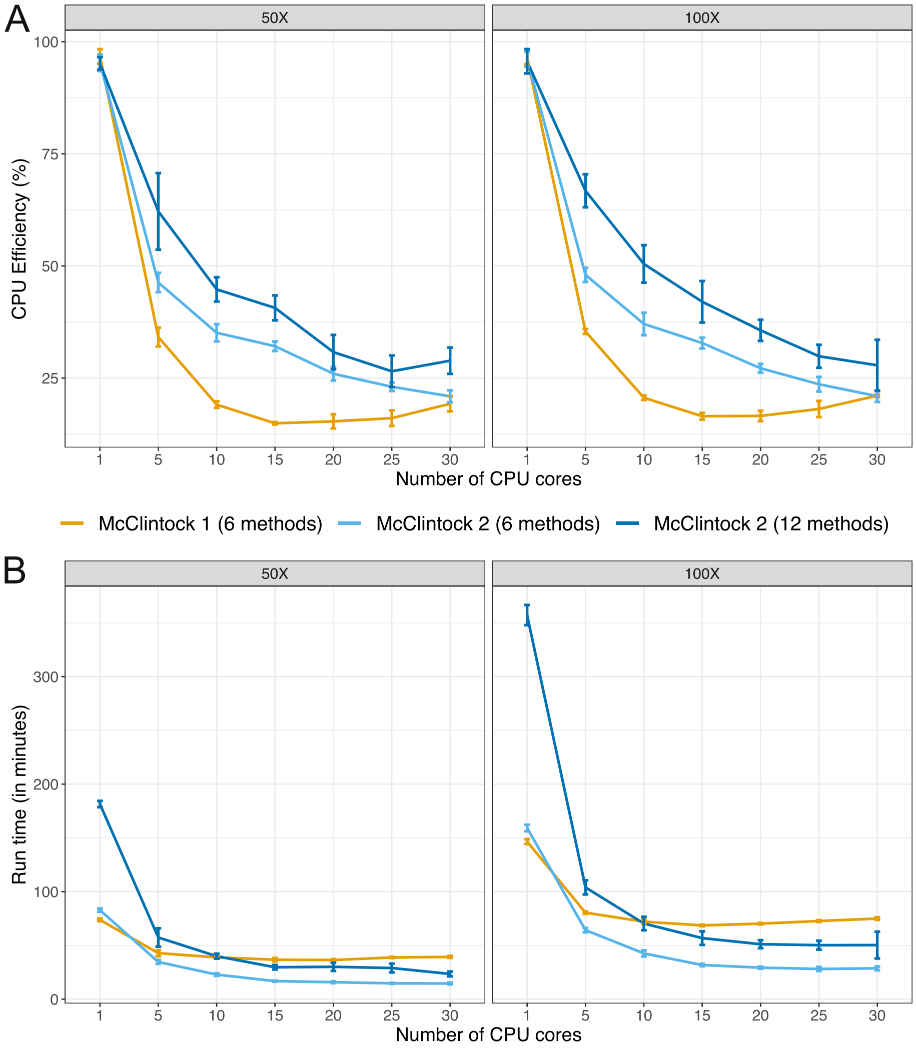

Fig. 2 McClintock 2 re-implementation improves CPU efficiency and run time on multi-core architectures. Shown are average (**A**) CPU efficiency and (**B**) run times across 5 replicates of McClintock 1 (six component methods, orange line) or McClintock 2 (same six component methods as for McClintock 1, light blue line; all 12 component methods in McClintock 2, dark blue line) applied to 50× and 100× Illumina 101-bp paired-end sample for *S. cerevisiae* strain YJM1460 (SRA: SRR800842). Error bars indicate standard deviations across replicates. To allow compatibility with McClintock 1, all runs were performed on unzipped, untrimmed fastq files and thus run times do not include these processes

insertion into non-repetitive DNA (Simulation 4). Simulation 4 helps us interpret how insertion into repetitive DNA in the reference genome influences component performance in yeast and gives insight into potential component performance in other organisms that don't have strong TE targeting preferences.

As shown previously for the original six components in McClintock 1 [15], different methods in McClintock 2 vary in their ability to accurately predict the position of synthetic TE insertions in simulated data (Figs. S2, S3, S4, S5). Several components (i.e., ngs_te_mapper, ngs_te_mapper2, RelocaTE, RelocaTE2, TEFLoN and TEBreak) make most predictions very close the expected location of the synthetic insertion, whereas other components (i.e., PoPoolationTE, PoPoolationTE2, and TE-locate) predict insertions over a wider distance around the synthetic insertion (~500bp). Even for components that attempt to identify insertion breakpoints at nucleotide resolution using split-read information, only ngs_te_mapper and RelocaTE2 have predictions with the expected 5-bp TSD length for all Ty families (Figs. S6, S7). This method-dependent variation in positional accuracy makes it difficult to fairly compare performance of different components at a single window size. Therefore, we calculated performance metrics over a range of window sizes for overlaps between predicted and simulated insertions (exact, within-5, within-100, within-300, and within-500 bp). Inspection of the complete set of results for all window sizes (Figs. S8, S9, S10, S11) revealed that two window sizes (exact and within-100 bp) could illustrate the majority of key features in McClintock 2 component performance across the insertion models and range of fold-coverages investigated here.

To guide selection of TE detectors in the model species *S. cerevisiae*, we first analyzed the performance of McClintock 2 components using data from Simulation 3 where TEs were inserted in their biologically realistic locations for yeast in regions upstream of tRNA genes [76–80] (Fig. 3; purple lines). Three general trends can be observed in the recall and precision curves for McClintock 2 components for non-reference TEs in tRNA promoter regions. First, component performance is higher when allowing a more relaxed window size (within-100 bp) to classify TP predictions (dashed lines) than when requiring TP predictions to be exact (solid lines), consistent with most methods making many predictions that are not precise to exact nucleotide coordinates. Only ngs_te_mapper makes all its predictions exactly, and thus performance curves are overlayed for both window sizes shown for this method. Second, component performance typically increases with sequencing depth and then plateaus at a method-specific coverage. Some exceptions to this trend are observed for PoPoolationTE recall and RelocaTE and TEFLoN precision, which show decreasing performance with increasing coverage. Performance curves for most components suggest that 25× is the most cost-effective depth of sequencing coverage, and 50× is sufficient to optimize performance for most component methods. Third, second generation versions of TE detectors – apart from PoPoolationTE2 – typically show improved performance relative to their first generation counterparts, suggesting that gains in TE detector performance can be made by continued method development. Overall, results from Simulation 3 suggest that RelocaTE2 has remarkably high recall (~92%) and precision (~98%) to predict non-reference TE insertions in tRNA promoter regions of the yeast genome (at 50× coverage and ≥5 bp window size). The recall and precision of TEMP, TEMP2, RetroSeq and TEBreak to detect TE insertions in yeast promoter regions are all both greater than 75% when allowing non-exact predictions in WGS datasets with greater than 50× coverage.

While biologically relevant for *S. cerevisiae*, Simulation 3 potentially underestimates component performance since synthetic insertions in the yeast insertion model are often placed into fragments of TE sequences that occur upstream of tRNA genes in the reference genome [61, 62]. To gain insight into other organismal contexts and understand how insertion into reference TE sequences may impact McClintock 2 component performance in yeast, we also investigated a model of random insertion into unique regions of the *S. cerevisiae* genome (Simulation 4). Recall and precision curves for the random insertion model (Fig. 3; orange lines) show the same general trends but have consistently higher performance relative to results for the same component in the tRNA promoter insertion model (Fig. 3; purple lines). General performance improvements in Simulation 4 relative to Simulation 3 indicate that prediction of non-reference TE insertions into locally TE-rich regions (like tRNA promoters in yeast) is more challenging for most component methods than for those in non-repetitive DNA. Consistent with results from the tRNA promoter model, RelocaTE2 has the highest recall and precision of any component in McClintock 2, with essentially perfect performance in non-repetitive regions of the *S. cerevisiae* genome (~100%) even at exact base-pair resolution. At the within-100 bp window size, TEMP, TEMP2, RetroSeq and TEBreak also perform very well at identifying non-reference insertions in unique regions. We note that two additional methods – ngs_te_mapper2 and TEFLoN – show dramatically better recall for non-exact predictions in non-repetitive regions relative to tRNA promoter regions, putting these methods in a similar performance class as TEMP, TEMP2, RetroSeq and TEBreak for insertions in non-repetitive DNA.

In summary, results from our reproducible simulation experiments show that RelocaTE2 is the best-performing component method currently in the McClintock 2 meta-pipeline to detect non-reference TE insertions in *S. cerevisiae* under both the biologically relevant tRNA promoter and non-repetitive insertion models. If exact base-pair positional accuracy is not required, TEMP, TEMP2, RetroSeq and TEBreak also exhibit high performance to detect TE insertions in *S. cerevisiae* tRNA promoter regions at 50× coverage. Furthermore, our results suggest that at 50× coverage TEMP, TEMP2, RetroSeq, TEBreak, ngs_te_mapper2, and TEFLoN may have high performance to detect TE insertions in non-repetitive regions of other genomes. Finally, our results indicate that users should provide McClintock 2 with WGS datasets of at least 25× fold-coverage to generate the best performance from most component methods, but that coverage higher than 50× may not lead to further performance benefits.

Best-in-class component methods reveal consistent insights into patterns of TE insertion in yeast.

Our reproducible simulation system allowed us to validate the McClintock 2 re-implementation and evaluate component methods under ideal conditions, where only one TE insertion per strain is to be detected and all sequencing data is generated in an identical way.

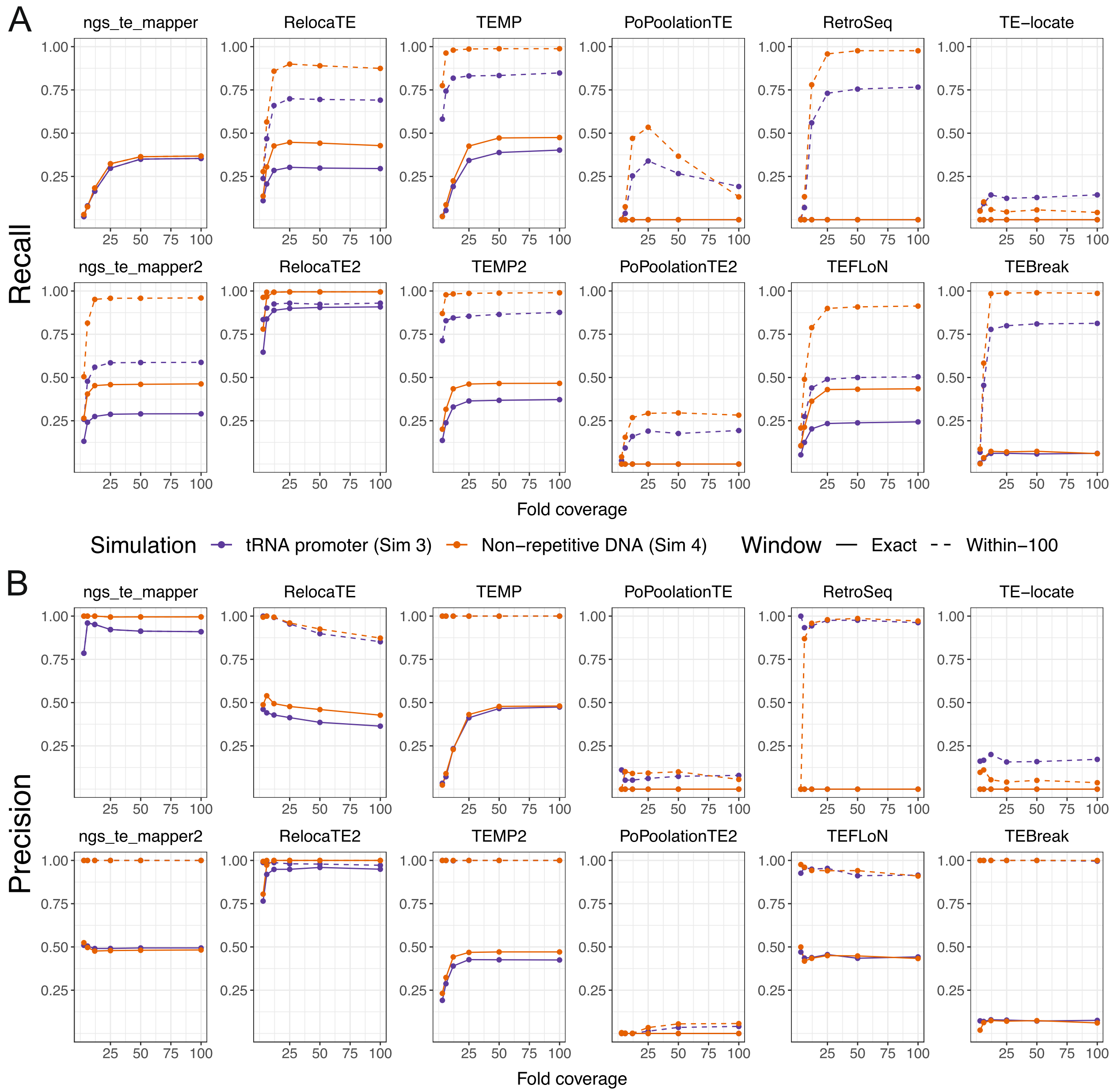

Fig. 3 Performance of McClintock 2 component methods in simulated yeast WGS data. Shown are the (**A**) recall and (**B**) precision across different fold-coverage for individual compnent methods to detect single synthetic insertions in an otherwise unmodified *S. cerevisiae* reference genome. Purple lines (Simulation 3) model the biologically realistic insertion preferences of yeast TEs, with synthetic Ty insertions created upstream of tRNA genes in regions that often have fragments of prior TE insertions in the reference genome. Orange lines (Simulation 4) model random insertions in non-repetitive regions, which allows insight into the effects of insertion within repetitive DNA and component's performance for organisms without strong TE targeting preferences. Points indicate tested fold-coverage configurations, i.e, 3×, 6×, 12×, 25×, 50× and 100×. Solid lines represent performance estimates for non-reference TE predictions made at the exact site of the synthetic insertion. Dashed lines represent performance estimates for non-reference TE predictions made within 100 bp surrounding the synthetic insertion site. The six original component methods in McClintock 1 are on the top row of each panel, and the six new methods in McClintock 2 are on the second row of each panel

However, analysis of large-scale empirical re-sequencing data is expected to present additional challenges to TE detectors, such as more complex insertion patterns in a given strain (e.g., multiple insertions, tandem elements) and variability in WGS dataset composition across strains (e.g., fold-coverage or median insert size). To assess component method performance under a more realistic population genomic scenario, we applied the complete McClintock 2 system (all 12 components plus the coverage module) to empirical WGS datasets from a world-wide sample of 1,011 *S. cerevisiae* strains isolated from diverse environments and geographical locations [64, 66]. Since these WGS datasets varied significantly in their fold-coverage (50× to 900×), we down-sampled all WGS datasets to 50× prior to running McClintock 2 based on simulation results above that indicated 50× coverage is sufficient to maximize component method performance (Fig. 3). Our goals for this analysis were to evaluate whether McClintock 2 component methods: (i) predict consistent numbers of non-reference TE insertions per strain (overall and by Ty family), and (ii) have sufficient positional accuracy to identify expected patterns of targeting in yeast tRNA promoter regions. This analysis also allowed us to generate the first species-wide Ty insertion variant call sets for *S. cerevisiae* (Additional file 5).

To evaluate consistency among component methods on empirical WGS data, we first quantified the distribution of non-reference TE predictions made by each method across 1,011 yeast strains (Fig. 4A). We observe substantial variation in median values of non-reference Ty insertions across all 12 methods, as well as in the correlation among methods in the number of predicted non-reference Ty insertions (Fig. S12). Interestingly, median values of ~50 non-reference Ty insertions per strain are consistently predicted by the five component methods (RelocaTE2, TEMP, TEMP2, RetroSeq and TEBreak) (Fig. 4A; bold outlines) that showed the best performance (when considering non-exact predictions) in simulated data under the biologically relevant tRNA promoter model (Fig. 3; purple dotted lines). These five components also have the highest pairwise correlation of non-reference Ty counts per strain across methods (Fig. S12). We note that the original six components do not show the general pattern of ~50 non-reference Ty insertions per strain (Fig 4A; top row) and have some of the lowest correlation of non-reference Ty counts across methods (Fig. S12), underscoring the value of incorporating additional recently-developed methods into McClintock 2 to reveal consistent biological patterns that could not be revealed by the McClintock 1 system. Since McClintock 2 component methods do not distinguish full-length elements from solo LTRs, the estimate of ~50 non-reference Ty insertions per yeast strain is likely to represent a combination of both structural types. Indeed, applying the McClintock 2 coverage module to internal coding regions of all Ty families, we estimate that the number of full-length elements per strain is ~20 (Fig. S13), suggesting that many non-reference Ty insertions detected by McClintock 2 components in this population sample are polymorphic solo LTRs [62, 81].

Next, we assessed the consistency among component methods to classify the TE family of non-reference predictions across all 1,011 isolates (Fig. 4B). As for estimates of overall non-reference Ty abundance per strain, we observe variable patterns in relative Ty family abundance across component methods, with most methods being able to differentiate Ty families that are known to be active (Ty1, Ty2, Ty3 and Ty4) from those that are inactive or low-abundance (Ty3_1p, Tsu4, Ty5). As observed for numbers of non-reference TEs per strain above, we note that no clear pattern of Ty family abundance can be discerned across the original six components (Fig. 4B; top row). However, a consistent pattern of relative Ty family abundance is observed for four of the five component methods (RelocaTE2, TEMP, TEMP2 and TEBreak) that show both consistent non-reference TE abundance per strain and high performance in simulated data. These four methods all indicate that Ty2 has the highest overall number of non-reference Ty insertions in this sample of *S. cerevisiae* strains. While Ty1 is often cited as the most abundant Ty family in *S. cerevisiae* because of its high copy number in the reference strain S288c [61, 62], the finding that Ty2 has the highest number of non-reference insertions in this diverse worldwide sample of *S. cerevisiae* strains is supported by orthogonal copy-number estimates from the McClintock 2 coverage module (Fig. S14) and a similar depth-based approach used in recent independent study [82]. In contrast, the fifth component method that shows consistent Ty abundance per strain and high performance in simulated data – RetroSeq – predicts more Ty1 and Ty2 insertions in empirical yeast data, which we infer to be misidentification because of the similarity in LTR sequences for these two Ty families [61, 83].

Previously [15], we used the well-established pattern that Ty1, Ty2, Ty3 and Ty4 non-randomly target promoters of genes transcribe by RNA polymerase III such as tRNAs (reviewed in [84]) to validate non-reference TE predictions made by McClintock 1 component methods in small sample of 93 *S. cerevisiae* genomes [15, 64]. Here, we confirm that the majority of non-reference predictions made by all component methods in McClintock 2 for these four Ty families are in the expected vicinity of tRNA genes in the species-wide *S. cerevisiae* resequencing dataset analyzed here (Fig. 4B) [64, 66]. Moreover, the four methods (RelocaTE2, TEMP, TEMP2 and TEBreak) that consistently predict similar numbers and families of non-reference Ty insertions in empirical yeast genome data, also predict similar proportions of insertions in tRNA genes. Combined with their high performance in simulated data, we conclude that RelocaTE2, TEMP, TEMP2 and TEBreak are the "best-in-class" methods for

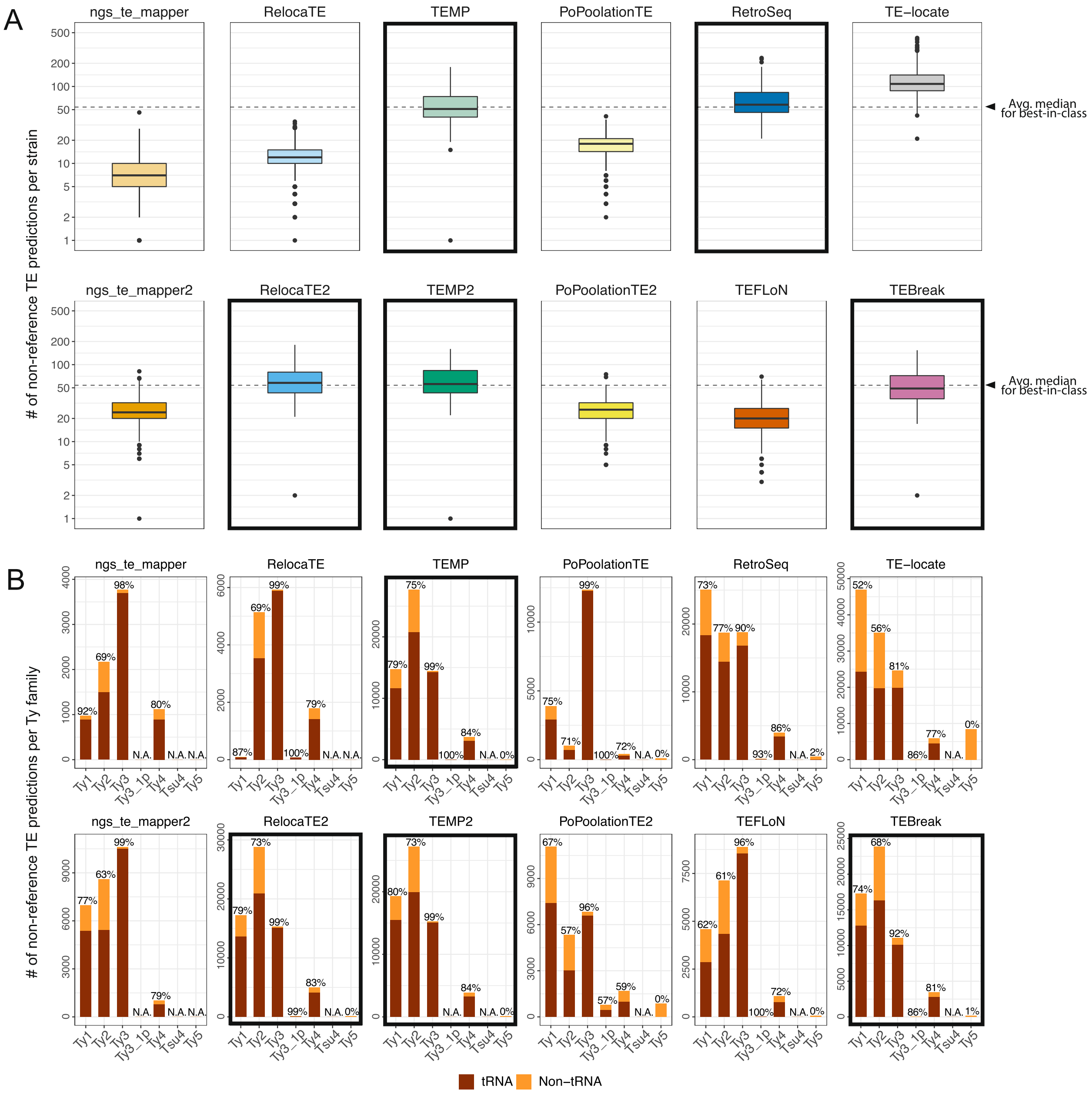

Fig. 4 Numbers of Ty elements predicted by McClintock 2 components in a world-wide sample of yeast strains. **A** Numbers of non-reference TE predictions per strain (summed over all Ty families) and (**B**) numbers of non-reference TE predictions across Ty families (summed over all strains) in 1,011 *S. cerevisiae* WGS samples [64, 66], down-sampled to 50× fold-coverage. In panel (**A**), lines inside boxes indicate median values, colored boxes show interquartile ranges (IQR), whiskers show values 1.5 ×IQR of the upper or lower quartiles, and the dots indicate outliers that beyond 1.5 × IQR. Components with bold outlines in panel (**A**) have have median values of ~50 non-reference Ty insertions per strain, as well as recall and precision both >75% in tRNA promoter insertion simulations when allowing non-exact predictions in WGS datasets with >50× coverage (see Fig. 3). We note that the y-axis is on a $\log_{10}$ scale, and that 16 zero-count data points and one extreme TE-locate data point (count=749) is removed to aid with visualization. In panel (**B**) total numbers of non-reference TE predictions are partitioned as "tRNA" (dark red) if they are located between 1000 bp upstream and 500 bp downstream of tRNA genes, or "non-tRNA" (orange) if outside these windows. Note that the y-scale varies for each component method. The percentage of near tRNA gene predictions is annotated at the top of each bar. "N.A." means no such Ty family was found using that component. Components with bold outlines in panel (**B**) predict consistent relative TE family abundance and also have properties of components with bold outlines in panel (**A**), and thus we designate them as "best-in-class" methods for predicting non-reference TE insertions in *S. cerevisiae*. Dashed lines in panel (**A**) represent the average of the median number of non-reference TE insertions across the four best-in-class methods (n=54)

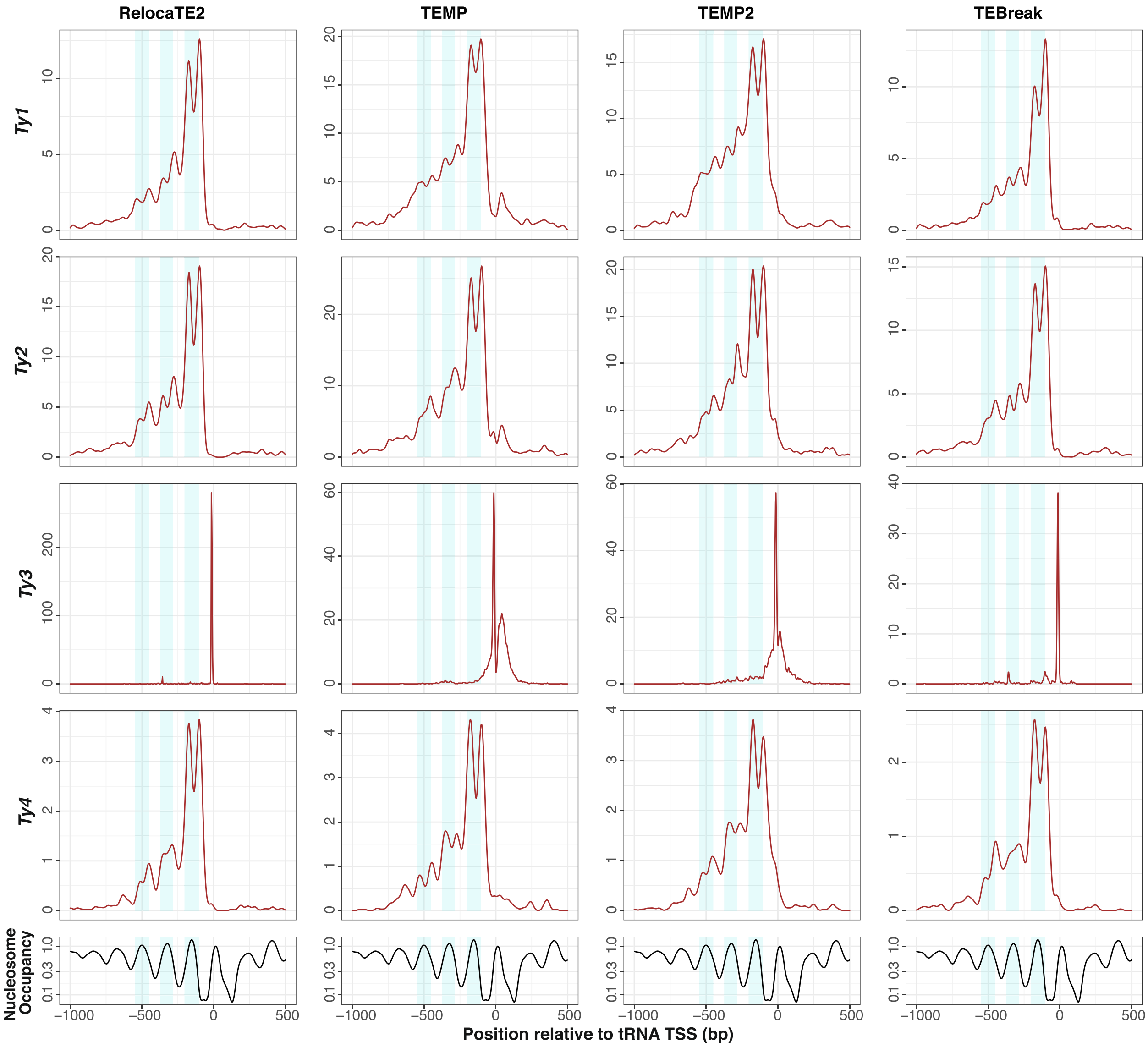

Fig. 5 *copia*-superfamily retrotransposons show a dyad pattern of insertion in nucleosome-bound regions upstream of yeast tRNA genes. The top four rows show density profiles of non-redundant insertion sites for non-reference Ty predictions made by best-in-class McClintock 2 components (RelocaTE2, TEMP, TEMP2 and TEBreak) in tRNA promoter regions in a panel of 1,011 *S. cerevisiae* WGS samples [64, 66], down-sampled to 50× fold-coverage. Only the four Ty familes (Ty1, Ty2, Ty3 and Ty4) that are know to non-randomly target tRNA genes are included in this analysis. The bottom row shows nucleosome occupancy inferred using MNase-seq data from [69]. Light blue shaded areas indicate 100-bp regions surrounding peaks of nucleosome occupancy

predicting Ty insertions in *S. cerevisiae* WGS resequencing data.

Finally, we sought to test whether predictions in species-wide *S. cerevisiae* WGS data made by the best-in-class components in McClintock 2 have sufficient positional accuracy to recapitulate the association between Ty1 insertion profiles and nucleosome occupancy in tRNA promoters observed for experimentally-induced insertions [77, 79, 85]. Recent work by Hays et al. [40] using WGS data from experimentally-evolved genomes has shown that Ty insertions predicted by RelocaTE2, aggregated across all Ty families, exhibit periodicity upstream of tRNA genes in *S. cerevisiae* [40]. However, their analysis does not partition insertions to individual Ty families, show a direct correlation with nucleosome profiles, or demonstrate that this signal is not an artifact of a single TE detection system. As shown in Fig. 5, spontaneous Ty1 insertions predicted in species-wide *S. cerevisiae* resequencing data by the four “best-in-class” component methods show a clear profile of dyad peaks on the first nucleosome-bound regions upstream of tRNA TSS with weaker dyad signals on the second and third upstream nucleosome-bound regions, as observed for experimentally-induced Ty1 insertions [77, 79, 85]. Intriguingly, spontaneous Ty2 and Ty4 insertions predicted by all four best-in-class methods (and by RetroSeq, ngs_te_mapper2 and TEFLoN; Figs. S15 and S16) also have dyad peaks on the first nucleosome-bound region upstream of *S. cerevisiae* tRNA genes. These results are consistent with the observation that the domain of Ty1 integrase shown to be responsible for targeting nucleosomes upstream of tRNA genes is conserved in Ty2 and Ty4 [86, 87]. In contrast, spontaneous Ty3 insertions show no correlation with nucleosomal profiles and for RelocaTE2 and TEBreak are restricted to ~15 bp upstream of tRNA TSSs as seen for experimentally-induced insertions [80]. Profiles of spontaneous Ty3 insertions predicted by TEMP and TEMP2 are more variable than the other best-in-class methods, which can be explained by lower positional accuracy for a proportion of Ty3 predictions made by these methods (see Fig S2 and S3). A proportion of lower resolution predictions for TEMP may also explain the additional peak of insertions for Ty1 and Ty2 downstream of the tRNA TSS, which is not observed for spontaneous insertions predicted by RelocaTE2, TEMP2, or TEBreak or in experimentally-induced Ty1 insertions [77, 79, 85]. Overall, these results demonstrate that the insertion profiles for Ty1 and Ty3 upstream of tRNA genes are not artifacts of experimental induction, and suggest that the molecular mechanisms responsible for the targeting of Ty1 to nucleosomes are conserved in the related *copia*-superfamily retrotransposons Ty2 and Ty4.

Conclusions

In this report we present a re-implemented version of the McClintock TE detection system with improved installation, error handling, speed, stability, and extensibility. Relative to the original system [15], McClintock 2 includes six new short-read TE detectors, pre-processing and analysis modules, interactive HTML reports, and a reproducible simulation system that we used here to cross-validate the re-implemented system and evaluate component TE detectors. We acknowledge that not all available short-read TE detectors are currently incorporated into the McClintock 2 meta-pipeline, however use of Conda and Snakemake in the re-implemented meta-pipeline puts us in a better position to integrate additional or improved components in the future. Other potential improvements include creating a Bioconda package for McClintock 2 itself, which could aid installation and enhance portability by the automatic construction of Docker/Singularity containers containing McClintock 2, its component methods, and all relevant dependencies [88]. We are also considering modifying McClintock 2 to allow installation of TE detectors from user-downloaded executables or source code to be able to integrate high-performing methods like MELT [89] which have restrictive licenses that prevent automatic installation.

A major improvement in McClintock 2 is the availability of a reproducible simulation framework that allows evaluation of component method performance in a wide range of organismal contexts besides *S. cerevisiae*, including humans and other model organisms. In the future, we plan to modify the simulation framework to allow multiple insertions in a single synthetic genome to reduce the number of replicates needed to analyze larger genomes that have longer run times. We also plan to implement a complementary evaluation system that uses TE annotations in long-read assemblies as gold standards to empirically benchmark TE detectors using short-read WGS data from the same strain [16, 52].

Based on performance evaluations in simulated WGS data, we conclude that RelocaTE2 is the best McClintock 2 component to detect non-reference TE insertions in *S. cerevisiae* at base-pair accuracy, which supports the use of this method in recent yeast experimental evolution studies [40, 90]. However, we caution against the general conclusion that RelocaTE2 is the best McClintock 2 component for other species, since this method can have long run times that precludes its use in larger genomes [16, 52]. By combining results from simulated data with consistency analyses in empirical WGS data, we also identify TEMP, TEMP2 and TEBreak as additional "best-in-class" methods for predicting Ty insertions in *S. cerevisiae* WGS resequencing data with 50× fold-coverage or higher, albeit with lower positional accuracy than RelocaTE2. Results from the four best-in-class McClintock 2 components for *S. cerevisiae* provide additional support to the emerging view that Ty2 is the most abundant TE in this species [82], although we acknowledge that non-independence of strains and non-random sampling in the strain panel analyzed [64, 66] may bias this conclusion. Finally, we show that best-in-class McClintock 2 components for *S. cerevisiae* generate reasonable non-reference TE predictions with sufficient resolution to reveal a dyad pattern of integration in nucleosome-bound regions upstream of yeast tRNA genes for Ty1, Ty2, and Ty4. This finding allows knowledge about fine-scale target preferences revealed in experimentally-induced Ty1 insertions [77, 79, 85] to be extended to natural Ty1 insertions and related *copia*-superfamily retrotransposons in yeast. Together with the new bioinformatics resources provided in McClintock 2, our work provides novel biological insights about TEs in one of the best-studied model systems and a paradigm for selecting optimal non-reference TE detectors in a diversity of organisms.

Abbreviations

BED	Browser extensible data
CPU	Central processing unit
FN	False negative
FP	False positive
HPC	High-performance computing
IQR	Inter-quartile range
JSON	JavaScript object notation
QC	Quality control
TE	Transposable element
TP	True positive
TSD	Target site duplication
TSS	Transcription start site
VCF	Variant call format
WGS	Whole genome sequencing

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13100-023-00296-4.

Additional file 1. Supplemental Text, Tables and Figures. Combined PDF of all Supplemental Text, Figures, and Tables.

Additional file 2. Average numbers of TEs predicted by McClintock component methods in simulated data. CSV file with average numbers of reference and non-reference TEs predicted by McClintock 1 or 2 component methods in replicate simulations of yeast genomes with a single additional synthetic insertion, overall and at varying window size distances relative to the synthetic insertion.

Additional file 3. Recall and precision for McClintock 2 component methods in simulated data. CSV file with numbers of true positive, false positive, and false negative non-reference TEs predicted by McClintock component methods (and corresponding recall and precision values) in replicate simulations of yeast genomes with a single additional synthetic insertion, at varying window size distances relative to the synthetic insertion.

Additional file 4. Overlaps between numbers of non-reference TEs predicted by McClintock component methods in simulated data. UpSet plots visualizing overlaps among component methods for true positive predictions at different window sizes and fold-coverages for Simulations 3 and 4.

Additional file 5. Non-reference TEs predicted by McClintock component methods in a global sample of *S. cerevisiae* isolates. Multi-sample VCF files for non-reference TEs predicted by McClintock 2 component methods in 1,011 *S. cerevisiae* WGS datasets.

Additional file 6. Scripts to generate and analyze results for simulations, run-time performance, and empirical data analysis. Zip file containing READMEs and scripts to to generate and analyze results for simulations, run-time performance, and empirical data analysis.

Acknowledgements

We thank members of the Bergman and Garfinkel Labs for helpful discussions throughout the project; the BioConda community for contributing software recipes for software dependencies used in this project; and the Georgia Advanced Computing Resource Center for computing resources and HPC support.

Authors' contributions

P.J.B. developed the McClintock 2 meta-pipeline with contributions from J.C., S.H. and C.M.B. J.C. and P.J.B. developed the reproducible simulation framework. J.C. conducted the analysis of simulated and empirical yeast genomes. J.C., P.J.B., and C.M.B analyzed the data. C.M.B. conceived of and managed the project. J.C., P.J.B. and C.M.B. drafted the manuscript with contributions from S.H. and D.J.G. All authors revised and approved the final manuscript.

Funding

This work was funded by a University of Georgia Research Education Award Traineeship (P.J.B.), the University of Georgia Research Foundation (C.M.B.), and NIH grant R01GM124216 (D.J.G and C.M.B.).

Availability of data and materials

The McClintock TE detection meta-pipeline and reproducible simulation system are available at: https://github.com/bergmanlab/mcclintock/. Output from McClintock runs on simulated and empirical yeast data can be found in Additional files 2-5. Scripts to generate results presented in this report are available in Additional file 6.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Author details

[1]Institute of Bioinformatics, University of Georgia, Athens, GA, USA. [2]Department of Biochemistry and Molecular Biology, University of Georgia, Athens, GA, USA. [3]Department of Genetics, University of Georgia, Athens, GA, USA.

Received: 21 March 2023 Accepted: 9 June 2023
Published online: 14 July 2023

References

1. Bourque G, Burns KH, Gehring M, Gorbunova V, Seluanov A, Hammell M, Imbeault M, Izsvák Z, Levin HL, Macfarlan TS, Mager DL, Feschotte C. Ten things you should know about transposable elements. Genome Biol. 2018;19(1):199. https://doi.org/10.1186/s13059-018-1577-z.
2. Biemont C, Monti-Dedieu L, Lemeunier F. Detection of transposable elements in Drosophila salivary gland polytene chromosomes by in situ hybridization. Methods Mol Biol. 2004;260:21–8. https://doi.org/10.1385/1-59259-755-6:021.
3. Yu W, Lamb JC, Han F, Birchler JA. Cytological visualization of DNA transposons and their transposition pattern in somatic cells of maize. Genetics. 2007;175(1):31–9. https://doi.org/10.1534/genetics.106.064238.
4. Bergman CM, Quesneville H. Discovering and detecting transposable elements in genome sequences. Brief Bioinforma. 2007;8(6):382–92. https://doi.org/10.1093/bib/bbm048.
5. Saha S, Bridges S, Magbanua ZV, Peterson DG. Computational approaches and tools used in identification of dispersed repetitive DNA sequences. Trop Plant Biol. 2008;1(1):85–96. https://doi.org/10.1007/s12042-007-9007-5.
6. Ray DA, Batzer MA. Reading TE leaves: New approaches to the identification of transposable element insertions. Genome Res. 2011;21(6):813–20. https://doi.org/10.1101/gr.110528.110.
7. Xing J, Witherspoon DJ, Jorde LB. Mobile element biology: new possibilities with high-throughput sequencing. Trends Genet. 2013;29(5):280–9. https://doi.org/10.1016/j.tig.2012.12.002.
8. Ewing AD. Transposable element detection from whole genome sequence data. Mob DNA. 2015;6:24. https://doi.org/10.1186/s13100-015-0055-3.
9. Treangen TJ, Salzberg SL. Repetitive DNA and next-generation sequencing: computational challenges and solutions. Nat Rev Genet. 2012;13(1):36–46. https://doi.org/10.1038/nrg3117.

10. Peona V, Blom MPK, Xu L, Burri R, Sullivan S, Bunikis I, Liachko I, Haryoko T, Jonsson KA, Zhou Q, Irestedt M, Suh A. Identifying the causes and consequences of assembly gaps using a multiplatform genome assembly of a bird-of-paradise. Mol Ecol Resour. 2021;21(1):263–86. https://doi.org/10.1111/1755-0998.13252.
11. De Coster W, Weissensteiner MH, Sedlazeck FJ. Towards population-scale long-read sequencing. Nat Rev Genet. 2021;22(9):572–87. https://doi.org/10.1038/s41576-021-00367-3.
12. Elliott TA, Heitkam T, Hubley R, Quesneville H, Suh A, Wheeler TJ. The TE Hub Consortium TE Hub: A community-oriented space for sharing and connecting tools, data, resources, and methods for transposable element annotation. Mob DNA. 2021;12(1):16. https://doi.org/10.1186/s13100-021-00244-0.
13. Norel R, Rice JJ, Stolovitzky G. The self-assessment trap: can we all be better than average? Mol Syst Biol. 2011;7(1):537. https://doi.org/10.1038/msb.2011.70.
14. Rishishwar L, Marino-Ramirez L, Jordan IK. Benchmarking computational tools for polymorphic transposable element detection. Brief Bioinforma. 2017;18(6):908–18. https://doi.org/10.1093/bib/bbw072.
15. Nelson MG, Linheiro RS, Bergman CM. McClintock: an integrated pipeline for detecting transposable element insertions in whole-genome shotgun sequencing data. G3. 2017;7:2749–2762. https://doi.org/10.1534/g3.117.040915.
16. Vendrell-Mir P, Barteri F, Merenciano M, González J, Casacuberta JM, Castanera R. A benchmark of transposon insertion detection tools using real data. Mob DNA. 2019;10(1):53. https://doi.org/10.1186/s13100-019-0197-9.
17. Bergman CM, Han S, Nelson MG, Bondarenko V, Kozeretska I. Genomic analysis of P elements in natural populations of Drosophila melanogaster. PeerJ. 2017;5:3824. https://doi.org/10.7717/peerj.3824.
18. Manee MM, Jackson J, Bergman CM. Conserved noncoding elements influence the transposable element landscape in Drosophila. Genome Biol Evol. 2018;10(6):1533–45. https://doi.org/10.1093/gbe/evy104.
19. Daigle AT, Deiss TC, Melde RH, Bergthorsson U, Katju V. Bergerac strains of Caenorhabditis elegans revisited: expansion of Tc1 elements imposes a significant genomic and fitness cost. G3. 2022;12(11):214. https://doi.org/10.1093/g3journal/jkac214.
20. Linheiro RS, Bergman CM. Whole genome resequencing reveals natural target site preferences of transposable elements in Drosophila melanogaster. PLoS ONE. 2012;7(2):30008. https://doi.org/10.1371/journalpone0030008.
21. Kofler R, Betancourt AJ, Schlotterer C. Sequencing of pooled DNA samples (pool-seq) uncovers complex dynamics of transposable element insertions in Drosophila melanogaster. PLoS Genet. 2012;8(1):1002487. https://doi.org/10.1371/journalpgen1002487.
22. Robb SMC, Lu L, Valencia E, Burnette JM, Okumoto Y, Wessler SR, Stajich JE. The use of RelocaTE and unassembled short reads to produce high-resolution snapshots of transposable element generated diversity in rice. G3. 2013;3(6):949–957. https://doi.org/10.1534/g3.112.005348.
23. Keane TM, Wong K, Adams DJ. RetroSeq: transposable element discovery from next-generation sequencing data. Bioinformatics. 2013;29(3):389–90. https://doi.org/10.1093/bioinformatics/bts697.
24. Platzer A, Nizhynska V, Long Q. TE-Locate: a tool to locate and group transposable element occurrences using paired-end next-generation sequencing data. Biology. 2012;1(2):395–410. https://doi.org/10.3390/biology1020395.
25. Zhuang J, Wang J, Theurkauf W, Weng Z. TEMP: a computational method for analyzing transposable element polymorphism in populations. Nucleic Acids Res. 2014;42(11):6826–38. https://doi.org/10.1093/nar/gku323.
26. Ilyin AA, Ryazansky SS, Doronin SA, Olenkina OM, Mikhaleva EA, Yakushev EY, Abramov YA, Belyakin SN, Ivankin AV, Pindyurin AV, Gvozdev VA, Klenov MS, Shevelyov YY. Piwi interacts with chromatin at nuclear pores and promiscuously binds nuclear transcripts in Drosophila ovarian somatic cells. Nucleic Acids Res. 2017;45(13):7666–80. https://doi.org/10.1093/nar/gkx355.
27. Ryazansky S, Radion E, Mironova A, Akulenko N, Abramov Y, Morgunova V, Kordyukova MY, Olovnikov I, Kalmykova A. Natural variation of piRNA expression affects immunity to transposable elements. PLoS Genet. 2017;13(4):1006731. https://doi.org/10.1371/journalpgen1006731.
28. Duc C, Yoth M, Jensen S, Mouniee N, Bergman CM, Vaury C, Brasset E. Trapping a somatic endogenous retrovirus into a germline piRNA cluster immunizes the germline against further invasion. Genome Biol. 2019;20(1):127. https://doi.org/10.1186/s13059-019-1736-x.
29. Bast J, Jaron KS, Schuseil D, Roze D, Schwander T. Asexual reproduction reduces transposable element load in experimental yeast populations. eLife. 2019;8:48548. https://doi.org/10.7554/eLife.48548.
30. Kordyukova M, Sokolova O, Morgunova V, Ryazansky S, Akulenko N, Glukhov S, Kalmykova A. Nuclear Ccr4-Not mediates the degradation of telomeric and transposon transcripts at chromatin in the Drosophila germline. Nucleic Acids Res. 2020;48(1):141–56. https://doi.org/10.1093/nar/gkz1072.
31. Jimenez-Ruiz J, Ramirez-Tejero JA, Fernandez-Pozo N, Leyva-Perez MdlO, Yan H, Rosa Rdl, Belaj A, Montes E, Rodriguez-Ariza MO, Navarro F, Barroso JB, Beuzon CR, Valpuesta V, Bombarely A, Luque F. Transposon activation is a major driver in the genome evolution of cultivated olive trees (Olea europaea L.). Plant Genome. 2020;13(1):20010. https://doi.org/10.1002/tpg2.20010.
32. Ilyin AA, Stolyarenko AD, Klenov MS, Shevelyov YY. Various modes of HP1a interactions with the euchromatic chromosome arms in Drosophila ovarian somatic cells. Chromosoma. 2020;129(3):201–14. https://doi.org/10.1007/s00412-020-00738-5.
33. Oggenfuss U, Badet T, Wicker T, Hartmann FE, Singh NK, Abraham L, Karisto P, Vonlanthen T, Mundt C, McDonald BA, Croll D. A population-level invasion by transposable elements triggers genome expansion in a fungal pathogen. eLife. 2021;10:69249. https://doi.org/10.7554/eLife.69249.
34. Taylor MB, Skophammer R, Warwick AR, Geck RC, Boyer JM, yEvo Students, Walson M, Large CRL, Hickey AS-M, Rowley PA, Dunham MJ. yEvo: experimental evolution in high school classrooms selects for novel mutations that impact clotrimazole resistance in Saccharomyces cerevisiae. G3. 2022;12(11):246. https://doi.org/10.1093/g3journal/jkac246.
35. Han S, Basting PJ, Dias GB, Luhur A, Zelhof AC, Bergman CM. Transposable element profiles reveal cell line identity and loss of heterozygosity in Drosophila cell culture. Genetics. 2021;219(2):113. https://doi.org/10.1093/genetics/iyab113.
36. Yan H, Haak DC, Li S, Huang L, Bombarely A. Exploring transposable element-based markers to identify allelic variations underlying agronomic traits in rice. Plant Commun. 2022;3(3):100270. https://doi.org/10.1016/j.xplc.2021.100270.
37. Han S, Dias GB, Basting PJ, Nelson MG, Patel S, Marzo M, Bergman CM. Ongoing transposition in cell culture reveals the phylogeny of diverse Drosophila S2 sublines. Genetics. 2022;221(3):iyac077. https://doi.org/10.1093/genetics/iyac077.
38. Han S, Dias GB, Basting PJ, Viswanatha R, Perrimon N, Bergman CM. Local assembly of long reads enables phylogenomics of transposable elements in a polyploid cell line. Nucleic Acids Res. 2022;50(21):e124. https://doi.org/10.1093/nar/gkac794.
39. Navarro-Dominguez B, Chang C-H, Brand CL, Muirhead CA, Presgraves DC, Larracuente AM. Epistatic selection on a selfish Segregation Distorter supergene – drive, recombination, and genetic load. eLife. 2022;11:78981. https://doi.org/10.7554/eLife.78981.
40. Hays M, Schwartz K, Schmidtke DT, Aggeli D, Sherlock G. Paths to adaptation under fluctuating nitrogen starvation: The spectrum of adaptive mutations in Saccharomyces cerevisiae is shaped by retrotransposons and microhomology-mediated recombination. PLoS Genet. 2023;19(5):e1010747. https://doi.org/10.1371/journal.pgen.1010747.
41. Hiltunen M, Ament-Velasquez SL, Ryberg M, Johannesson H. Stage-specific transposon activity in the life cycle of the fairy-ring mushroom Marasmius oreades. Proc Natl Acad Sci USA. 2022;119(46):e2208575119. https://doi.org/10.1073/pnas.2208575119.
42. Hemmer LW, Negm S, Geng X, Courret C, Navarro-Dominguez B, Speece I, Wei X, Altidor E, Chaffer J, Sproul JS, Larracuente AM. Centromere-associated retroelement evolution in Drosophila melanogaster reveals an underlying conflict. bioRxiv. 2022;2022–1125518008. https://doi.org/10.1101/2022.11.25.518008.
43. Scarlett VT, Lovell JT, Shao M, Phillips J, Shu S, Lusinska J, Goodstein DM, Jenkins J, Grimwood J, Barry K, Chalhoub B, Schmutz J, Hasterok R, Catalán P, Vogel J.P. Multiple origins, one evolutionary trajectory: gradual evolution characterizes distinct lineages of allotetraploid Brachypodium. Genetics. 2023;223(2):146. https://doi.org/10.1093/genetics/iyac146.

44. Moresi NG, Geck RC, Skophammer R, Godin D, yEvo Students, Taylor M.B, Dunham MJ. Caffeine-tolerant mutations selected through an at-home yeast experimental evolution teaching lab. microPublication Biol. 2023. https://doi.org/10.17912/micropub.biology.000749.
45. Treepong P, Guyeux C, Meunier A, Couchoud C, Hocquet D, Valot B. panISa: ab initio detection of insertion sequences in bacterial genomes from short read sequence data. Bioinformatics. 2018;34(22):3795–800. https://doi.org/10.1093/bioinformatics/bty479.
46. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R. The variant call format and VCFtools. Bioinformatics. 2011;27(15):2156–8. https://doi.org/10.1093/bioinformatics/btr330.
47. Koster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. Bioinformatics. 2012;28(19):2520–2. https://doi.org/10.1093/bioinformatics/bts480.
48. Kofler R, Gomez-Sanchez D, Schlotterer C. PoPoolationTE2: comparative population genomics of transposable elements using pool-seq. Mol Biol Evol. 2016;33(10):2759–64. https://doi.org/10.1093/molbev/msw137.
49. Chen J, Wrightsman TR, Wessler SR, Stajich JE. RelocaTE2: a high resolution transposable element insertion site mapping tool for population resequencing. PeerJ. 2017;5:2942. https://doi.org/10.7717/peerj.2942.
50. Schauer SN, Carreira PE, Shukla R, Gerhardt DJ, Gerdes P, Sanchez-Luque FJ, Nicoli P, Kindlova M, Ghisletti S, Santos AD, Rapoud D, Samuel D, Faivre J, Ewing AD, Richardson SR, Faulkner GJ. L1 retrotransposition is a common feature of mammalian hepatocarcinogenesis. Genome Res. 2018;28(5):639–53. https://doi.org/10.1101/gr.226993.117.
51. Adrion JR, Song MJ, Schrider DR, Hahn MW, Schaack S. Genome-wide estimates of transposable element insertion and deletion rates in Drosophila melanogaster. Genome Biol Evol. 2017;9(5):1329–40. https://doi.org/10.1093/gbe/evx050.
52. Yu T, Huang X, Dou S, Tang X, Luo S, Theurkauf W, Lu J, Weng Z. A benchmark and an algorithm for detecting germline transposon insertions and measuring de novo transposon insertion frequencies. Nucleic Acids Res. 2021;49(8):44. https://doi.org/10.1093/nar/gkab010.
53. Gruning B, Dale R, Sjodin A, Chapman BA, Rowe J, Tomkins-Tinch CH, Valieris R, Koster J. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat Methods. 2018;15(7):475–6. https://doi.org/10.1038/s41592-018-0046-7.
54. Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics. 2016;32(19):3047–8. https://doi.org/10.1093/bioinformatics/btw354.
55. Weilguny L, Kofler R. DeviaTE: Assembly-free analysis and visualization of mobile genetic element composition. Mol Ecol Resour. 2019;19(5):1346–54. https://doi.org/10.1111/1755-0998.13030.
56. Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. Bioinformatics. 2012;28(4):593–4. https://doi.org/10.1093/bioinformatics/btr708.
57. Li H. wgsim. https://github.com/lh3/wgsim. Accessed 10 July 2023.
58. Gafner J, Philippsen P. The yeast transposon Ty1 generates duplications of target DNA on insertion. Nature. 1980;286(5771):414–8. https://doi.org/10.1038/286414a0.
59. Rinckel LA, Garfinkel DJ. Influences of histone stoichiometry on the target site preference of retrotransposons Ty1 and Ty2 in Saccharomyces cerevisiae. Genetics. 1996;142(3):761–76. https://doi.org/10.1093/genetics/142.3.761.
60. Chalker DL, Sandmeyer SB. Transfer RNA genes are genomic targets for de novo transposition of the yeast retrotransposon Ty3. Genetics. 1990;126(4):837–50.
61. Kim JM, Vanguri S, Boeke JD, Gabriel A, Voytas DF. Transposable elements and genome organization: a comprehensive survey of retrotransposons revealed by the complete Saccharomyces cerevisiae genome sequence. Genome Res. 1998;8(5):464–78.
62. Carr M, Bensasson D, Bergman CM. Evolutionary genomics of transposable elements in Saccharomyces cerevisiae. PLoS ONE. 2012;7(11):50978. https://doi.org/10.1371/journalpone0050978.
63. Lex A, Gehlenborg N, Strobelt H, Vuillemot R, Pfister H. UpSet: Visualization of Intersecting Sets. IEEE Trans Vis Comput Graph. 2014;20(12):1983–92. https://doi.org/10.1109/TVCG.2014.2346248.
64. Strope PK, Skelly DA, Kozmin SG, Mahadevan G, Stone EA, Magwene PM, Dietrich FS, McCusker JH. The 100-genomes strains, an S. cerevisiae resource that illuminates its natural phenotypic and genotypic variation and emergence as an opportunistic pathogen. Genome Res. 2015;25(5):762–774. https://doi.org/10.1101/gr.185538.114.
65. Li H. seqtk. https://github.com/lh3/seqtk. Accessed 10 July 2023.
66. Peter J, Chiara MD, Friedrich A, Yue J-X, Pflieger D, Bergstrom A, Sigwalt A, Barre B, Freel K, Llored A, Cruaud C, Labadie K, Aury J-M, Istace B, Lebrigand K, Barbry P, Engelen S, Lemainque A, Wincker P, Liti G, Schacherer J. Genome evolution across 1,011 Saccharomyces cerevisiae isolates. Nature. 2018;556(7701):339–44. https://doi.org/10.1038/s41586-018-0030-5.
67. SRAtools. https://github.com/ncbi/sra-tools. Accessed 10 July 2023.
68. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26(6):841–2. https://doi.org/10.1093/bioinformatics/btq033.
69. Oberbeckmann E, Wolff M, Krietenstein N, Heron M, Ellins JL, Schmid A, Krebs S, Blum H, Gerland U, Korber P. Absolute nucleosome occupancy map for the Saccharomyces cerevisiae genome. Genome Res. 2019;29(12):1996–2009. https://doi.org/10.1101/gr.253419.119.
70. Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009;10(3):25. https://doi.org/10.1186/gb-2009-10-3-r25.
71. Quintales L, Vázquez E, Antequera F. Comparative analysis of methods for genome-wide nucleosome cartography. Brief Bioinform. 2015;16(4):576–87. https://doi.org/10.1093/bib/bbu037.
72. Kent WJ, Zweig AS, Barber G, Hinrichs AS, Karolchik D. BigWig and BigBed: enabling browsing of large distributed datasets. Bioinformatics. 2010;26(17):2204–7. https://doi.org/10.1093/bioinformatics/btq351.
73. Pohl A, Beato M. bwtool: a tool for bigWig files. Bioinformatics. 2014;30(11):1618–9. https://doi.org/10.1093/bioinformatics/btu056.
74. Wickham H. Ggplot2: Elegant Graphics for Data Analysis. Springer; 2016. https://ggplot2.tidyverse.org. Accessed 10 July 2023.
75. Conway JR, Lex A, Gehlenborg N. UpSetR: an R package for the visualization of intersecting sets and their properties. Bioinformatics. 2017;33(18):2938–40. https://doi.org/10.1093/bioinformatics/btx364.
76. Qi X, Daily K, Nguyen K, Wang H, Mayhew D, Rigor P, Forouzan S, Johnston M, Mitra RD, Baldi P, Sandmeyer S. Retrotransposon profiling of RNA polymerase III initiation sites. Genome Res. 2012;22(4):681–92. https://doi.org/10.1101/gr.131219.111.
77. Baller JA, Gao J, Stamenova R, Curcio MJ, Voytas DF. A nucleosomal surface defines an integration hotspot for the Saccharomyces cerevisiae Ty1 retrotransposon. Genome Res. 2012;22(4):704–13. https://doi.org/10.1101/gr.129585.111.
78. Bridier-Nahmias A, Lesage P. Two large-scale analyses of Ty1 LTR-retrotransposon de novo insertion events indicate that Ty1 targets nucleosomal DNA near the H2A/H2B interface. Mob DNA. 2012;3(1):22. https://doi.org/10.1186/1759-8753-3-22.
79. Mularoni L, Zhou Y, Bowen T, Gangadharan S, Wheelan SJ, Boeke JD. Retrotransposon Ty1 integration targets specifically positioned asymmetric nucleosomal DNA segments in tRNA hotspots. Genome Res. 2012;22(4):693–703. https://doi.org/10.1101/gr.129460.111.
80. Patterson K, Shavarebi F, Magnan C, Chang I, Qi X, Baldi P, Bilanchone V, Sandmeyer SB. Local features determine Ty3 targeting frequency at RNA polymerase III transcription start sites. Genome Res. 2019;29(8):1298–309. https://doi.org/10.1101/gr.240861.118.
81. Bleykasten-Grosshans C, Friedrich A, Schacherer J. Genome-wide analysis of intraspecific transposon diversity in yeast. BMC Genomics. 2013;14:399. https://doi.org/10.1186/1471-2164-14-399.
82. Bleykasten-Grosshans C, Fabrizio R, Friedrich A, Schacherer J. Species-wide transposable element repertoires retrace the evolutionary history of the Saccharomyces cerevisiae host. Mol Biol Evol. 2021;38(10):4334–45. https://doi.org/10.1093/molbev/msab171.
83. Roeder GS, Farabaugh PJ, Chaleff DT, Fink GR. The origins of gene instability in yeast. Science. 1980;209(4463):1375–80. https://doi.org/10.1126/science.6251544.
84. Bonnet A, Lesage P. Light and shadow on the mechanisms of integration site selection in yeast Ty retrotransposon families. Curr Genet. 2021;67(3):347–57. https://doi.org/10.1007/s00294-021-01154-7.
85. Bridier-Nahmias A, Tchalikian-Cosson A, Baller JA, Menouni R, Fayol H, Flores A, Saïb A, Werner M, Voytas DF, Lesage P. An RNA polymerase III subunit determines sites of retrotransposon integration. Science. 2015;348(6234):585–588. https://doi.org/10.1126/science.1259114.

86. Asif-Laidin A, Conesa C, Bonnet A, Grison C, Adhya I, Menouni R, Fayol H, Palmic N, Acker J, Lesage P. A small targeting domain in Ty1 integrase is sufficient to direct retrotransposon integration upstream of tRNA genes. EMBO J. 2020;39(17):104337. https://doi.org/10.15252/embj.2019104337.
87. Nguyen PQ, Huecas S, Asif-Laidin A, Plaza-Pegueroles A, Capuzzi B, Palmic N, Conesa C, Acker J, Reguera J, Lesage P, Fernández-Tornero C. Structural basis of Ty1 integrase tethering to RNA polymerase III for targeted retrotransposon integration. Nat Commun. 2023;14(1):1729. https://doi.org/10.1038/s41467-023-37109-4.
88. Gruening B, Sallou O, Moreno P, Leprevost FdV, Menager H, Søndergaard D, Rost H, Sachsenberg T, O'Connor B, Madeira F, Angel VDD, Crusoe MR, Varma S, Blankenberg D, Jimenez RC, Community B, Perez-Riverol Y. Recommendations for the packaging and containerizing of bioinformatics software. F1000Research. 2019;7(ELIXER):742. https://doi.org/10.12688/f1000research.15140.2.
89. Gardner EJ, Lam VK, Harris DN, Chuang NT, Scott EC, Pittard WS, Mills RE, Consortium TGP, Devine SE. The Mobile Element Locator Tool (MELT): population-scale mobile element discovery and biology. Genome Res. 2017;27(11):1916–29. https://doi.org/10.1101/gr.218032.116.
90. Chen P, Zhang J. Asexual experimental evolution of yeast does not curtail transposable elements. Mol Biol Evol. 2021;38(7):2831–42. https://doi.org/10.1093/molbev/msab073.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

BMC