Genome-wide mapping of individual replication fork velocities using nanopore sequencing

Bertrand Theulot, Laurent Lacroix, Jean-Michel Arbona, Gael A Millot, Etienne Jean, Corinne Cruaud, Jade Pellet, Florence Proux, Magali Hennion, Stefan Engelen, et al.

HAL Id: pasteur-03693859

https://pasteur.hal.science/pasteur-03693859

Submitted on 13 Jun 2022

Genome-wide mapping of individual replication fork velocities using nanopore sequencing

Bertrand Theulot [1,2,9], Laurent Lacroix [1,9✉], Jean-Michel Arbona [3], Gael A. Millot [4], Etienne Jean[1], Corinne Cruaud[5], Jade Pellet[1], Florence Proux[1], Magali Hennion [6], Stefan Engelen [7], Arnaud Lemainque[5], Benjamin Audit [8], Olivier Hyrien [1✉] & Benoît Le Tallec [1✉]

Little is known about replication fork velocity variations along eukaryotic genomes, since reference techniques to determine fork speed either provide no sequence information or suffer from low throughput. Here we present NanoForkSpeed, a nanopore sequencing-based method to map and extract the velocity of individual forks detected as tracks of the thymidine analogue bromodeoxyuridine incorporated during a brief pulse-labelling of asynchronously growing cells. NanoForkSpeed retrieves previous *Saccharomyces cerevisiae* mean fork speed estimates (≈2 kb/min) in the BT1 strain exhibiting highly efficient bromodeoxyuridine incorporation and wild-type growth, and precisely quantifies speed changes in cells with altered replisome progression or exposed to hydroxyurea. The positioning of >125,000 fork velocities provides a genome-wide map of fork progression based on individual fork rates, showing a uniform fork speed across yeast chromosomes except for a marked slowdown at known pausing sites.

[1] Institut de Biologie de l'Ecole Normale Supérieure (IBENS), Ecole Normale Supérieure, CNRS, INSERM, Université PSL, 46 rue d'Ulm, F-75005 Paris, France. [2] Sorbonne Université, Collège Doctoral, F-75005 Paris, France. [3] Laboratoire de Biologie et Modélisation de la Cellule, Ecole Normale Supérieure de Lyon, CNRS, UMR5239, INSERM, U1293, Université Claude Bernard Lyon 1, 46 allée d'Italie, F-69364 Lyon, France. [4] Institut Pasteur, Université Paris Cité, Bioinformatics and Biostatistics Hub, F-75015 Paris, France. [5] Genoscope, Institut de biologie François-Jacob, Commissariat à l'Energie Atomique (CEA), Université Paris-Saclay, Evry, France. [6] Université Paris Cité, Epigenetics and Cell Fate, UMR7216, CNRS, Paris 75013, France. [7] Génomique Métabolique, Genoscope, Institut François Jacob, CEA, CNRS, Univ. Evry, Université Paris-Saclay, 91057 Evry, France. [8] ENSL, CNRS, Laboratoire de physique, F-69342 Lyon, France. [9] These authors contributed equally: Bertrand Theulot, Laurent Lacroix. ✉email: laurent.lacroix@inserm.fr; hyrien@bio.ens.psl.eu; letallec@bio.ens.psl.eu

Efficient genome duplication in eukaryotes depends on the proper progression of multiple replication forks. Perturbations of fork movement lead to fork stalling and collapse, generating genomic instability that likely drives cancer development[1]. Despite this pivotal role, the determinants of fork progression are still elusive.

Single-molecule (SM) analyses of DNA replication by DNA fibre autoradiography and its fluorographic evolutions have revealed relatively constant mean fork speeds (1–2 kb/min) in eukaryotic cells together with a broad dispersion of individual fork velocities (0.5–4.0 kb/min) within a given cell line[2–4] and even within a single cell[5], suggesting large fluctuations of replication fork speed along eukaryotic genomes. When combined with fluorescence in situ hybridization (FISH) of DNA probes, DNA fibre studies can unveil fork progression within specific regions[6]. In some cases no marked differences between local and genome-wide speeds were observed (e.g., refs. [6–9]), but slower forks along the pericentromeric and centromeric portions of human chromosomes[10] and faster forks in long transcribed genes in chicken cells[9] have been reported. However, only a handful of loci could be analysed given the excessively low throughput of such approaches. To map replication genome-wide, Raghuraman and colleagues determined the mean replication timing (RT) profile of the entire *S. cerevisiae* genome and interpreted the slopes connecting peaks and valleys (i.e., regions of initiation and termination of DNA replication, respectively) as a proxy for local, population-averaged fork velocities, calculating a broad range of speeds depending on genomic location[11]. In sharp contrast, a time-course monitoring of replisome progression found that population-averaged fork velocity is homogeneous throughout yeast chromosomes[12]. It therefore remains unclear whether forks travel at variable or constant speed along eukaryotic genomes.

We have recently developed FORK-seq, a high-throughput, high-resolution, SM-based replication mapping technique relying on the detection by nanopore sequencing of 5-bromo-2'-deoxyuridine (BrdU), a thymidine analogue incorporated in replicating DNA[13]. Here we introduce NanoForkSpeed (NFS), a method capable of positioning, orienting and extracting the velocity of replication forks from BrdU tracks synthesized during a brief pulse-labelling of asynchronously growing cells. NFS allows the determination of the speed of single forks with unprecedented spatial and temporal resolutions, with an unparalleled throughput, and with remarkable simplicity in terms of sample preparation and analysis. Thanks to NFS, we generated in *S. cerevisiae* a genome-wide map of fork progression based on the measurement of individual fork velocities.

Results

Detection of BrdU incorporated into replicating DNA. We previously developed the RepNano software to detect BrdU on sequencing reads produced by Oxford Nanopore Technologies (ONT) devices[13]. Here we anchored our BrdU detection process in ONT's Megalodon program (https://github.com/nanoporetech/megalodon) and estimated BrdU incorporation probability at each thymidine site rather than over 96 bp windows as in RepNano[13] (see the 'Methods' section). Megalodon combines Guppy (ONT) GPU-accelerated basecalling and read mapping to the reference genome into one single step, making it a fast, straightforward and easy-to-use pipeline. Its neural network was trained using nanopore reads of genomic DNA from thymidine-auxotroph MCM869 yeast cells[14] grown with 11 different proportions of BrdU in the culture medium, from 0 to 100% with increments of 10% ('Methods'). We first tested the ability of Megalodon to recover from nanopore reads the total BrdU content of these genomic DNA samples determined by mass spectrometry

(Supplementary Fig. 1a). RepNano[13] and other published BrdU basecallers, DNAscent[15] and DNAscent v2[16], were also assessed for comparison. Megalodon exhibited the lowest background signal without BrdU and most closely paralleled the perfect correlation line, despite a slight tendency to underestimate BrdU content (Supplementary Fig. 1a). Megalodon also gave the most balanced estimates of BrdU proportion per 1 kb window over the entire range of tested BrdU contents, including the narrowest peaks of null BrdU content corresponding to parental DNA, showing again the lowest background of all basecallers (Supplementary Fig. 1b).

BT1, a yeast strain with optimised BrdU incorporation and wild-type growth. The budding yeast *S. cerevisiae* lacks a thymidine salvage pathway and is therefore unable to incorporate exogenous thymidine into DNA. Several strategies have been adopted to reconstitute this pathway in vivo in order to render yeasts amenable to DNA labelling with thymidine analogues. They converged towards the combined expression of human equilibrative nucleoside transporter 1 (hENT1), which improves exogenous thymidine uptake, and of Herpes simplex virus thymidine kinase (hsvTK) allowing the conversion of thymidine into thymidine monophosphate (dTMP) (e.g., refs. [17–19]). Additional inactivation of the thymidylate synthase-encoding *CDC21* gene that is essential for de novo dTMP biosynthesis resulted in strains entirely dependent on external thymidine, or its analogues, for growth[14,20]. We previously analysed the replication of the yeast genome by FORK-seq using one such strain, namely MCM869[13,14]. BrdU incorporation is very high in MCM869 cells due to the absence of competing intracellular thymidine, enabling the detection of replication tracks synthesized during short pulse-chase experiments[13]. However, MCM869 yeasts grew at a reduced rate compared to wild-type (WT) cells even at saturating thymidine concentration (Supplementary Fig. 2a, b). To examine fork progression in WT conditions, we engineered the BT1 strain that retained a functional de novo pathway, and therefore had WT growth properties (Supplementary Fig. 2a–c), while containing codon-optimised *hsvTK* and *hENT1* genes for maximal protein expression and BrdU incorporation in yeast cells. BrdU content profiles of nanopore-sequenced genomic DNA from asynchronously growing BT1 cells pulsed with 100 μM BrdU for 2 min followed by a 20 min chase with 1 mM thymidine showed that BrdU incorporation was of comparable efficiency as in MCM869 cells (Supplementary Fig. 2d). In contrast, replication tracks were hardly detectable on pulse-labelled DNA from thymidine-prototroph, BrdU-incorporating strains constructed using currently available hsvTK and hENT1 integrative vectors[17,19] (BT2 and BT3 strains, Supplementary Fig. 2d). BT1 thus constitutes a potent tool to examine fork progression in physiologically relevant conditions.

Fork orientation and fork speed measurement by NFS. Typical BrdU signals in sequencing reads of genomic DNA from pulse-labelled BT1 cells (Fig. 1a) displayed an asymmetrical shape, consisting of a steep ascending slope starting from a segment of null BrdU content (BrdU signal increasing from 0 to ≈0.5), followed by a shallower decreasing slope (from ≈0.5 to ≈0.1) (Fig. 1b). These signals resembled those from MCM869 cells (Supplementary Fig. 2d), previously demonstrated to correspond to elongating replication forks, with the steep and shallow slopes reflecting BrdU incorporation during the pulse and the chase, respectively, and the signal asymmetry revealing fork direction[13]. Since fork speed can be measured as the length of the labelled track divided by the corresponding labelling time, we developed a pipeline named NFS to capture the track length synthesized during the BrdU pulse, that is the section between the starting
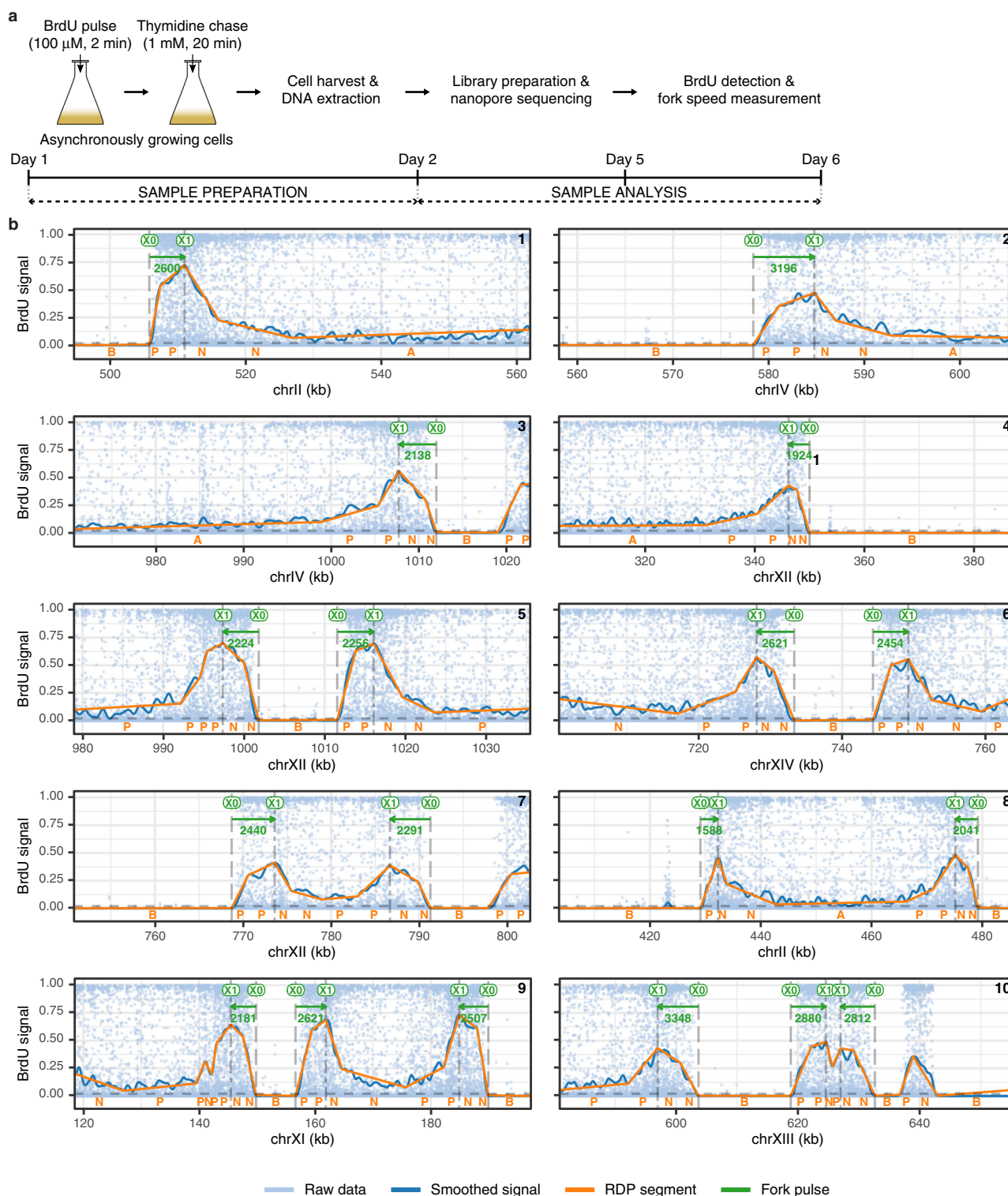
Fig. 1 Replication fork speed measurement procedure by NFS. a Scheme of the protocol for BrdU pulse-labelling of DNA replication in BT1 cells. The usual timeline is indicated. **b** BrdU content profiles of nanopore sequencing reads of genomic DNA from pulse-labelled BT1 cells processed by NFS. Panels show typical replication signals, namely rightward, leftward, diverging and converging forks. Light blue dots, raw data from Megalodon (dots represent the probability of BrdU at each thymidine position); blue curve, smoothed signal; orange lines, segments resulting from the piecewise linear simplification method using the Ramer-Douglas-Peucker algorithm (RDP) to detect and orient BrdU tracks (B, flat segments with background BrdU level; A, flat segments with a BrdU level above background; P, segments with a positive slope; N, segments with a negative slope); X0, estimated position of the start of BrdU incorporation; X1, estimated position of the start of the thymidine chase; green arrow, fork direction, with fork velocity (bp/min, in green) indicated below.

and ending points of the steep slope. A piecewise linear simplification method first converted reads into a sequence of segments classified into 4 categories based on their slope and mean BrdU signal: (i) flat segments with background BrdU level (B); (ii) flat segments with a BrdU level above background (A); (iii) segments with a positive slope (P); and (iv) segments with a negative slope (N) (Fig. 1b). We then identified two patterns for elongating forks depending on their direction. Rightward forks were preceded by a B segment which replicated before the BrdU pulse, then consisted in one or successive P segments, occasionally interrupted by A segments owing to noise, and followed by at least one N segment corresponding to DNA replicated during the chase (Fig. 1b, reads 1, 2). Consequently, we used a regular expression procedure to search for the "BP(P|A)*N+" pattern. Leftward forks were recognized with the symmetrical pattern "P+(N|A)*NB" (Fig. 1b, reads 3, 4). This procedure excluded both incomplete replication tracks (Fig. 1b, read 3, position 1020 kb, and read 7, position 800 kb) and symmetrical signals due to pairs of forks initiated after the start of the pulse or terminated before the end of the pulse, for which the actual labelling time could not be precisely estimated (Fig. 1b, read 10, position 640 kb). The precision of track detection and orientation by NFS was confirmed by (i) a virtually null false-positive rate (≈50 mapped forks per 10 Gb of DNA with no BrdU labelling versus ≈15,000 mapped forks per 10 Gb of pulse-labelled DNA; Supplementary Data 1); (ii) the remarkable similarity of BT1 replication fork directionality (RFD) profile aggregating oriented replication tracks detected by NFS with MCM869 RFD profiles computed by FORK-seq or Okazaki fragment sequencing[13] (Spearman's pairwise correlation coefficients of 0.83 and 0.88, respectively, Supplementary Fig. 3a, b) and (iii) the spatial coincidence between known yeast origins[21], upward slopes (i.e., initiation regions) on BT1 RFD profile and individual initiation sites defined as the midpoints between diverging forks (Supplementary Fig. 3a, c).

For every read, NFS yielded the coordinates of each of the B/P/A/N segments decomposing replication tracks. The length of DNA replicated during the BrdU pulse was then readily computed as the distance between the B/P (X0, start of BrdU incorporation) and (P|A)/N transitions (X1, start of the thymidine chase) for rightward forks and between N/B (X0) and P/(A|N) (X1) transitions for leftward forks (Fig. 1b). Fork velocity was subsequently calculated as the ratio between track length and BrdU pulse duration. The latter was set to 2 min, which was sufficient for detection of replication signals while brief enough to maximize the probability of obtaining complete tracks along nanopore reads averaging ≈15 kb in length. We first conducted a pilot experiment in which BT1 cells were pulsed with BrdU doses ranging from 10 μM to 1 mM and chased with a ten-fold excess of thymidine to determine the optimal labelling conditions. As anticipated, the maximal BrdU signal amplitude rose with increasing BrdU concentrations, whereas similar velocities were computed regardless of the BrdU dose (Supplementary Fig. 4). This indicated that our pipeline was functional for a broad range of signal amplitudes and that fork speed measurement by NFS was independent of BrdU concentration in the tested range. However, in line with reports that high doses of BrdU are toxic for TK-expressing S. cerevisiae strains[20,22], we found that BrdU concentrations over 100 μM both slowed down BT1 S phase and triggered checkpoint activation (Supplementary Fig. 5). Although these doses did not visibly impact fork speed (Supplementary Fig. 4), probably because of the very limited exposure to BrdU during the 2 min pulse, we opted for a pulse concentration of 100 μM BrdU.

Validation of fork speed measurement by NFS. We performed multiple independent pulse-labelling experiments using the aforementioned conditions (Fig. 2a). Thousands (tens of thousands) of individual measurements were typically collected in a single run using the ONT MinION (PromethION) device. All experiments yielded similar fork speed distributions and mean velocities between 2045 and 2206 bp/min (average of 2128 bp/min, Fig. 2a), emphasizing the reproducibility of our analysis. Above all, these values are in excellent agreement with previous estimates of ≈2 kb/min in S. cerevisiae cells grown at 30 °C[23–26]. NFS also found that the mean fork speed was 1.7 kb/min in cells grown at 25 °C, which again agreed very well with the average progression rate of 1.6 kb/min determined at this temperature in a preceding study[12] (Fig. 2b).

Using simulated replication forks to evaluate NFS accuracy. To further gauge NFS, we generated in silico reads with BrdU signals mimicking replication forks travelling at a defined velocity ('Methods') and tested NFS ability to recover these velocities. Simulated forks had similar amplitude range, signal dynamics and noise as those from authentic reads of 2-min, 100 μM BrdU pulse-labelled BT1 DNA (Supplementary Fig. 6). In addition, they displayed the same speed distribution as that of real sequencing data to allow the comparison of our estimates after NFS processing to a known ground truth under the conditions of an actual experiment ('Methods'). NFS measurements were extremely precise on simulated reads bearing one or multiple forks in the absence of signal noise (median error of 7 and 30 bp/min and interquartile range (IQR) of 123 and 186 bp/min for single and multiple forks, respectively; Fig. 3a). Importantly, although speed error distributions broadened when NFS was confronted to simulated forks with noise closely resembling experimental data (IQR of 371 and 438 bp/min for single and multiple forks, respectively; Fig. 3a), the median error remained remarkably low (−32 and −11 bp/min for single and multiple forks, respectively; Fig. 3a), demonstrating that the overall evaluation of fork velocity by NFS was extremely accurate. We noted that NFS tended to overestimate and underestimate slow and fast forks, respectively, and that the dispersion of NFS measurements increased with fork speed (Supplementary Fig. 7). However, the median speed error remained below 10% relative to theoretical velocities in the 1000 to 3000 bp/min range, which encompasses the vast majority of physiological fork rates (Supplementary Fig. 7).

Estimating the true fork speed distribution in yeast. The distribution of fork velocities determined by NFS in BT1 cells is an approximation of the genuine distribution of individual fork speeds in yeast owing to signal noise and measurement errors. Since we could characterize how NFS responded to fork velocities of known value thanks to simulated reads, we adopted a deconvolution strategy to predict the original fork speed distribution from our >125,000 experimental measurements at 30 °C ('Methods'). Interestingly, the main peak of our estimate of the true speed distribution consisted of values in the 2486 ± 150 bp/min range that accounted for two-thirds of all fork velocities (Fig. 3b), suggesting a globally low dispersion of individual fork speeds in S. cerevisiae. Finally, to evaluate the deconvolution procedure, we generated 100,000 reads bearing forks simulated on the basis of the deconvolved true fork speed distribution and analysed them with NFS; the obtained distribution closely approximated the experimental one (Fig. 3c), validating our approach.

NFS retrieves expected shifts in fork speed. We next investigated if NFS was able to detect changes in replication fork velocity in conditions known to alter fork progression (Fig. 4). We first performed pulse-chase experiments in cells exposed to hydroxyurea (HU), a commonly used drug inducing replication fork
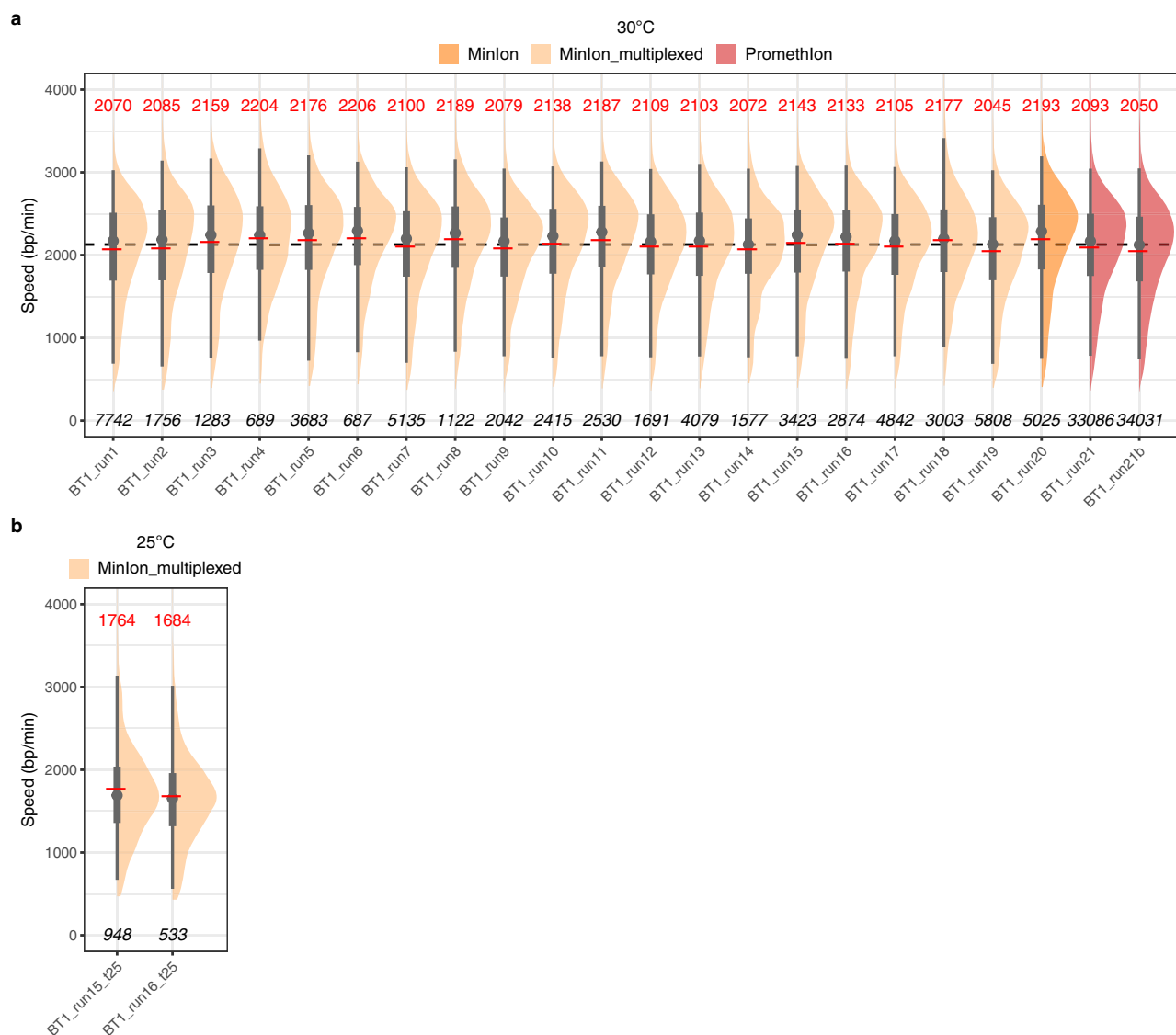
Fig. 2 Measurement of replication fork speed in yeast by NFS. a, b Replication fork speed in BT1 cells grown at 30 °C (**a**) or 25 °C (**b**). Half-eye plots of individual fork velocities detected by NFS on sequencing reads from MinION, multiplexed MinION or PromethION runs performed on pulse-labelled DNA from independent BT1 cell cultures (the two PromethION samples are technical replicates). Red line, mean speed, value indicated in red on top; grey dot, median speed; thick and thin grey vertical lines, 50 and 95% intervals, respectively; bottom, number of individual fork speed measurements; dotted line in **a**, average fork speed of all samples (2128 bp/min). The name of the sequencing run is indicated below each plot; detailed run information is presented in Supplementary Data 1.

slowdown. As expected, we observed a gradual decrease in fork speed with increasing HU concentrations (Fig. 4a). We could detect fork slowdown induced by doses as low as 1 mM HU, indicating that NFS is a very sensitive tool to reveal replication stress. We then measured fork velocity in cells with mutations impacting replisome progression. We were able to recover the previously reported decrease in fork speed in the absence of Mrc1[25,27–30], Tof1[28,30] and Csm3[30] (Fig. 4b). Interestingly, fork speed was equally reduced in *tof1Δ* and *csm3Δ* cells, consistent with the fact that these proteins form a complex at the front of the replisome[31,32]. Moreover, fork progression was affected to a lesser extent in those mutants than without Mrc1, as already described[29,30,33]. NFS also detected the acceleration of replication forks observed in the absence of Rtt109 acetyltransferase[26], as well as in *sml1Δ* mutant exhibiting increased dNTP pools[34] (Fig. 4b). Altogether, these results demonstrate that NFS can precisely quantify fork speed variations both in physiological and perturbed conditions.

Genome-wide mapping of replication fork progression. NFS allowed us to generate a genome-wide map of replication fork progression based on individual fork velocities (Fig. 5 and Supplementary Fig. 8). In order to distinguish meaningful changes in local fork speed distribution, the experimental map was superimposed to a randomized map where fork locations remained unchanged but velocity values were shuffled. This showed that fork movement was largely uniform across the yeast genome, in agreement with the results obtained by Sekedat and colleagues[12], although several loci clearly stood out. We next performed a multi-scale statistical analysis to precisely identify regions of significantly lower or higher fork speed than the bulk genome along with the underlying factors likely responsible for the observed alterations. We detected several regions of fork slowdown but only one locus, located between the *FLO10* and *NFT1* genes on chromosome XI, showed fork acceleration (Fig. 5 and Supplementary Fig. 9). Salient features inside slow regions narrowed down to 1 kb included centromeres, telomeres, tRNA
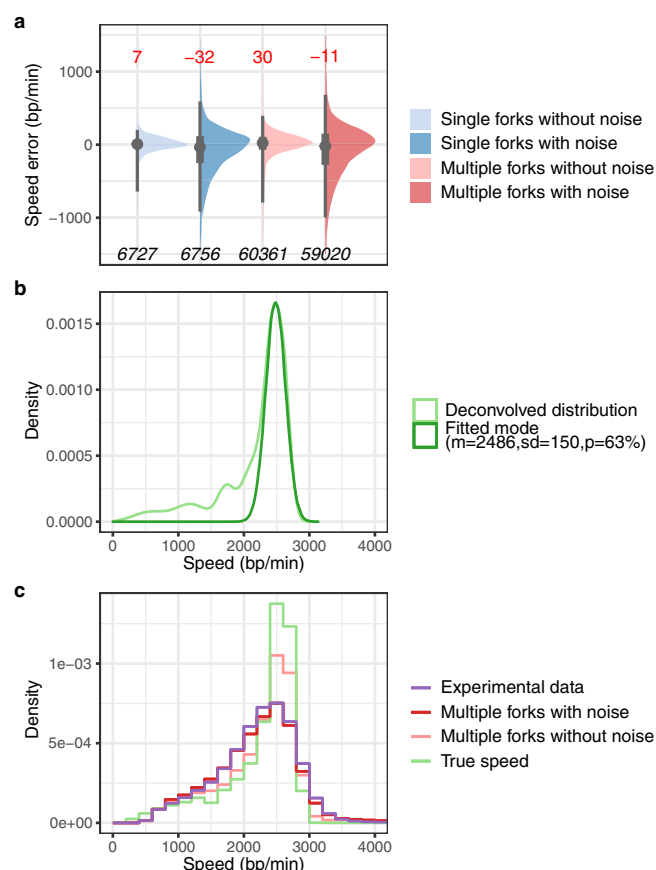
Fig. 3 Evaluation of NFS accuracy and estimation of the true fork speed distribution in yeast using simulated replication forks. a Half-eye plots showing the distribution of measurement errors made by NFS on simulated forks. Fork velocities were determined by NFS on simulated reads containing either a single or multiple forks of known speed, with or without noise, and differences between NFS measurements and true speeds were computed (see 'Methods'). Grey dot, median speed error, value indicated in red on top; thick and thin grey vertical lines, 50% and 95% intervals, respectively; bottom, number of measurements. **b** Deconvolved distribution of individual fork speeds in yeast (light green curve) and Gaussian distribution fitting its main peak (green curve). m, mean speed (bp/min); sd, standard deviation (bp/min); p, percentage (correction factor to fit the Gaussian distribution to the deconvolved distribution). See text and 'Methods' for details. **c** Histograms showing fork speed distribution on 100,000 reads with one or multiple forks simulated according to the deconvolved true fork speed distribution (True Speed, light green), fork speed distribution after NFS processing of these reads without (pink) or with (red) noise, and the experimental distribution of >125,000 individual fork velocities from BT1 cells at 30 °C (purple).

genes, the rDNA locus and the HML silent origin cluster (Fig. 5 and Supplementary Fig. 8), all known impediments to replication forks[35,36]. Targeted analysis of forks overlapping these DNA elements confirmed that centromeres and telomeres quite consistently exhibited a reduced fork speed (Fig. 6a and Supplementary Fig. 10) and that fork slowdown at the rDNA locus occurred in the vicinity of the replication fork barrier (RFB), as anticipated (Supplementary Fig. 11). Fork progression was globally slower at tRNA genes having a direction of transcription opposing that of forks (Fig. 6b), in agreement with tRNA genes being polar obstacles to replication[35–37], although speed was markedly reduced only at a small subset of tRNA genes (Supplementary Fig. 12). We speculate that those exhibit a particularly strong binding of TFIIIB transcription factor, considered
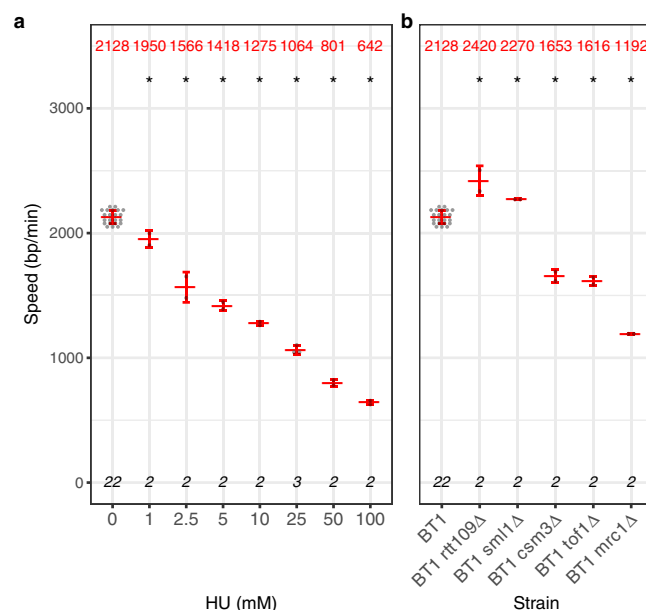
Fig. 4 Measurement of replication fork speed in yeast by NFS in conditions of altered replisome progression. a Replication fork speed in BT1 cells grown in the presence of increasing concentrations of hydroxyurea (HU). **b** Replication fork speed in mutant strains with altered fork progression. **a, b** Grey dot, mean fork speed of a sample; red centre line, average value of all mean fork speeds of the corresponding category, indicated in red on top; red whiskers, standard deviation; bottom, number of samples. Two-sided contrast comparisons between untreated BT1 cells and each HU concentration in **a** and between BT1 cells and each mutant in **b** are indicated by a star ($p \leq 0.05$) or n.s. (not significant). Statistical analyses are detailed in Supplementary Table 2.

as the main impediment to fork movement at tRNA genes[38]. Thanks to the mapping of oriented tracks to Watson or Crick strands, we could distinguish fork progression on the leading and lagging strands. We found that DNA synthesis proceeds significantly faster on the leading strand (Fig. 6c), although the very weak difference in speed with the lagging strand (12 bp/min) is possibly devoid of biological significance. Finally, as it remains unclear whether fork velocity changes during S phase in eukaryotes[2,3,8,39–41], we compared the speed of replication forks with the timing of the region they are replicating[42], noting that forks appear to slightly accelerate during S phase in yeast (Fig. 6d).

Discussion

Under conditions of BrdU incorporation that do not perturb DNA replication, NFS recovers former estimations of fork speed in yeast in a WT-like growing strain and in mutants known to have altered fork progression, with an excellent reproducibility between biological replicates. This validates our measurement procedure and emphasizes the robustness of our method. Moreover, NFS is remarkably efficient in detecting replication fork slowdown, from the slightest to major decelerations, making it a suitable tool to detect replication stress. NFS accuracy has been further verified on simulated forks of known speed, which also offered the opportunity to recreate the presumed true fork velocity distribution in *S. cerevisiae* from the experimental distribution that is inevitably altered by NFS measurement errors, however small they may be. Our results suggest that the majority of fork speeds are narrowly distributed in yeast (two-thirds within 2500 ± 150 bp/min), in apparent contrast with the large dispersion of individual fork velocities observed by conventional SM
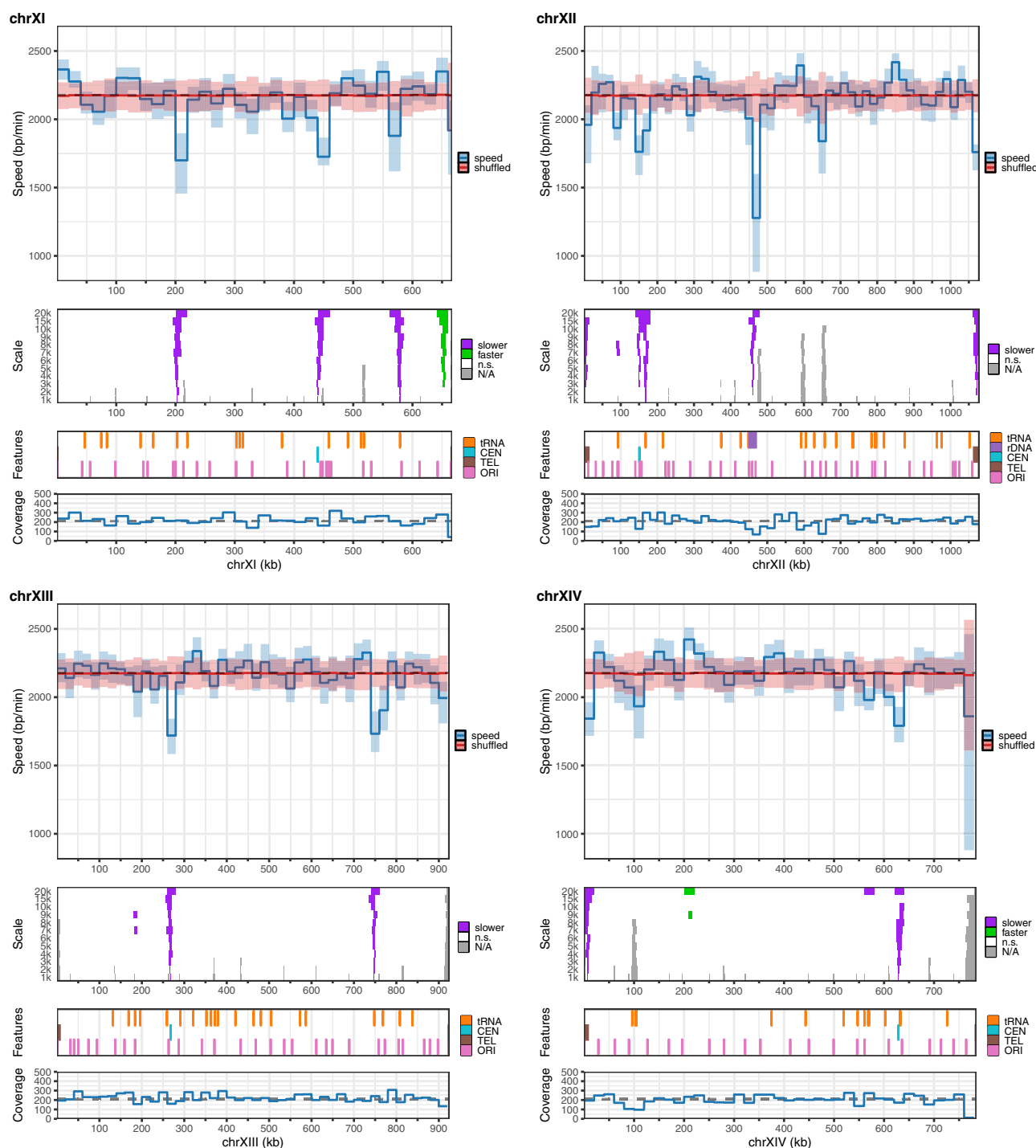
Fig. 5 Replication fork progression map of yeast chromosomes. Shown are chromosomes XI to XIV. Panels from top to bottom: (1) median of experimental fork speeds (blue line) with 98% confidence interval of the median (light blue) and median of reshuffled speeds (red line) with 98% confidence interval of the median (light red) computed in 20 kb windows (dotted line, median fork speed in the whole genome); (2) results of Mann–Whitney-Wilcoxon tests with Holm correction (one-sided) performed along the chromosome to compare the speed distribution in a given window of a given width (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 and 20 kb) to the speed distribution on the whole genome (purple, regions of lower fork speed; green, regions of higher fork speed; white, n.s., not significant; statistical significance was set to $p < 0.01$; grey, N/A, not applicable, regions with no fork); (3) position of selected genomic features (CEN, centromere; TEL, telomeres; ORI, known *S. cerevisiae* replication origins from ref. [21]); (4) coverage of individual replication fork velocities (dotted line, median coverage of the genome). N/A peaks in panel 2 and droughts in coverage in panel 4 essentially map to the position of active origins since fork pairs at initiation sites cannot be resolved by our analysis (see text for details).
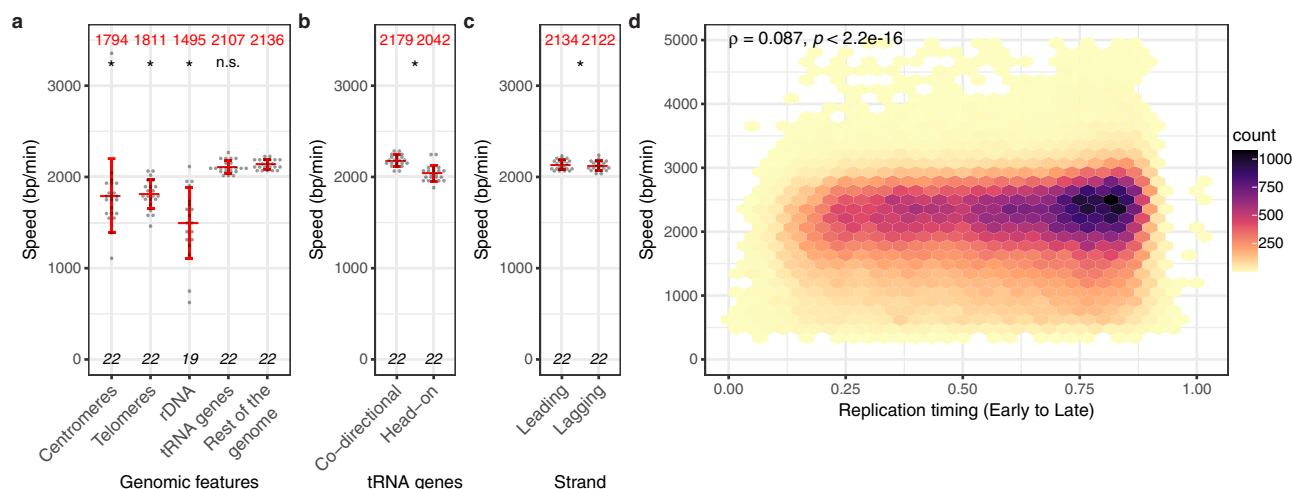
Fig. 6 Analysis of replication fork progression in the yeast genome. a Replication fork speed at selected genomic features. Please note that "telomeres" correspond to subtelomeric sequences and do not comprise the terminal stretch of telomeric repeats. **b** Replication fork speed at tRNA genes with a direction of transcription in the same (Co-directional, 129 genes) or opposite (Head-on, 115 genes) orientation to the main direction of fork progression determined using BT1 RFD profile (see 'Methods' and Supplementary Fig. 3). **c** Replication fork speed on the leading and lagging strands. **a–c** Grey dot, mean speed of forks overlapping the corresponding DNA element (**a**, **b**) or progressing on the leading or lagging strand (**c**) in a sample; red centre line, average value of all mean fork speeds of the corresponding category, indicated in red on top; red whiskers, standard deviation; bottom, number of samples ($n_{rDNA} = 19$ since no fork overlapped the rDNA locus in 3 samples). Two-sided contrast comparisons between a given genomic feature and the rest of the genome in **a**, between co-directional and head-on tRNA genes in **b** and between the leading and lagging strands in **c** are indicated by a star ($p \leq 0.05$) or n.s. (not significant). Statistical analyses are detailed in Supplementary Table 2. **d** Fork speed versus replication timing (data from ref. [42] normalized between 0 and 1 corresponding to the start and end of S phase, respectively) plotted as a 2D density plot using hexagonal bins. Coefficient (ρ) and p-value of Spearman's correlation test (two-sided) between fork speed and replication timing are indicated.

methods in eukaryotes[2–4]. However, it is important to note that although a 10-fold difference between the fastest and slowest forks is commonly found in standard DNA fibre analysis of a given cell line, most fork rates actually fall within less than twofold of the median velocity[4]. For instance, half of the forks travel within ±20% of the 1.5 kb/min median speed in chicken DT40 cells[9]. Extreme values may reflect rare events or may be, at least in part, artefactual. For instance, exceptionally high velocities may originate from the fusion of forks emanating from adjacent origins that fired during the labelling pulse, which may go unnoticed even when using a robust analysis methodology[4]. In that regard, the use of a much shorter labelling pulse in our conditions compared to standard SM techniques minimizes the probability that BrdU tracks result from the activity of more than one fork.

In agreement with a low dispersion of fork speeds in *S. cerevisiae*, the mapping of individual fork velocities shows that the rate of fork progression does not significantly change across the yeast genome except at previously known fork pausing sites[35,36], plus at a single locus situated between *FLO10* and *NFT1* genes where forks appear to move faster for reasons that remain to be determined. We have not attempted here to verify whether low fork speeds around pausing sites result from transient stalling rather than uniformly slow progression, but the former seems likely, especially since fork slowdown at the rDNA locus mapped to the ribosomal RFB pausing site. Our observations of a consistent velocity across the yeast genome support several earlier studies[12,24] and are in line with a very recent nanopore sequencing-based analysis of excised full-length replicons concluding that fork progression is remarkably homogeneous along *S. cerevisiae* chromosomes[43], but they are at odds with those of Raghuraman and colleagues who interpreted variations in RT profile slopes along different parts of the genome as fluctuations in fork speed[11]. However, it has since been demonstrated that RT slopes depend not only on fork speed but also on the local

proportion of rightward and leftward forks in the cell population[8,44,45]. Our results suggest that RT profile slope variations observed in that pioneering study predominantly if not exclusively stem from variations in the proportion of rightward and leftward forks.

Determining fork speed is mandatory in circumstances known or speculated to influence replication dynamics. Existing methods include DNA fibre analysis[2–4,46], hydrodynamic techniques[2,3], slope computation from RT profiles (e.g., refs. [8,11,25,26,47–50]), mathematical modelling of DNA replication (e.g., refs. [24,44]) and time-resolved (i) dense-isotope substitution experiments (for instance, see ref. [51]), (ii) two-dimensional gels[27], (iii) chromatin immunoprecipitation of replisome components[12,23,26,27] or (iv) isolation of BrdU- or 5-ethynyl-2'-deoxyuridine-labelled DNA[52,53]. All these approaches are demanding and difficult to implement, and none of them allows the mapping of individual fork velocities on entire genomes. Not only does NFS have this ability, but it also combines a straightforward protocol with a ready-to-use analytical pipeline (available on GitHub at https://github.com/LacroixLaurent/NanoForkSpeed) enabling an automated measurement of fork speed. NFS should advantageously replace DNA fibre analysis as the "gold standard" method to determine fork velocity as it surpasses it in simplicity, rapidity (estimation of fork speed in 6 days for NFS versus weeks or months for DNA combing or spreading especially when combined with FISH probe detection), number of measurements (thousands of values versus a few tens to a few hundreds) as well as in spatial (precision approaching the nucleotide resolution of sequencing versus precision limited to ≈1 kb by the resolution of optical microscopes) and temporal (fork speed averaged over 2 min versus typically 20 min) resolutions. Estimation of fork velocity by NFS is certainly applicable to BrdU pulse-labelled DNA from other eukaryotes, notably mammalian cells, although a higher sequencing depth will be required to obtain a comparable coverage. The rapid evolution of nanopore sequencing

technologies offering increased throughputs at reduced costs will contribute to overcome this limitation. Meanwhile, the simultaneous processing of multiplexed samples should facilitate the screening of knockouts libraries, particularly in yeast, to uncover new factors involved in replisome progression and broaden our understanding of eukaryotic DNA replication.

Methods

Yeast strains and growth conditions. All strains used in this study are W303 derivatives and are listed in Supplementary Table 1. Standard yeast genetic techniques and media were used[54]. Cells were grown at 30 °C in YPD medium (MP Biomedicals #114001032) unless stated otherwise. BT1 strain was obtained by integrating at the ura3-1 locus of W303 cells StuI-linearized pBL-hsvTK$_{CO}$-hENT1$_{CO}$ plasmid (Supplementary Table 1), a modified version of the p306-BrdU-Inc vector[17] in which both the hsvTK and hENT1 genes were replaced by versions that had been codon-optimized for expression in yeast. hsvTK was codon-optimized for yeast using JCat[55] and synthesized by GeneScript; codon-optimized hENT1[19,56] was a generous gift from E. Schwob (IGMM, Montpellier, France). Detailed cloning procedures are available upon request. The BT2 strain was obtained by reintroducing the WT CDC21 allele encoding the thymidylate synthase into the MCM869 strain[14] by crossing. The BT3 strain was obtained by inserting the regular p306-BrdU-Inc plasmid[17] purchased from Addgene at the ura3-1 locus of W303 cells.

Growth curve and doubling time. Yeasts were grown overnight in YPD (supplemented with 100 μM thymidine for MCM869 cells), diluted in fresh medium (supplemented with 100 μM thymidine for MCM869 cells) at an optical density at 600 nm (OD$_{600}$ ≈ 0.1) and grown for 6 h. OD$_{600}$ was measured every 2 h. Doubling times (T) were estimated on the basis of the growth curves based on the formula $T = [\text{culture duration}*\log(2)]/[\log(\text{final OD}_{600}) - \log(\text{initial OD}_{600})]$.

Analysis of S phase progression by fluorescence-activated cell sorting (FACS). Exponentially growing cells were synchronized in G1 by addition of 0.2 μM α-factor (Sigma #T6901) for 3 h then washed and resuspended in fresh, prewarmed YPD medium containing 50 μg.mL^{-1} pronase (Millipore #53702) to release them into the cell cycle. In the experiment examining the impact of BrdU on S phase progression, BrdU was added 15 min after cell release. Aliquots were taken at regular time intervals and fixed in ethanol. Fixed cells were washed with 50 mM sodium citrate pH 7.4, incubated in sodium citrate buffer supplemented with 0.25 mg.mL^{-1} RNAse A for 1 h at 50 °C, added with 2 mg.mL^{-1} proteinase K and incubated for one additional hour. DNA was counterstained overnight with 1 μM SYTOX Green (Invitrogen #S7020). Samples were analysed using a Beckman Coulter CytoFLEX LX flow cytometer. Data were collected using CytExpert v2.4.0.28 and analysed using FlowJo v10.7.1. Gating strategy is illustrated in Supplementary Fig. 13.

Rad53 immunoblot analysis. Total protein extracts were obtained by trichloroacetic acid (TCA) precipitation, separated by SDS-PAGE on a 7.5% gel and transferred to a nitrocellulose membrane. Rad53 immunoblot was performed with rabbit anti-Rad53 antibody at 1:10,000 (Abcam #104232, batch GR3353005-2), using HRP-conjugated anti-rabbit (Promega #W401B) at 1:5000 as secondary antibody. Detection was performed with SuperSignal West Pico (Thermo Scientific #34078) chemiluminescent reagents. ImageQuant LAS 4000 mini (GE Healthcare, software version 1.3) was used for imaging.

Samples used for neural network training. Neural network training was performed using nanopore-sequenced genomic DNA displaying variable BrdU substitution rates extracted from the thymidine-auxotroph MCM869 strain. For that purpose, MCM869 cells were grown overnight in synthetic complete medium (Dropout Base medium with Complete Supplement Mixture, MP Biomedicals #114025032 and #114500012) with 100 μM thymidine, washed twice to remove thymidine, transferred at OD$_{600}$ ≈ 0.1 into fresh synthetic complete medium supplemented with various mixtures of BrdU and thymidine (0:100; 10:90; 20:80; 30:70; 40:60; 50:50; 60:40; 70:30; 80:20; 90:10 and 100:0) and grown for 24 h. Genomic DNA was isolated by zymolyase, RNAse A, and proteinase K digestion using Genomic DNA Buffer Set (Qiagen #19060) and Qiagen Genomic-tips 20/G (Qiagen #10223) according to the manufacturer's instructions and subsequently subjected to nanopore sequencing.

Mass spectrometry. LC-MS/MS was performed on the 11 samples prepared for neural network training (see above) using a TSQ Quantiva triple quadrupole mass spectrometer (Thermo Scientific) coupled to an UltiMate 3000 XRS HPLC system (Dionex, Thermo Scientific) as described in ref. [13]. Samples were analysed in technical duplicates using the software TraceFinder (Thermo Scientific, version 5.1).

Pulse-chase labelling. In most experiments, BT1 cells and its derivatives as well as BT2 and BT3 strains were grown overnight in YPD, diluted in fresh medium at an OD$_{600}$ ≈ 0.1 and pulsed after 3 doublings (OD$_{600}$ ≈ 0.8) with 100 μM BrdU for 2 min followed by a 20 min incubation with 1 mM thymidine. For the experiment carried out with different BrdU concentrations, BT1 cells were pulsed in the same conditions as above with either 10, 25, 50, 100, 250, 500, 750 μM or 1 mM BrdU and chased with a ten-fold excess of thymidine. For experiments in the presence of HU, cells were grown for 2 doublings (OD$_{600}$ ≈ 0.4) and treated for 1 h with HU prior to BrdU pulse-labelling. MCM869 cells were cultured overnight in YPD supplemented with 100 μM thymidine, diluted in fresh YPD with thymidine at an OD$_{600}$ ≈ 0.8 and grown to OD$_{600}$ ≈ 0.8 before being washed twice with YPD to remove thymidine, transferred to YPD for 30 min, pulsed with 100 μM BrdU for 2 min and chased with 1 mM thymidine for 45 min. Cells were pelleted after the thymidine chase, washed with water before DNA extraction as in ref. [57].

Library preparation and data acquisition. All samples were sequenced using R9.4.1 chemistry flow cells from Oxford Nanopore Technology (ONT). MinION and GridION sequencing libraries were prepared using ligation sequencing kits SQK-LSK108 or SQK-LSK109 (ONT) in combination with ONT EXP-NBD103 Native Barcoding kit or ONT EXP-NBD104 Native Barcoding Expansion 1–12 pack in case of multiplexing according to ONT protocols with the modifications presented in ref. [57]. PromethION sequencing libraries were prepared using the ultra-long DNA sequencing kit SQK-ULK001 (ONT) according to ONT protocols. Data were acquired using MinKNOW (ONT, MinKNOW Core versions 1.14.1 to 4.4.13, Supplementary Data 1) with default parameters. Demultiplexing of barcoded nanopore reads was performed as described in ref. [57].

Training of the BrdU basecalling model. Our model was built in three successive steps. First, we modified the architecture of RepNano convolutional neural network[13] in order to obtain a nucleotide resolution for BrdU detection, creating RepNano v2. We next performed a first training of our model with ONT's Taiyaki from the Megalodon program (version 2.2.9 downloaded from https://nanoporetech.github.io/megalodon/, Guppy version 4.4.1) using RepNano v2 outputs, then trained our model a second time using the outputs of the first training and adding specific false positive BrdU signals to the training dataset in order to reduce background and false-positive signals.

RepNano v2 architecture. We used two convolutional layers with kernel size 7 and filter size 32 with no padding followed by a long short-term memory (LSTM) layer with 32 hidden units that output a vector of the same length as the input, then a convolutional layer with kernel size 1 and filter size 1 and finally an averaging layer outputting the BrdU percent of the segment. The initial length of the input vector was 112 so that after two convolutions without padding the final length was 100. The LSTM layer also produced a 100 × 32 vector and the last convolutional layer produced a 100 × 1 vector that was averaged and compared to the expected output of BrdU content via a logcosh loss. We extracted the vector from the last convolutional layer to basecall BrdU at the nucleotide level.

RepNano v2 training. RepNano v2 was trained using nanopore reads of the 11 genomic DNA samples with various BrdU substitution rates described above (BrdU contents measured by mass spectrometry of 0, 9.4, 16.6, 27.9, 35.1, 46.1, 54.8, 59, 72.6, 78.8 and 80.3%). For each of the 11 samples, composed of a mix of substituted and unsubstituted reads (corresponding to parental DNA), 400 reads were used for the training. To separate substituted reads from unsubstituted ones, we first ran two training cycles with a neural network made of only three convolutional layers (no LSTM and no averaging layers) in order to avoid an overfitting of the data. At the end of the first cycle, the quality of the prediction was good enough to allow the discrimination of substituted from unsubstituted reads for the 40 to 100% BrdU samples. Reads were relabelled as substituted or unsubstituted based on these results, and a second training cycle was run. This allowed the separation of substituted from unsubstituted reads for the 20 and 30% BrdU samples. The final network (RepNano v2 architecture) was trained on this cleaner dataset of 400 reads per sample.

Megalodon model training. We performed a first training of our model with ONT's Taiyaki software on 400 reads from each of the 11 BrdU samples described above basecalled with RepNano v2 using a LSTM architecture referred to as mLstm_cat_mod_flipflop with default parameters. Meanwhile, we selected 100 reads from the 0% BrdU sample where forks had been detected, corresponding to false positives (these were mainly found in the rDNA and at the positions of Ty elements). We also selected 100 reads in the same regions from the 100% BrdU sample not to introduce a bias in the neural network at these locations. These reads were added to the outputs of the first model to perform a second training. Finally, the Taiyaki model was converted into a Guppy model.

BrdU basecalling and read mapping. Basecalling and read mapping were performed from MinKNOW-generated fast5 files upon sequencing using Megalodon, that combines Guppy basecalling (using the model trained as described above to detect BrdU at the nucleotide level) and read mapping to the reference genome (see

basecalling_sample.sh script on NanoForkSpeed GitHub repository). This step was carried out using a GPU-enhanced computer to allow fast processing. As the resulting BAM files could be quite big, a custom R[58] script (R version 4.0.3) named basecalling_sample.r first split BAM files into 50,000 read subfiles. The parsing was then performed in R with the mega_parsing homemade function. For each read, this function imported from the BAM file the read identity (read_id), the DNA strand, the chromosome name, the start position and length of the read, the sequence and the Mm and Ml fields which contain the position and output score of detected BrdUs, respectively, using Samtools (version 1.10). The function then allocated a probability of being a BrdU at every position corresponding to a thymidine in the mapped sequence. All discontinuities (gaps) longer than 100 nucleotides introduced during the mapping step were recorded; this information was used after the fork detection procedure to filter out forks overlapping such gaps as discontinuities in the BrdU signal may interfere with fork speed measurement. Data were then smoothed using a combination of a rolling mean on 100 nucleotides and a Gaussian weighted rolling mean on 2.5 kb, with exclusion of the first and last 2.5 kb windows (which removed reads <5 kb from our analysis). This smoothing procedure was designed to allow the piecewise linear simplification (see below) without altering the shape of the signal too much. The output file contained both the raw and the smoothed signal as well as the mapping information (read_id, chromosome name, start and end positions, strand and gap positions).

BrdU basecaller comparison. For each of the 11 genomic DNA samples with different BrdU substitution rates described above, a set of 8000 nanopore reads were basecalled either with Megalodon or with DNAscent v1[15], DNAscent v2[16], RepNano[13] transition matrices (RepNano_TM) and RepNano convolutional neural network (RepNano_CNN). RepNano_TM and RepNano_CNN were used as in ref. [13]. For each basecaller, the BrdU content was computed in 1 kb bins for every read. The overall BrdU content of a given DNA sample corresponded to the mean of the BrdU content values per 1 kb window. Means of the 11 samples were plotted against BrdU contents measured by mass spectrometry (MS) in Supplementary Fig. 1a. The proportionality between basecaller and MS estimates was assessed using a linear regression and by computing the mean square error between the observed and ideal lines. Distributions of BrdU content values per 1 kb window for the reads from the 11 samples were represented as normalized densities using the geom_freqpoly() function of the ggplot2 R package in Supplementary Fig. 1b–f. Although reads used for the analyses shown in Supplementary Fig. 1 came from a different sequencing run than reads used for the training of Megalodon, we cannot completely exclude that Megalodon's performance might benefit from having been trained on DNA originating from the same samples as those used to compare basecallers.

Fork detection and orientation. After pulse-labelling, BrdU signals corresponding to ongoing replication forks are visualized on nanopore reads as a steep ascending slope starting from a segment of null BrdU content followed by a shallower decreasing slope; the steep and shallow slopes reflect BrdU incorporation during the pulse and the chase, respectively, and this signal asymmetry allows the determination of the direction of fork progression[13]. Fork detection and orientation were performed with custom R scripts using a piecewise linear simplification approach deriving from the original FORK-seq manuscript[13]. Reads were first converted into a series of segments using the Ramer-Douglas-Peucker algorithm[59,60] (Hausdorff distance epsilon = 0.1, using DouglasPeuckerEpsilon function of the kmlShape R package). Only reads with 3 or more segments were kept as reads with less segments could not form a complete fork. Segments were classified into 4 categories using their slope and mean BrdU signal (B = flat segment with a background BrdU level, A = flat segment with a BrdU level above background, P = segment with a positive slope and N = segment with a negative slope). In order to set the background threshold, the distribution of the mean BrdU signal on 1 kb windows computed for the reads containing at least 3 segments of a given experiment was plotted; in every experiment, DNA replicated before the BrdU pulse (i.e., DNA with a theoretical null BrdU signal) was separated from the rest of the DNA by local minimum near 0.02, which was therefore set as the background threshold (b2a.thr parameter) for all samples.

We were specifically interested in capturing the segment corresponding to BrdU incorporation during the pulse, that is the section between the starting and ending points of the steep slope, in order to extract fork speed defined as the length of DNA replicated during the duration of the pulse (i.e., 2 min) divided by the labelling time. We thus focused on pulse segments for which we could determine the start and end positions. We determined two patterns to identify forks depending on their orientation on the nanopore reads. Rightward forks must be preceded by a B segment, then must contain one or several P segments that may be interrupted by A segments owing to noise and then at least one N segment; they must not be directly followed by a B segment as the BrdU level corresponding to the thymidine chase is above background. We then used a regular expression approach to recognize the "BP(P|A)*N+" pattern, excluding forks for which the following segment was a B. Leftward forks were identified thanks to a symmetrical pattern "P+(N|A)*NB", excluding forks for which the preceding segment was a B. This prevented any overlap between forks detected on a given read (forks must start with a B segment and cannot end on a B segment) and excluded incomplete replication tracks as well as symmetrical signals due to pairs of forks initiated after

the start of the pulse or terminated before the end of the pulse, for which the actual labelling time could not be precisely estimated. Furthermore, in case of multiple replication signals on a single read detected as a succession of leftward and rightward forks, we could determine the position of individual initiation (termination) sites defined as the midpoints between diverging (converging) forks.

Our procedure was performed using the homemade NFSmaster function of the NFS_function.r script. It creates an output file saved in the .rds format in which the data are organised as a list of 4 elements: (1) a list of tibble (specialised type of data.frame in the R tidyverse) containing (1.1) the reads filtered according to their length and the presence of 3 or more linearized segments and (1.2) the reads with detected forks; (2) a tibble of all the detected forks; (3) a tibble containing the detected initiation (Ini) and termination (Ter) events; (4) a table summary of different metrics of the experiment. Fork detection was performed on the split data corresponding to the basecalling of 50,000 reads and results were merged using the NFS_merging function of the NFS_function.r file. Merged files have the same organization with a slightly simplified summary table but do not contain reads without fork; they were used to produce the figures and data discussed in this manuscript.

Fork speed analysis. Fork data from every read and experiment were collected in a master data table containing read information (read_id, chromosome name, read start and end positions, read strand), fork parameters ($X0$ = start of BrdU incorporation = position of the B/P and N/B transitions for rightward and leftward forks, respectively; $X1$ = start of the thymidine chase = position of the (P|A)/N and P/(A|N) transitions for rightward and leftward forks, respectively; speed in bp/min = fork speed averaged over the 2 min of the pulse = absolute value of $(X1 - X0)/2$; direction = Left or Right; type = leading or lagging; d.Y = BrdU signal amplitude at $X1$ position minus BrdU signal amplitude at $X0$ position) and experimental parameters (Exp = name of the experiment; B_pulse = BrdU concentration in the medium during the pulse (in μM); t_pulse = duration of the pulse (2 min in all experiments); T_chase = thymidine concentration in the medium during the chase (in μM); temp = growing temperature (in °C); strain = yeast strain used; mutant = wild-type (WT) or name of the inactivated gene; HU = hydroxyurea concentration in the medium (in mM)). Data were filtered using these parameters to group fork speeds according to the criteria presented in the figures of this manuscript. Detailed information for every sample sequenced in our study are presented in Supplementary Data 1.

Computing fork velocity from reads of increasing minimal length resulted in a negligible rise in the median fork speed estimated by NFS, confirming that a 2 min BrdU pulse duration was adequately suited with respect to read length to detect short as well as long tracks in our experiments; the median length of reads with forks is indicated for each sample in Supplementary Data 1.

Genomic map of fork speed. Forks detected in all BT1 biological replicates grown in standard conditions at 30 °C were converted into GenomicRanges using the GenomicRanges R package and reduced to their centre. The velocities of these forks were then binned into non-overlapping windows of different width (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 and 20 kb). The number of forks (coverage), the median speed and the 98% confidence interval of the median (using the MedianCI function of the DescTools R package) were computed for every window. In addition, a Mann–Whitney–Wilcoxon test was performed to compare the speed distribution in a given window to the speed distribution on the whole genome. *P*-values for slower and faster speed were corrected for the multiplicity of testing using the Holm method with the p.adjust function, taking into account the number of windows for all scales. Results were saved as bigwig files and used for plots in Fig. 5 and Supplementary Figs. 8, 9 and 11, setting statistical significance to *p* < 0.01. Bins with no fork were tagged N/A. In order to distinguish meaningful changes in local fork velocity distribution, the experimental map was superimposed to a randomized speed map. To generate this map, fork locations were kept unchanged but speed values were randomized 1000 times and binned into 20 kb (Fig. 5 and Supplementary Fig. 8) or 1 kb (Supplementary Figs. 9 and 11) windows. For each bin and each randomization, the median speed was computed, generating 1000 medians of randomized speeds per bin. The overall median and the 1st and 99th percentiles were extracted from this data for every bin and plotted as the median speed and 98% confidence interval of the median to build the shuffled speed map. In Fig. 5 and Supplementary Figs. 8 and 11, genomic feature coordinates are from the UCSC SGD_other track and replication origins (ORIs) are from ref. [21]. Oriented genes in Supplementary Fig. 9 are from the Saccharomyces_cerevisiae.R64-1-1.104.gtf file from Ensembl (see "Computational resources" below for further details).

Replication fork directionality (RFD) computation. RFD is calculated for a given position as the difference between the proportions of rightward- (R) and leftward- (L) moving forks (RFD = R − L). RFD profiles from MCM869 FORK-seq and OK-seq data were computed as in ref. [13]. BT1 RFD profile from NFS data (i.e., $X0$ and $X1$ coordinates of oriented forks) was generated in a similar way with the simpleRFD function of the script helper_function.r (this function produces 4 bigwig files corresponding to the RFD and the total, leftward and rightward forks coverage) using forks from all BT1 biological replicates grown in standard conditions

at 30 °C. RFD data were binned into non-overlapping 100 nucleotide windows to reduce the size of the plot files. The correlation table in Supplementary Fig. 3b reporting Spearman's pairwise correlation coefficients was produced using the cor.rfd function of the helper_function.r script and the ggcorrplot function of the ggcorrplot R package. The cor.rfd function rests on the base R cor function but works on a coverage type of data (RleList) and excludes positions where N/A are present in at least one of the RFD profiles for which the correlation is computed.

Analysis of initiation and termination events. Initiation (ini) and termination (ter) event positions, defined as the midpoints between diverging and converging forks, respectively, were extracted from all BT1 biological replicates grown in standard conditions at 30 °C and processed as in ref. [13]. Replication origins (ORIs) are from ref. [21]. Distance of each ini or ter to the centre of the nearest ORI was computed using the distanceToNearest function of the GenomicRanges R package. Empirical cumulative distribution functions were plotted using the Ecdf function of the Hmisc R package. Shuffled versions of ini and ter positions were produced as a control using the custom shuffleGRen function of the helper_function.r script, which randomizes GenomicRanges positions while conserving the number of events *per* chromosome.

Simulation of BrdU incorporation during DNA replication. We simulated reads containing a BrdU signal mimicking an elongating fork as well as reads bearing multiple forks. Simulations were performed using Python scripts (Python version 3.6).

Simulation of BrdU level. We modelled intracellular BrdU level $b(t)$ as a function of time t as (i) zero before the BrdU pulse; (ii) an ascending section modelled by an exponential increase $As(t) = M(1 - \exp(-\frac{t}{\tau_1}))$ where t is the time since the start of the pulse, M the saturating concentration of BrdU and τ_1 the characteristic time of BrdU intake; (iii) a decreasing section starting at the start of the chase at time t_{pulse} (also corresponding to the duration of the pulse) that was modelled by a similar exponential decrease from the value attained at $t = t_{pulse}$:

$$Ds(t) = As(t_{pulse}) + (m - As(t_{pulse}))(1 - \exp(-\frac{t - t_{pulse}}{\tau_2})), \text{ where } m \text{ is the asymptotical}$$

BrdU concentration and τ_2 the characteristic time of BrdU outtake.

Simulation of a single replication fork. Given a fork speed v and a start position used as $x = 0$, BrdU time pulse shape was converted into a spatial BrdU incorporation pattern $BrdU(x)$ using the simple relation: $BrdU(x) = b(x/v)$. 10,000 reads containing a single fork were simulated with or without noise (see below) for the analyses presented in Fig. 3a and Supplementary Fig. 7.

Simulation of multiple replication forks. To simulate multiple forks, we created a DNA segment of 300 kb in length. We then randomly positioned 6 origins on it to have on average an origin every 50 kb. The firing time of each origin was randomly chosen between 0 and 30 min. A fork speed randomly drawn from a given distribution (see below) was assigned to the whole segment, which was simulated only if firing times, origin positions and fork speed created a set of 12 forks. Next, the moment of the pulse was randomly chosen between the firing time of the earliest origin and the three-fifths of the firing time of the latest origin (this time interval was chosen because it allowed to obtain replication signals in most simulated reads). Finally, to simulate molecules of similar length as experimental reads, the DNA segment was sliced into fragments according to a truncated log-normal distribution (truncations at 5 and 300 kb) with a shape parameter of 0.5 and a scale of 350. 100,000 reads containing one or several forks were simulated with or without noise (see below) for the analyses in Fig. 3a and Supplementary Fig. 7.

Choice of the simulation parameters. A replication fork was modelled by two shifted exponentials, each having two parameters (M, τ_1 and m, τ_2 for the ascending and descending parts, respectively; see above). To simulate replication signals resembling those found on real sequencing reads, we used parameters coming from experimental data. To do so, we selected among the >125,000 individual forks detected by NFS in BT1 cells grown at 30 °C those for which the chase was long enough (>2.5 kb), fitted the ≈90,000 forks meeting this criterion using the model described in the 'Simulation of BrdU level' section and the speed measured by NFS, and extracted fork parameters. In order to retain potential correlations between parameters, all four parameters of a randomly chosen experimental fork were assigned to a simulated fork. Fork speed v was randomly chosen from the deconvolved true fork speed distribution (see below) except when indicated otherwise.

Adding simulated experimental noise to the signal. To take into account signal variability between reads, we added to the $BrdU(x)$ signal a read-dependent offset value (O) drawn from a log-normal distribution of shape 1.98, location −4.09e−06 and scale 0.001 fitting the BrdU signal distribution of nanopore reads of DNA with no BrdU labelling. O was drawn again if its value was >0.2 or <1e−7. $BrdU(x)+O$ signals reaching values >1 were truncated to 1. We then randomly assigned "Nan" (Not a number) values to ≈77% of the signal so that only thymidine (T) positions had information about the BrdU content. Finally, to mimic Megalodon's output

signal, which is a probability of being a BrdU at each T site peaking at either 0 or 1, we assigned to each T position a value B drawn from a binomial law B(n,p) with parameters $p = BrdU(x)+O$ and n randomly chosen in [1,2,3]; B was subsequently divided by n to normalize between 0 and 1. Visual inspection of simulated versus experimental reads confirmed that they were virtually indistinguishable.

Mean BrdU traces and signal noise analysis for experimental and simulated forks. The experimental and simulated mean BrdU traces of oriented replication signals in Supplementary Fig. 6f, g were computed from ≈90,000 forks selected among the >125,000 detected by NFS in BT1 cells grown at 30 °C for which the chase was >2.5 kb (please note that leftward forks were reoriented in the rightward direction) and from 10,000 simulated reads containing a single fork, respectively, with the BrdU signal being smoothed using a 100 bp running average to have information even at non-thymidine sites. In order to characterize signal fluctuations (i.e., signal noise) in experimental and simulated data, the autocorrelation of the mean BrdU signal smoothed at 100 bp minus the signal smoothed at 1000 bp was computed. Subtracting the signal smoothed at 1000 bp removed slow BrdU variations from each track to solely keep local fluctuations that define signal noise.

Analysis of simulated data. The smoothing process of simulated reads during the parsing procedure and fork detection with NFS was performed with the same settings as for experimental reads. Speed error was computed by subtracting, for each simulated replication fork, its true speed to the speed estimated by NFS. For simulated reads with multiple replication forks, all forks within a given read had the same speed. To estimate NFS error for different speed categories, true speeds from simulated reads were grouped into bins of 100 bp/min.

Estimation of the true fork speed distribution in yeast from experimental measurements. A measured fork speed is an approximation of the true fork speed as (i) BrdU incorporation and detection by nanopore sequencing are subject to noise, and (ii) the measurement by NFS is sensitive to the noise and amplitude of BrdU signals in a complex manner. We used simulated reads to estimate the "transfer function" between the true and the observed fork speed distributions allowing us to deconvolve the measurement errors from the observed fork speed distribution and in turn to estimate the true speed distribution underlying the observed speed distribution. We first built a library of 100,000 reads with known true speeds following a uniform distribution between 50 and 5000 bp/min. We processed each noisy BrdU incorporation profile with NFS and obtained, for each true speed, the response fork speed distribution of the experimental procedure. We then determined the weight of each true speed category required to recover the experimentally observed distribution of >125,000 fork velocities at 30 °C. The weights corresponded to the deconvolved true fork speed distribution. The computational details of this procedure are detailed in the following paragraph.

We considered 44 discretised true fork speed values $v_k = k*100$ bp/min for $1 \leq k \leq 44$. The 100,000 simulated reads following the uniform speed distribution were then affected to 44 classes if their true speed fell inside a window centred on v_k and of width 400 bp/min, i.e., reads belonged to more than one class. This strategy was adopted in order to have enough reads in each class. Then, for each class k, we collected fork speeds estimated by NFS processing, leading to NFS fork speed distribution in response to true speed v_k that was subsequently fitted with a five-component Gaussian mixture. We then determined the weight w_k so that the weighted sum of those 44 distributions best fitted the experimental fork speed distribution. The weights w_k are an estimate of the true fork speed distribution within the 44 classes. This procedure was repeated independently 5 times and the average of the 5 weight sets were used as the deconvolved true fork speed distribution. Finally, this discretised distribution was fitted using a Gaussian mixture with six components to create a continuous true fork speed distribution. To evaluate the deconvolution procedure, we generated a new set of 100,000 reads with multiple forks following the estimated true fork speed distribution and processed them with NFS. The obtained observed fork speed distribution closely resembled the experimental distribution of >125,000 fork velocities, validating our approach. Deconvolution was performed with custom Python scripts using a library from ref. [61].

The mode of the estimated true speed distribution in Fig. 3b was determined using the mlv1 function of the modeest R package. The major peak of the distribution was fitted with a normal distribution with a mean equal to this mode. Standard deviation and weight of the normal distribution were then adjusted to best fit the peak.

Fork speed versus replication timing. Replication timing (RT) data are from ref. [42]. The liftOver tool (http://hgdownload.soe.ucsc.edu/goldenPath/sacCer1/liftOver/) was used to convert genomic positions from sacCer1 to the sacCer3 version of the yeast genome. Data were normalized between 0 and 1 (start and end of S phase, respectively). The mean RT was computed for each fork, then fork speed versus RT was plotted as a 2D density plot with hexagonal bins using the geom_hex() function of the ggplot2 R package with default parameters. Spearman's correlation between fork speed and RT was computed using the stat_cor() function of the ggpubr R package.

Statistical analysis. The R environment v4.0.5 was used for all the analyses. Prior to analysis, fork speed values were averaged for every sample according to the classes of the studied factors, in order to (1) decrease the sensitivity of the tests, which otherwise detect fork speed differences of no practical significance when extremely large number of values are compared and (2) use as a unique source of variation the inter-experiment variations, which correspond to error variations, while fork speed variations have a biological origin. Data were fitted to a linear model that included fork speed as response variable, the factor of interest as predictive variable and the biological replicate/sequencing run as blocking factor. Two by two effect comparisons (two-sided contrast comparisons) were performed with the emmeans() function of the emmeans R package. When only two values were present in one group (Fig. 4a, b and Supplementary Fig. 12), a limma linear fitting and contrast analysis was performed with the limma R package. Statistical significance was set to $p \leq 0.05$. In each case, type I error was controlled by correcting the p-values according to the Benjamini & Hochberg method ("BH" option in the p.adjust() function of R). Speeds of forks overlapping two different features (e.g., a tRNA and a centromere) or two tRNAs were removed from the analyses presented in Fig. 6a, b and Supplementary Fig. 12 to respect the exclusive factor level requirement of statistical analysis. For the examination of speed at individual tRNAs (Supplementary Fig. 12), means above 5000 bp/min or showing a leverage of 1 in the fitted model were removed from the analysis as they strongly influenced the fitting. This concerned 11 values among the 2984 tRNA x sequencing run levels analysed. All samples corresponded to independent cell cultures except for the two BT1 PromethION runs, which were technical replicates (two sequencing libraries from the genomic DNA of one BT1 cell culture analysed on different PromethION flow cells); the "library plus flow cell" effect was deemed sufficient to consider these as distinct samples. Statistical analysis results are detailed in Supplementary Table 2.

Visualisation tools. Plots were made using custom R scripts. For half-eye plots, we used the stat_slab and stat_pointinterval functions of the ggdist R package. stat_slab outputs a density distribution of the experimental points; stat_pointinterval outputs the median and the 50% and 95% intervals of the data centred on the median.

Reference genome. The sacCer3 genome release (S288C_reference_sequence_R64-2-1) was used as the reference genome. The rDNA locus is composed of two ribosomal DNA units on chromosome XII (chrXII: 451,575–468,931) in this version. As rDNA reads containing more than two repeats tended to incorrectly map and to create artefactual BrdU signals, we added an artificial chromosome with 10 tandem rDNA repeats to the reference genome. The external rDNA repeats were flanked by 10 kb sequences located upstream and downstream of the original rDNA locus to allow the mapping of reads overlapping the first or last rDNA repeat of the rDNA array. In order to build this extra "chromosome", 8 copies of the second rDNA unit (chrXII: 459,793–468,929 fragment) were inserted between the first and second unit of the original S288C genome, resulting in a 113-kb-long "chromosome" that we named rDNA-10R. Because the proper alignment of reads on tandem repeats the unit of which is of similar length to that of the reads requires specialised procedures, reads mapping to rDNA-10R, which accounted for the most part of rDNA reads, were subsequently excluded from our analyses.

Computational resources. Genomic feature coordinates (centromeres, telomeres, HML/HMR loci, rDNA and tRNA genes) were extracted from the UCSC SGD_other track (https://genome.ucsc.edu/cgi-bin/hgTables) and saved as bed files (script Script_YeastAnnotation.r). Please note that "telomere" annotations in the SGD_other track encompass both the terminal stretch of telomeric repeats (when present in the sacCer3 genome) and subtelomeric sequences; moreover, since our smoothing procedure removes the first and last 2.5 kb of each read, telomeric repeats were excluded from our analyses. The UCSC SGD_other track was also used to determine the direction of transcription of tRNA genes in Fig. 6b and in Supplementary Fig. 12. In this figure, tRNA genes were subsequently categorized as co-directional (CD) or head-on (HO) with respect to replication if their transcription was in the same or opposite orientation to the main direction of fork progression, respectively, which was determined for each tRNA according to its mean RFD value (RFD > 0, forks travel mostly rightward; RFD < 0, forks travel mostly leftward). In Fig. 6a, b and Supplementary Figs. 10 and 12, forks were classified according to their overlap with either a group of features (centromeres, telomeres and tRNA genes) or individual features (rDNA locus and individual centromeres, telomeres and CD/HO-sorted tRNA genes) after fork conversion into GenomicRanges. The overlap was tested with the OverlapsAny function of the GenomicRanges R package (minimal overlap was one nucleotide). Yeast replication origins in Fig. 5 and Supplementary Figs. 3a, c, 8 and 11 are from ref. [21]. RT data in Fig. 6d are from ref. [42] (data accessible from NCBI's Gene Expression Omnibus repository, accession code GSM1036187, GSM1036187_T7107_normalised.wig.gz file). Oriented genes in Supplementary Fig. 9 are from the Saccharomyces_cerevisiae.R64-1-1.104.gtf file downloaded from Ensembl (http://ftp.ensembl.org/pub/release-104/gtf/saccharomyces_cerevisiae/). DNAscent v1 and v2 were downloaded from https://github.com/MBoemo/DNAscent.

R packages. R packages used in this study are kmlShape version 0.9.5 (https://CRAN.R-project.org/package=kmlShape); DescTools version 0.99.44 (https://CRAN.R-project.org/web/packages/DescTools); RcppRoll version 0.3.0 (https://CRAN.R-project.org/package=RcppRoll); Hmisc version 4.6-0 (https://CRAN.R-project.org/package=Hmisc); tidyverse[62]; GenomicRanges[63]; rtracklayer[64]; BSgenome version 1.56.0 (https://bioconductor.org/packages/BSgenome); ggdist version 3.0.1 (https://mjskay.github.io/ggdist/); patchwork version 1.1.1 (https://CRAN.R-project.org/package=patchwork); ggplot2 version 3.3.5 (https://CRAN.R-project.org/package=ggplot2); ggcorrplot version 0.1.3 (https://CRAN.R-project.org/package=ggcorrplot); ggpubr version 0.4.0 (https://CRAN.R-project.org/package=ggpubr); gridExtra version 2.3 (https://CRAN.R-project.org/package=gridExtra); modeest version 2.4.0 (https://CRAN.R-project.org/package=modeest); ggprism version 1.0.3 (https://CRAN.R-project.org/package=ggprism); ggrepel version 0.9.1 (https://CRAN.R-project.org/package=ggrepel); furrr version 0.2.3 (https://CRAN.R-project.org/package=furrr); devtools version 2.4.2 (https://CRAN.R-project.org/package=devtools); emmeans version 1.5.5-1 (https://CRAN.R-project.org/package=emmeans); and limma version 3.46.0 (https://bioconductor.org/packages/release/bioc/html/limma.html).

Genomic coordinates. Coordinates are given according to the sacCer3 yeast genome assembly.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability
Nanopore sequencing data generated in this study have been deposited in the ENA database under accession code PRJEB50302. Source data are available at https://github.com/LacroixLaurent/NanoForkSpeed and https://doi.org/10.5281/zenodo.5958270. Yeast genomic feature coordinates used in this study originate from UCSC SGD_other track (https://genome.ucsc.edu/cgi-bin/hgTables), oriented genes are from Ensembl database (Saccharomyces_cerevisiae.R64-1-1.104.gtf file available at http://ftp.ensembl.org/pub/release-104/gtf/saccharomyces_cerevisiae/), replication origins are from ref. [21] and replication timing data are from ref. [42] (data accessible at NCBI GEO database, accession code GSM1036187). Source data are provided with this paper.

Code availability
Megalodon-based BrdU basecaller, NFS software and associated R scripts can be accessed at https://github.com/LacroixLaurent/NanoForkSpeed and https://doi.org/10.5281/zenodo.6535390. Custom R scripts for statistical analyses can be accessed at https://gitlab.pasteur.fr/gmillot/anova_contrasts/-/tree/v7.2.0. Python scripts are available at https://github.com/organic-chemistry/simunano.

References
1. Gaillard, H., García-Muse, T. & Aguilera, A. Replication stress and cancer. Nat. Rev. Cancer 15, 276–289 (2015).
2. Kapp, L. N. & Painter, R. B. in International Review of Cytology (eds Bourne, G. H., Danielli, J. F. & Jeon, K. W.) (Academic Press, 1982).
3. Liapunova, N. A. in International Review of Cytology (eds Jeon, K. W. & Jarvik, J.) (Academic Press, 1994).
4. Técher, H. et al. Replication dynamics: biases and robustness of DNA fiber analysis. J. Mol. Biol. 425, 4845–4855 (2013).
5. Yurov, Y. B. Rate of DNA replication fork movement within a single mammalian cell. J. Mol. Biol. 136, 339–342 (1980).
6. Anglana, M., Apiou, F., Bensimon, A. & Debatisse, M. Dynamics of DNA replication in mammalian somatic cells: nucleotide pool modulates origin choice and interorigin spacing. Cell 114, 385–394 (2003).
7. Letessier, A. et al. Cell-type-specific replication initiation programs set fragility of the FRA3B fragile site. Nature 470, 120–123 (2011).
8. Guilbaud, G. et al. Evidence for sequential and increasing activation of replication origins along replication timing gradients in the human genome. PLOS Comput. Biol. 7, e1002322 (2011).
9. Blin, M. et al. Transcription-dependent regulation of replication dynamics modulates genome stability. Nat. Struct. Mol. Biol. 26, 58–66 (2019).
10. Mendez-Bermudez, A. et al. Genome-wide control of heterochromatin replication by the telomere capping protein TRF2. Mol. Cell 70, 449–461.e5 (2018).
11. Raghuraman, M. K. et al. Replication dynamics of the yeast genome. Science 294, 115–121 (2001).

12. Sekedat, M. D. et al. GINS motion reveals replication fork progression is remarkably uniform throughout the yeast genome. *Mol. Syst. Biol.* **6**, 353 (2010).

13. Hennion, M. et al. FORK-seq: replication landscape of the *Saccharomyces cerevisiae* genome by nanopore sequencing. *Genome Biol.* **21**, 125 (2020).

14. Ma, E., Hyrien, O. & Goldar, A. Do replication forks control late origin firing in *Saccharomyces cerevisiae*? *Nucleic Acids Res.* **40**, 2010–2019 (2012).

15. Müller, C. A. et al. Capturing the dynamics of genome replication on individual ultra-long nanopore sequence reads. *Nat. Methods* **16**, 429–436 (2019).

16. Boemo, M. A. DNAscent v2: detecting replication forks in nanopore sequencing data with deep learning. *BMC Genomics* **22**, 430 (2021).

17. Viggiani, C. J. & Aparicio, O. M. New vectors for simplified construction of BrdU-Incorporating strains of *Saccharomyces cerevisiae*. *Yeast* **23**, 1045–1051 (2006).

18. Bianco, J. N. et al. Analysis of DNA replication profiles in budding yeast and mammalian cells using DNA combing. *Methods* **57**, 149–157 (2012).

19. Magiera, M. M., Gueydon, E. & Schwob, E. DNA replication and spindle checkpoints cooperate during S phase to delay mitosis and preserve genome integrity. *J. Cell Biol.* **204**, 165–175 (2014).

20. Vernis, L., Piskur, J. & Diffley, J. F. X. Reconstitution of an efficient thymidine salvage pathway in *Saccharomyces cerevisiae*. *Nucleic Acids Res.* **31**, e120 (2003).

21. Newman, T. J., Mamun, M. A., Nieduszynski, C. A. & Blow, J. J. Replisome stall events have shaped the distribution of replication origins in the genomes of yeasts. *Nucleic Acids Res.* **41**, 9705–9718 (2013).

22. Lengronne, A., Pasero, P., Bensimon, A. & Schwob, E. Monitoring S phase progression globally and locally using BrdU incorporation in TK+ yeast strains. *Nucleic Acids Res.* **29**, 1433–1442 (2001).

23. Ohya, T. et al. The DNA polymerase domain of polε is required for rapid, efficient, and highly accurate chromosomal DNA replication, telomere length maintenance, and normal cell senescence in *Saccharomyces cerevisiae*. *J. Biol. Chem.* **277**, 28099–28108 (2002).

24. Yang, S. C. H., Rhind, N. & Bechhoefer, J. Modeling genome-wide replication kinetics reveals a mechanism for regulation of replication timing. *Mol. Syst. Biol.* **6**, 404 (2010).

25. Gispan, A., Carmi, M. & Barkai, N. Model-based analysis of DNA replication profiles: predicting replication fork velocity and initiation rate by profiling free-cycling cells. *Genome Res.* **27**, 310–319 (2017).

26. Frenkel, N., Jonas, F., Carmi, M., Yaakov, G. & Barkai, N. Rtt109 slows replication speed by histone N-terminal acetylation. *Genome Res.* **31**, 426–435 (2021).

27. Szyjka, S. J., Viggiani, C. J. & Aparicio, O. M. Mrc1 is required for normal progression of replication forks throughout chromatin in *S. cerevisiae*. *Mol. Cell* **19**, 691–697 (2005).

28. Tourrière, H., Versini, G., Cordón-Preciado, V., Alabert, C. & Pasero, P. Mrc1 and Tof1 promote replication fork progression and recovery independently of Rad53. *Mol. Cell* **19**, 699–706 (2005).

29. Hodgson, B., Calzada, A. & Labib, K. Mrc1 and Tof1 regulate DNA replication forks in different ways during normal S phase. *Mol. Biol. Cell* **18**, 3894–3902 (2007).

30. Yeeles, J. T. P., Janska, A., Early, A. & Diffley, J. F. X. How the eukaryotic replisome achieves rapid and efficient DNA replication. *Mol. Cell* **65**, 105–116 (2017).

31. Eickhoff, P. et al. Molecular basis for ATP-hydrolysis-driven DNA translocation by the CMG helicase of the eukaryotic replisome. *Cell Rep.* **28**, 2673–2688.e8 (2019).

32. Baretić, D. et al. Cryo-EM structure of the fork protection complex bound to CMG at a replication fork. *Mol. Cell* **78**, 926–940.e13 (2020).

33. Lewis, J. S. et al. Single-molecule visualization of *Saccharomyces cerevisiae* leading-strand synthesis reveals dynamic interaction between MTC and the replisome. *Proc. Natl Acad. Sci. USA* **114**, 10630 (2017).

34. Poli, J. et al. dNTP pools determine fork progression and origin usage under replication stress. *EMBO J.* **31**, 883–894 (2012).

35. Mirkin, E. V. & Mirkin, S. M. Replication fork stalling at natural impediments. *Microbiol. Mol. Biol. Rev.* **71**, 13–35 (2007).

36. Gadaleta, M. C. & Noguchi, E. Regulation of DNA replication through natural impediments in the eukaryotic genome. *Genes* **8**, 98 (2017).

37. Osmundson, J. S., Kumar, J., Yeung, R. & Smith, D. J. Pif1-family helicases cooperatively suppress widespread replication-fork arrest at tRNA genes. *Nat. Struct. Mol. Biol.* **24**, 162–170 (2017).

38. Yeung, R. & Smith, D. J. Determinants of replication-fork pausing at tRNA genes in *Saccharomyces cerevisiae*. *Genetics* **214**, 825–838 (2020).

39. Rivin, C. J. & Fangman, W. L. Replication fork rate and origin activation during the S phase of *Saccharomyces cerevisiae*. *J. Cell Biol.* **85**, 108–115 (1980).

40. Malinsky, J. et al. The supply of exogenous deoxyribonucleotides accelerates the speed of the replication fork in early S-phase. *J. Cell Sci.* **114**, 747–750 (2001).

41. Takebayashi, S. I. et al. Regulation of replication at the R/G chromosomal band boundary and pericentromeric heterochromatin of mammalian cells. *Exp. Cell Res.* **304**, 162–174 (2005).

42. Müller, C. A. et al. The dynamics of genome replication using deep sequencing. *Nucleic Acids Res.* **42**, e3 (2014).

43. Claussin, C., Vazquez, J. & Whitehouse, I. Single-molecule mapping of replisome progression. *Mol. Cell* **82**, 1372–1382 (2022).

44. de Moura, A. P. S., Retkute, R., Hawkins, M. & Nieduszynski, C. A. Mathematical modelling of whole chromosome replication. *Nucleic Acids Res.* **38**, 5623–5633 (2010).

45. Baker, A. et al. Replication fork polarity gradients revealed by megabase-sized U-shaped replication timing domains in human cell lines. *PLOS Comput. Biol.* **8**, e1002443 (2012).

46. Norio, P. & Schildkraut, C. L. Visualization of DNA replication on individual Epstein-Barr virus episomes. *Science* **294**, 2361–2364 (2001).

47. Yabuki, N., Terashima, H. & Kitada, K. Mapping of early firing origins on a replication profile of budding yeast. *Genes Cells* **7**, 781–789 (2002).

48. Audit, B. et al. Multiscale analysis of genome-wide replication timing profiles using a wavelet-based signal-processing algorithm. *Nat. Protoc.* **8**, 98–110 (2013).

49. Zhao, P. A., Sasaki, T. & Gilbert, D. M. High-resolution Repli-Seq defines the temporal choreography of initiation, elongation and termination of replication in mammalian cells. *Genome Biol.* **21**, 76 (2020).

50. Ziane, R., Camasses, A. & Radman-Livaja, M. The asymmetric distribution of RNA polymerase II and nucleosomes on replicated daughter genomes is caused by differences in replication timing between the lagging and the leading strand. *Genome Res.* **32**, 337–356 (2022).

51. Reynolds, A. E., McCarroll, R. M., Newlon, C. S. & Fangman, W. L. Time of replication of ARS elements along yeast chromosome III. *Mol. Cell. Biol.* **9**, 4488–4494 (1989).

52. Peace, J. M., Villwock, S. K., Zeytounian, J. L., Gan, Y. & Aparicio, O. M. Quantitative BrdU immunoprecipitation method demonstrates that Fkh1 and Fkh2 are rate-limiting activators of replication origins that reprogram replication timing in G1 phase. *Genome Res.* **26**, 365–375 (2016).

53. Macheret, M. & Halazonetis, T. D. Intragenic origins due to short G1 phases underlie oncogene-induced DNA replication stress. *Nature* **555**, 112–116 (2018).

54. Sherman, F. In *Methods in Enzymology* (eds Guthrie, C. & Fink, G. R.) (Academic Press, 2002).

55. Grote, A. et al. JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. *Nucleic Acids Res.* **33**, W526–W531 (2005).

56. Talarek, N., Petit, J., Gueydon, E. & Schwob, E. In *DNA Replication: Methods and Protocols* (eds Vengrova, S. & Dalgaard, J.) (Springer New York, 2015).

57. Hennion, M., Theulot, B., Arbona, J. M., Audit, B. & Hyrien, O. In *Yeast Functional Genomics* (ed. Devaux, F.) (Humana Press, 2022).

58. R Core Team. R: a language and environment for statistical computing (R Foundation for Statistical Computing, 2020).

59. Ramer, U. An iterative procedure for the polygonal approximation of plane curves. *Computer Graph. Image Process.* **1**, 244–256 (1972).

60. Douglas, D. H. & Peucker, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: Int. J. Geographic Inf. Geovisualization* **10**, 112–122 (1973).

61. Schreiber, J. Pomegranate: fast and flexible probabilistic modeling in python. *J. Mach. Learn. Res.* **18**, 5992–5997 (2017).

62. Wickham, H. et al. Welcome to the Tidyverse. *J. Open Source Softw.* **4**, 1686 (2019).

63. Lawrence, M. et al. Software for computing and annotating genomic ranges. *PLOS Comput. Biol.* **9**, e1003118 (2013).

64. Lawrence, M., Gentleman, R. & Carey, V. rtracklayer: an R package for interfacing with genome browsers. *Bioinformatics* **25**, 1841–1842 (2009).

Acknowledgements

Author contributions

B.T., B.L.T., J.P. and F.P. performed the experiments. L.L. developed the NanoForkSpeed pipeline. J.M.A. implemented BrdU detection in ONT's Megalodon program, simulated replication forks and performed the deconvolution analysis. B.T., C.C., S.E. and A.L. performed the nanopore sequencing. L.L., B.L.T., B.T., J.M.A., B.A., E.J., O.H. and M.H. analysed the data. G.A.M., L.L. and B.A. performed the statistical analyses. B.L.T. and L.L. wrote the manuscript, with inputs from the other authors. B.L.T., L.L. and O.H. designed the project and supervised the study.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-022-31012-0.

Correspondence and requests for materials should be addressed to Laurent Lacroix, Olivier Hyrien or BenoîtLe Tallec.

Peer review information *Nature Communications* thanks Susan Gerbi and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at http://www.nature.com/reprints

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

Genome-wide mapping of individual replication fork velocities using nanopore sequencing

SUPPLEMENTARY INFORMATION
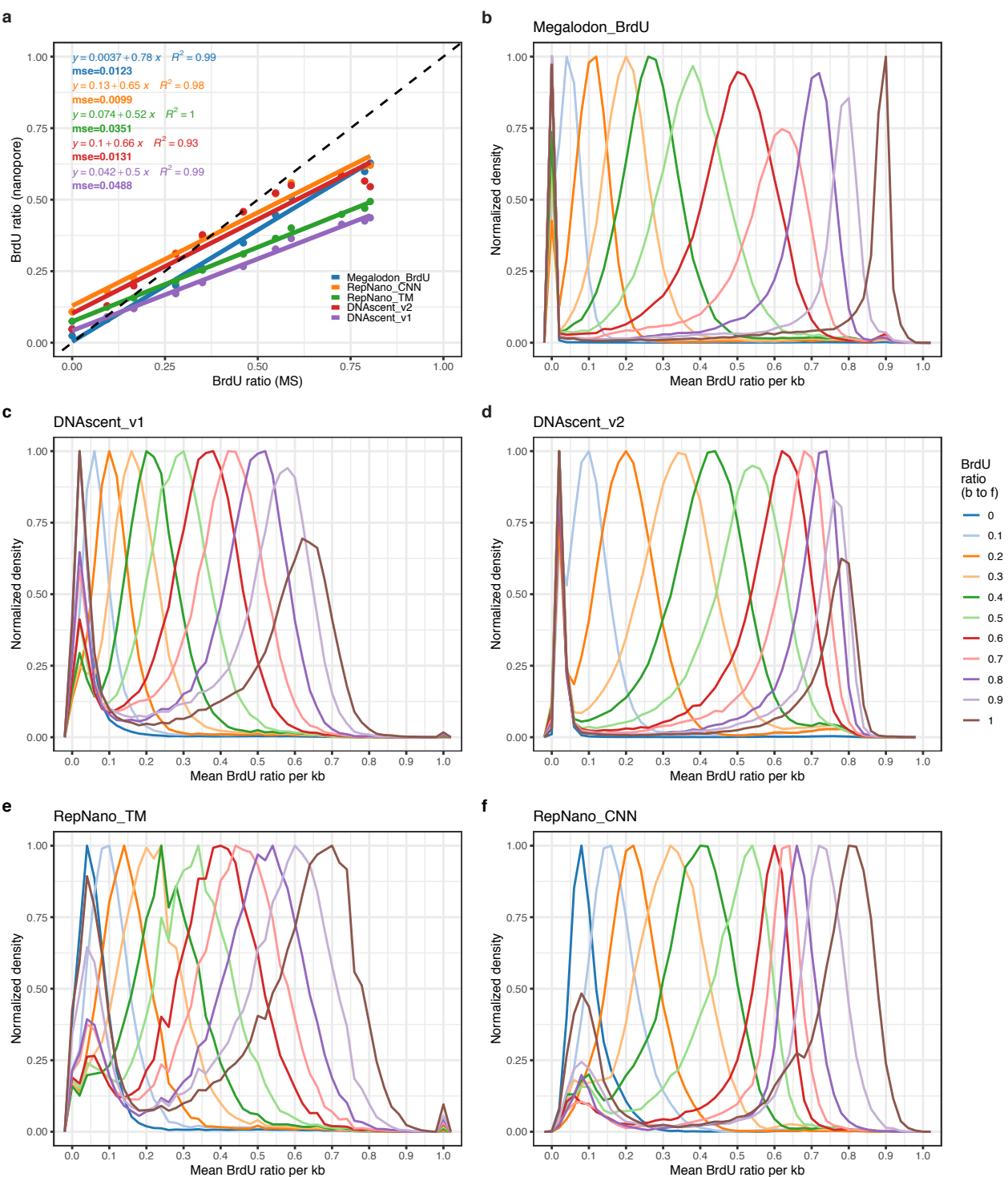
Supplementary Figure 1. Detection of BrdU in genomic DNA by Megalodon. a, Overall BrdU content estimated using Megalodon, DNAscent v1[15], DNAscent v2[16], RepNano[13] transition matrices (RepNano_TM) and RepNano convolutional neural network (RepNano_CNN) *versus* mass spectrometry measurements (MS) for 11 samples of genomic DNA from MCM869 cells grown with a percentage of 0 (thymidine control) to 100% BrdU in the culture medium with 10% increments. R^2, regression line R-squared value; mse, mean squared error relative to the perfect correlation line (y = x, dashed line). Regression line equations are indicated to compare slopes and intercepts; the equation for Megalodon is the closest to y = x. **b-f**, Normalized density of BrdU content averaged over 1 kb for nanopore reads of the 11 samples with different proportions of BrdU in the culture medium (0 to 1 with 0.1 increments) described above as determined using Megalodon (**b**), DNAscent (**c**), DNAscent v2 (**d**), RepNano transition matrices (**e**) and RepNano convolutional neural network (**f**) detection methods.

Supplementary Figure 2. Characterization of the BT1 yeast strain. a-c, BT1 cells exhibit WT growth and S phase progression. **a**, Representative growth curve of WT, BT1 and MCM869 cells with 100 μM thymidine. This experiment was performed independently three times for WT and BT1 cells, twice for the MCM869 strain, with similar results. **b**, Doubling time of WT, BT1 and MCM869 cells with 100 μM thymidine (black centre line and whiskers, median and extreme values, respectively; n values are independent cell cultures: $n_{WT} = 3$, $n_{BT1} = 3$ and $n_{MCM869} = 2$). **c**, Representative FACS analysis of S phase progression of BT1 cells. WT and BT1 cells were arrested in G1 with α-factor and released synchronously into S phase. DNA content was analysed by flow cytometry after DNA staining with SYTOX Green. This experiment was performed twice independently with similar results. Async., asynchronous. **d**, Examples of BrdU content profiles of nanopore reads with BrdU signal above background level mapping to the same genomic regions from pulse-labelled MCM869, BT1, BT2 (MCM869 strain with a functional *de novo* dTMP biosynthesis pathway) and BT3 (strain expressing hsvTK and hENT1 from five tandemly integrated copies of the p306-BrdU-Inc plasmid[17]) cells using Megalodon. A scheme below the name of each strain recapitulates the integrated transgenes allowing the incorporation of exogenous thymidine or thymidine analogues in the cognate strain, and specifies if endogenous thymidine synthesis was disrupted through inactivation of the thymidylate synthase-encoding *CDC21* gene. The number of integrated copies is indicated for each transgene; the name of the locus of insertion is given in parentheses. Diagrams are not drawn to scale. Pro, promoter; cod-opt, codon-optimised for expression in yeast. Light blue dots, raw data from Megalodon (dots represent the probability of BrdU at each thymidine position); blue curve, smoothed signal.

Supplementary Figure 3. RFD profiling and detection of initiation and termination events using NFS. a, RFD profiles in 100 bp adjacent windows of chromosome XV from NFS, FORK-seq and Okazaki fragment sequencing (OK-seq)[13]. Bottom, known *S. cerevisiae* replication origins (ORI) from[19]. **b**, Spearman's pairwise correlation coefficients between genome-wide RFD profiles obtained by the indicated methods. **a**, **b**, The strain name is given in parentheses. **c**, Empirical cumulative distribution function (ECDF) of the distances between diverging (NFS_ini, red) or converging (NFS_ter, blue) fork midpoints and the closest ORI centre[19]. Shuffled versions of NFS_ini (pink) and NFS_ter (light blue) positions were used as a control. ini, initiation event; ter, termination event; n, number of ini or ter; med, median distance between ini or ter and ORI centres.
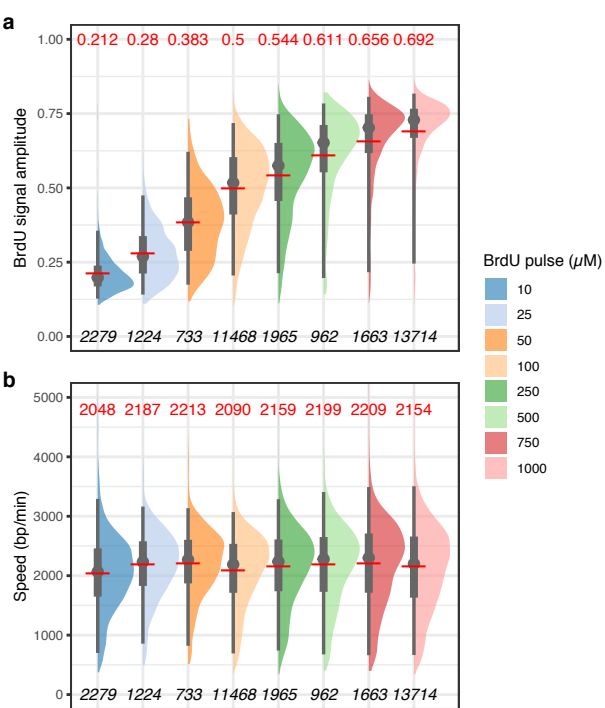
Supplementary Figure 4. Impact of BrdU concentration used for the pulse-labelling of replicating DNA on fork speed measurement by NFS. a, **b**, Half-eye plots of fork maximal BrdU signal amplitudes (**a**) and of fork velocities (**b**) determined by NFS on nanopore reads of genomic DNA from cells pulse-labelled with BrdU doses ranging from 10 µM to 1 mM. Red line, mean amplitude or speed, value indicated in red on top; grey dot, median; thick and thin grey vertical lines, 50 and 95% intervals, respectively; bottom, number of measurements.

Supplementary Figure 5. Doses of BrdU up to 100 μM do not impact S phase progression nor activate Rad53 in BT1 cells. a, Representative FACS analysis of S phase progression of BT1 cells in the presence of increasing doses of BrdU. BT1 cells were arrested in G1 with α-factor, released synchronously into S phase and added with BrdU after 15 min. This experiment was performed twice independently with similar results. Async., asynchronous. **b**, Representative western blot analysis with anti-Rad53 antibodies of extracts from exponentially growing BT1 cells treated with increasing doses of BrdU for 30 min. Retarded bands correspond to phosphorylated (activated) Rad53, which exhibit a slower mobility than unphosphorylated Rad53. Cells treated with 200 mM hydroxyurea (HU) were used as a positive control for Rad53 phosphorylation. This experiment was performed twice independently with similar results. Uncropped blot is available in Source Data.

Supplementary Figure 6. Comparison of experimental and simulated reads. a-e, Examples of simulated reads containing a single fork (**a**, **b**) or one or multiple forks (**c**, **d**) without (**a**, **c**) or with (**b**, **d**) noise. BrdU content profiles of experimental nanopore reads from pulse-labelled BT1 cells (smoothed data after Megalodon processing) are shown in (**e**) for comparison. Visual inspection shows that forks from authentic and simulated reads with noise are indistinguishable. Simulated and experimental reads shown here were randomly selected. **f**, Mean BrdU traces of oriented replication signals from experimental (purple curve) and simulated data (red curve). **g**, Autocorrelation of the mean BrdU signal of experimental and simulated forks. Autocorrelation allows the determination of the scale at which the noise at one position of the signal is correlated to the noise at a given distance. Autocorrelation of the mean BrdU signal dropped to 0 very fast (*i.e.*, within ≈ 100 bp) for both the experimental and simulated data (purple and red curve, respectively), indicating that the noise added to the simulated forks resembles the experimental one. See Methods for details.

Supplementary Figure 7. Distribution of measurement errors made by NFS according to fork speed. a-d, Fork velocities were determined by NFS on 10,000 simulated reads containing a single fork of known speed without (**a**) or with (**b**) noise and on 100,000 simulated reads harbouring one or multiple forks without (**c**) or with (**d**) noise, and speed errors (*i.e.*, the difference between NFS measurement and the actual speed of a given fork) were grouped according to fork speed sorted into 100 bp/min categories (0-100, 100-200, etc) and represented as boxplots. Thick horizontal black line, median speed error, value indicated in red on top, with the median relative speed error indicated below; box lower and upper hinges, 1st and 3rd quartiles, respectively; upper and lower whiskers, largest and smallest value no further than 1.5x interquartile range from the hinge, respectively; the number of values is indicated below each box. Horizontal dotted lines at speed errors of -250, 0 and 250 bp/min are here to guide the eyes. Forks were simulated on the basis of our estimate of the true fork speed distribution in yeast (see Methods).
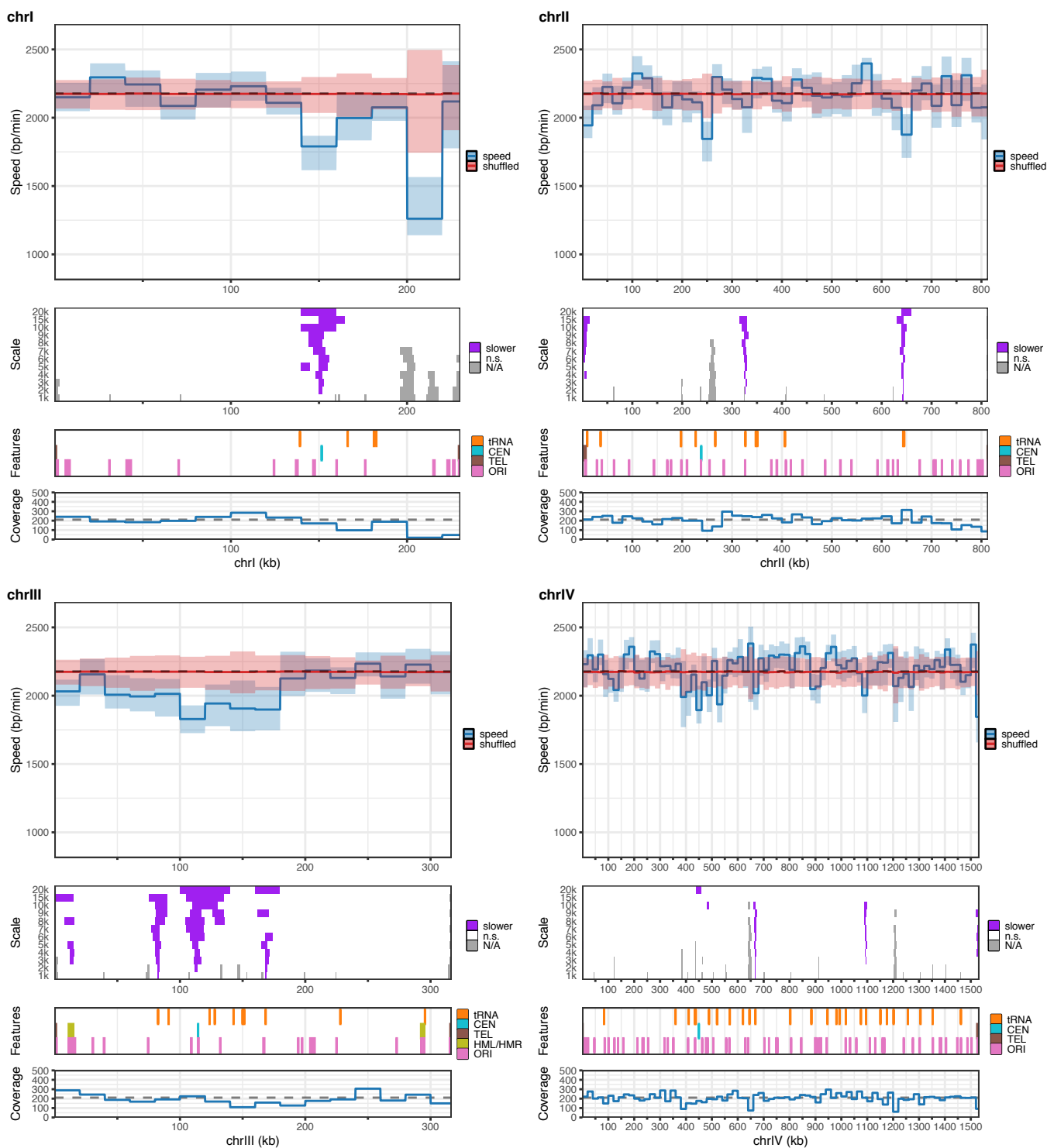
Supplementary Figure 8. Replication fork progression map of yeast chromosomes I to X, XV and XVI. See Fig. 5 caption for details.
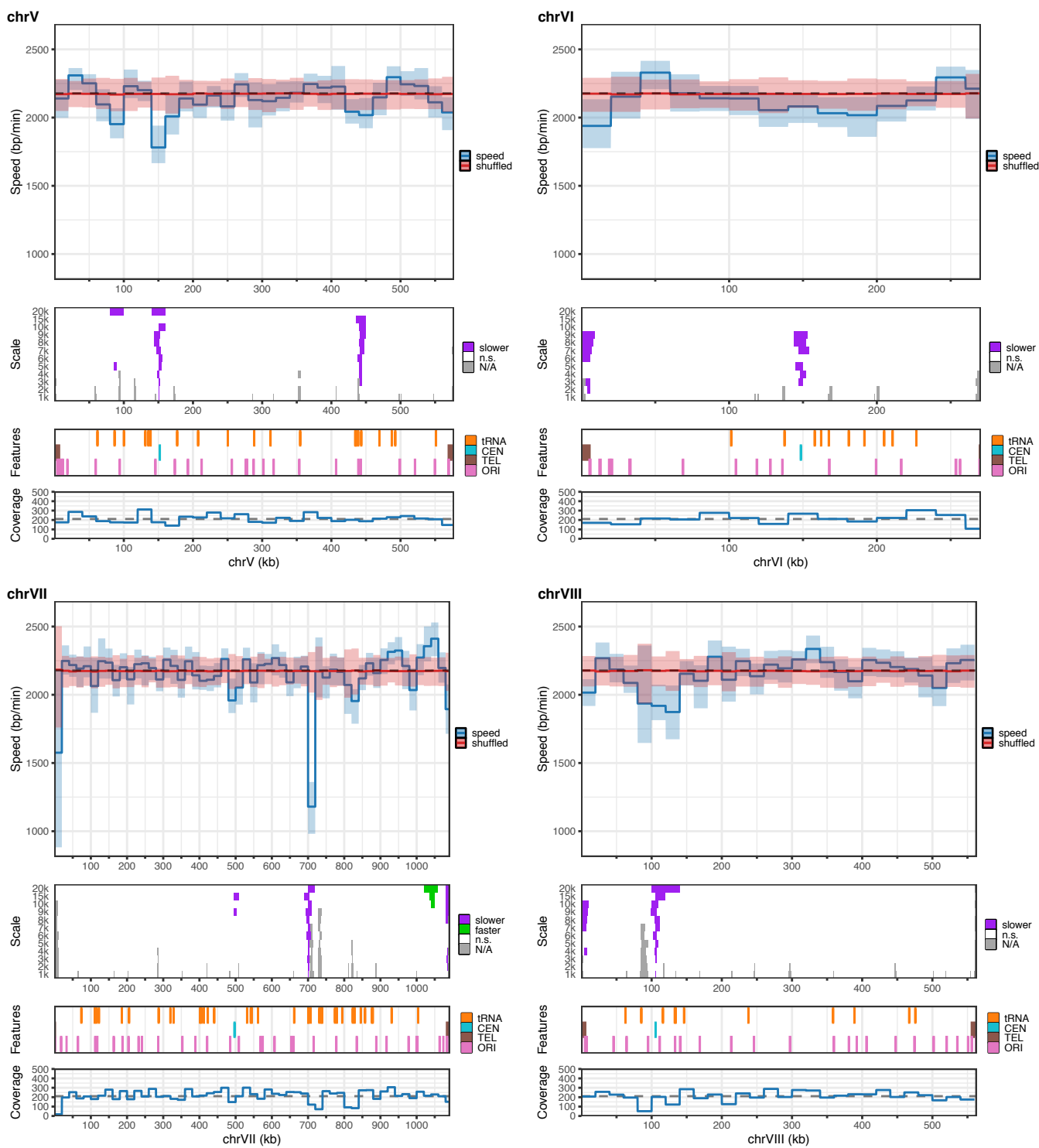
Supplementary Figure 8, continued.
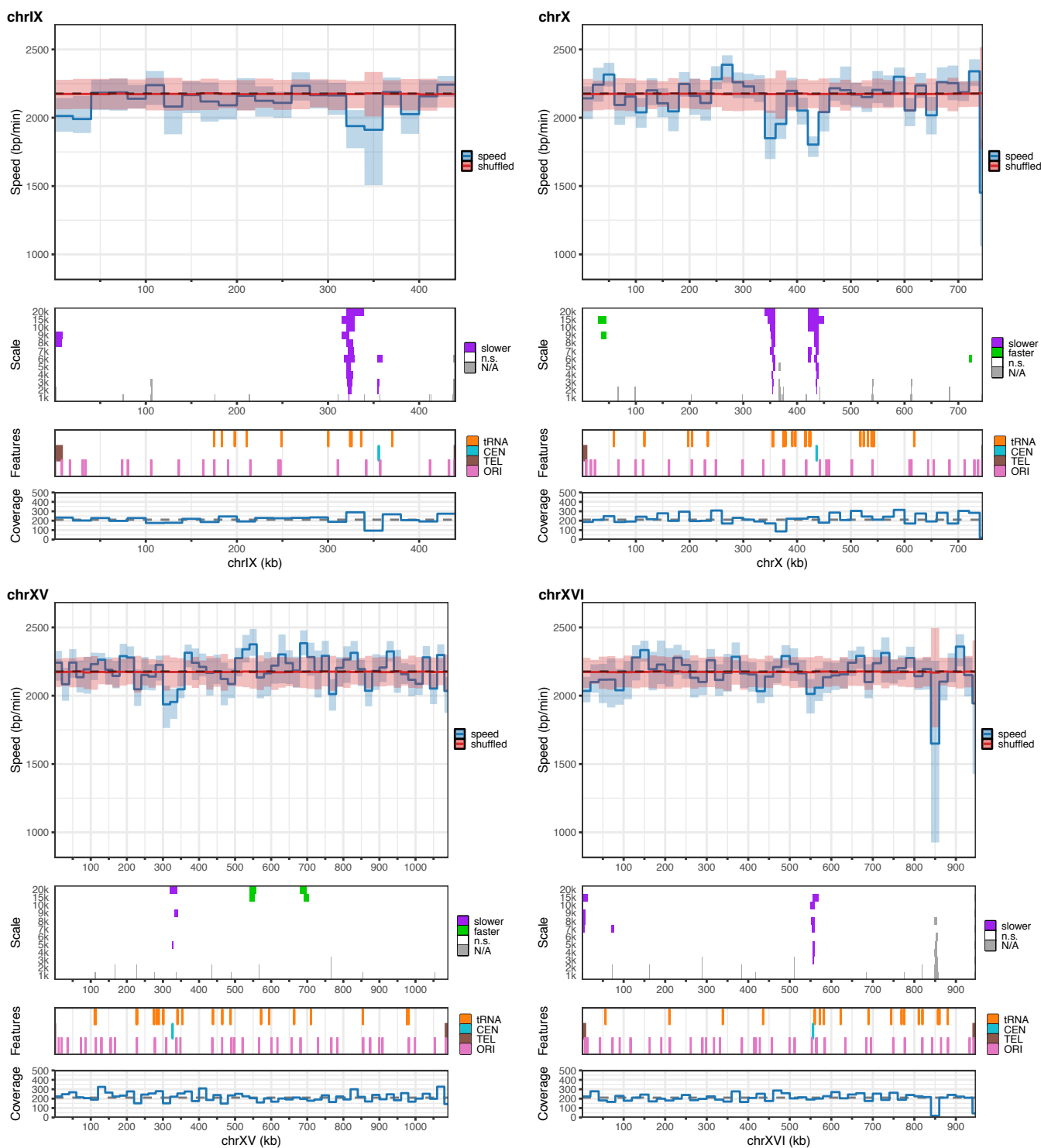
Supplementary Figure 8, continued.

Supplementary Figure 9. Replication fork progression map of a portion of chromosome XI. Shown is the region of chromosome XI where forks seemingly travel faster than in the rest of the genome. Panels from top to bottom: 1) median of experimental fork speeds (blue line) with 98% confidence interval of the median (light blue) and median of reshuffled speeds (red line) with 98% confidence interval of the median (light red) computed in 1 kb windows (dotted line, median fork speed in the whole genome); 2) results of Mann-Whitney-Wilcoxon tests with Holm correction (one-sided) performed along the chromosome to compare the speed distribution in a given window of a given width (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 and 20 kb) to the speed distribution on the whole genome (green, regions of higher fork speed; white, n.s., not significant; statistical significance was set to $p < 0.01$); 3) position of genes (geneR and geneL, rightward and leftward direction of transcription, respectively); 4) coverage of individual replication fork velocities (dotted line, median coverage of the genome). No tRNA nor replication origin (from[19]) are present in this region. Please note that it was not possible to compute a finite confidence interval for the median of the experimental fork speeds for windows with a coverage ≤ 5.
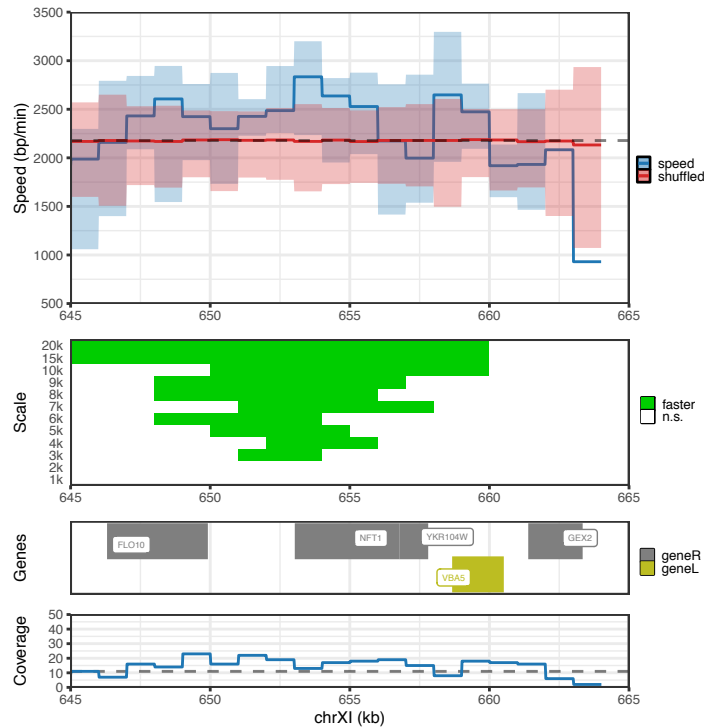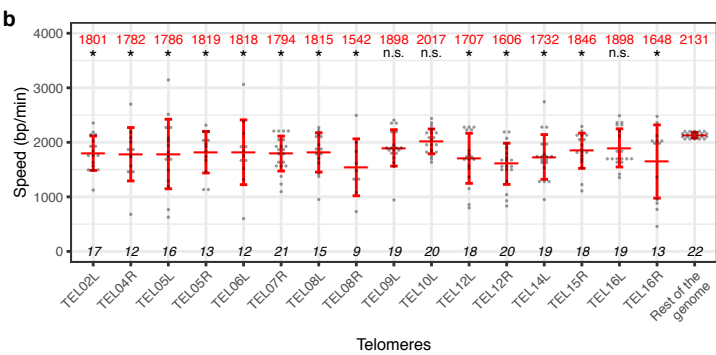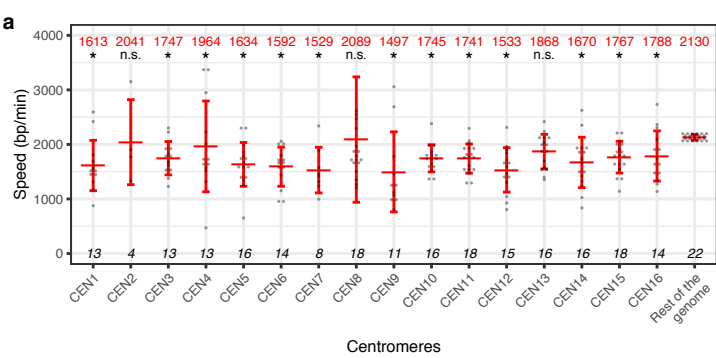
Supplementary Figure 10. Targeted analysis of replication fork progression at individual centromeres and telomeres. a, **b**, Replication fork speed at individual centromeres (**a**) or telomeres (**b**). See Fig. 6 caption for details. The number of samples varies because no overlapping fork was detected at certain centromeres or telomeres in some samples; please note that (i) telomeres with no overlapping fork in any of the samples are not represented, (ii) «telomeres» correspond to subtelomeric sequences and do not comprise the terminal stretch of telomeric repeats and (iii) one sample, with a value of 6,384 bp/min, is out of bounds for CEN8. Two-sided contrast comparisons between a given centromere (**a**) or telomere (**b**) and the rest of the genome are indicated by a star ($p \leq 0.05$) or n.s. (not significant). Statistical analyses are detailed in Supplementary Table 2. CEN#, centromere of the indicated chromosome; TEL#L/R, left (L) or right (R) telomere of the indicated chromosome.
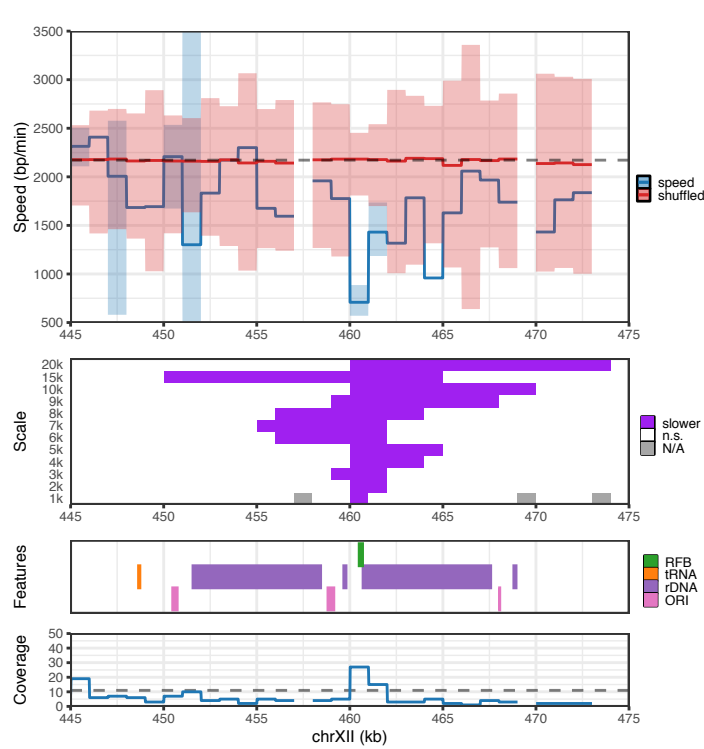
Supplementary Figure 11. Replication fork progression map of a portion of chromosome XII containing the rDNA locus. Panels from top to bottom: 1) median of experimental fork speeds (blue line) with 98% confidence interval of the median (light blue) and median of reshuffled speeds (red line) with 98% confidence interval of the median (light red) computed in 1 kb windows (dotted line, median fork speed in the whole genome); 2) results of Mann-Whitney-Wilcoxon tests with Holm correction (one-sided) performed along the chromosome to compare the speed distribution in a given window of a given width (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15 and 20 kb) to the speed distribution on the whole genome (purple, regions of lower fork speed; white, n.s., not significant; statistical significance was set to p < 0.01; grey, N/A, not applicable, regions with no fork); 3) position of selected genomic features (large and short horizontal light purple bars, 35S and 5S ribosomal genes, respectively; please note that the rDNA locus is composed of two rDNA units in the sacCer3 version of the yeast genome used in this study; RFB, replication fork barrier; ORI, known *S. cerevisiae* replication origins from[21]); 4) coverage of individual replication fork velocities (dotted line, median coverage of the genome). Please note that it was not possible to compute a finite confidence interval for the median of the experimental fork speeds for windows with a coverage ≤ 5.
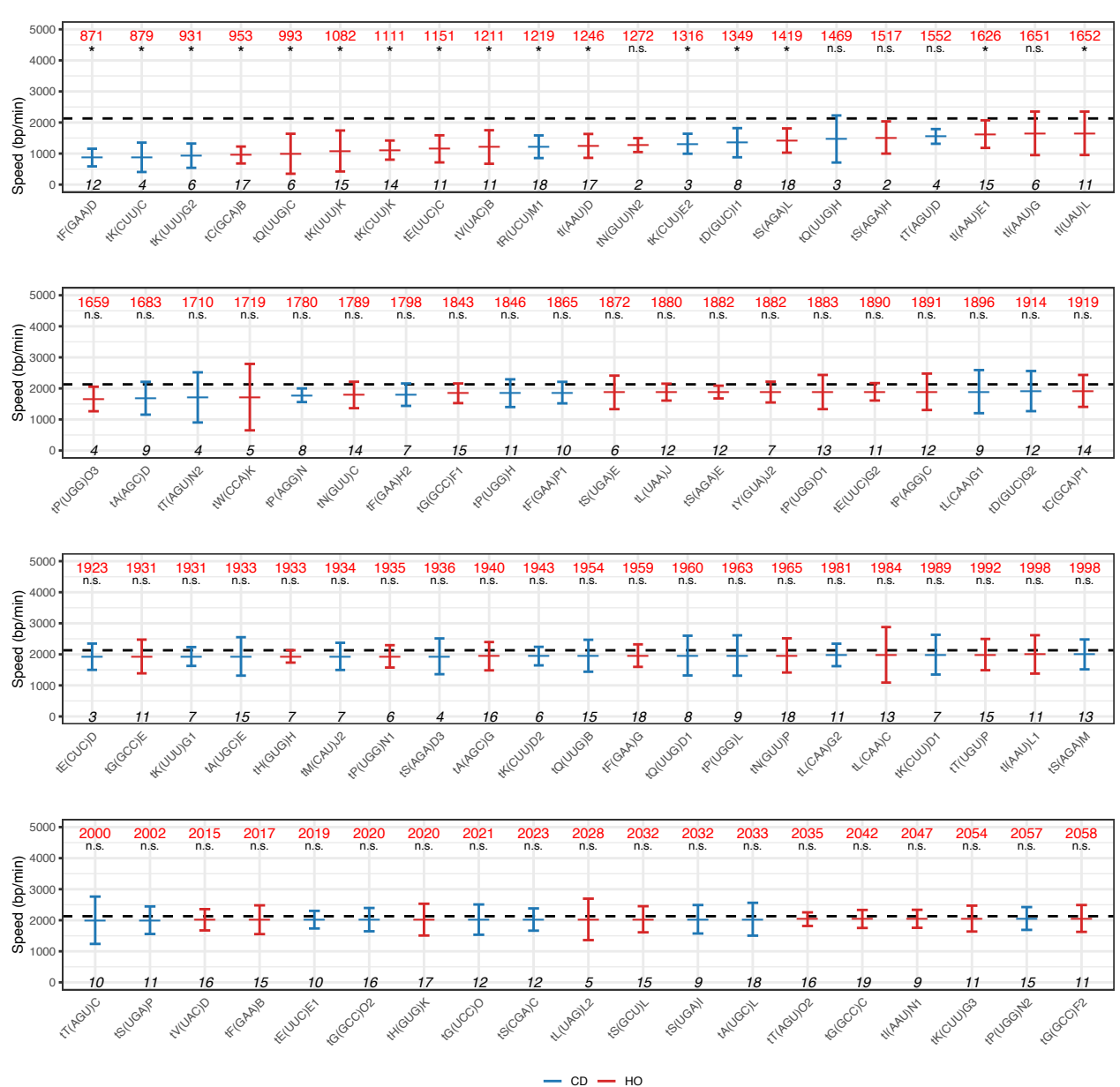
Supplementary Figure 12. Targeted analysis of replication fork progression at individual tRNA genes. The speed of forks overlapping individual tRNA genes was computed. The name of each tRNA gene is displayed below each plot; the predominant direction of replication relative to transcription (CD, co-directional, in blue; HO, head-on, in red; see Methods) is specified for every tRNA gene. tRNA genes are ordered according to increasing average fork speed. Centre line, average value of the mean speeds of forks overlapping the indicated tRNA gene in each sample, indicated in red on top (please note that individual data points are not shown for clarity); whiskers, standard deviation; bottom, number of samples (this number varies as no overlapping fork was detected at certain tRNA genes in some samples; please note that tRNA genes with no overlapping fork in any of the samples as well as tRNA genes with overlapping forks detected in only one sample, hence for which no contrast test could be performed, are not represented); dotted line at 2,130 bp/min, average fork speed of the 22 "Rest of the genome" samples. Two-sided contrast comparisons between a given tRNA gene and the rest of the genome are indicated by a star (p ≤ 0.05) or n.s. (not significant). Statistical analyses are detailed in Supplementary Table 2.
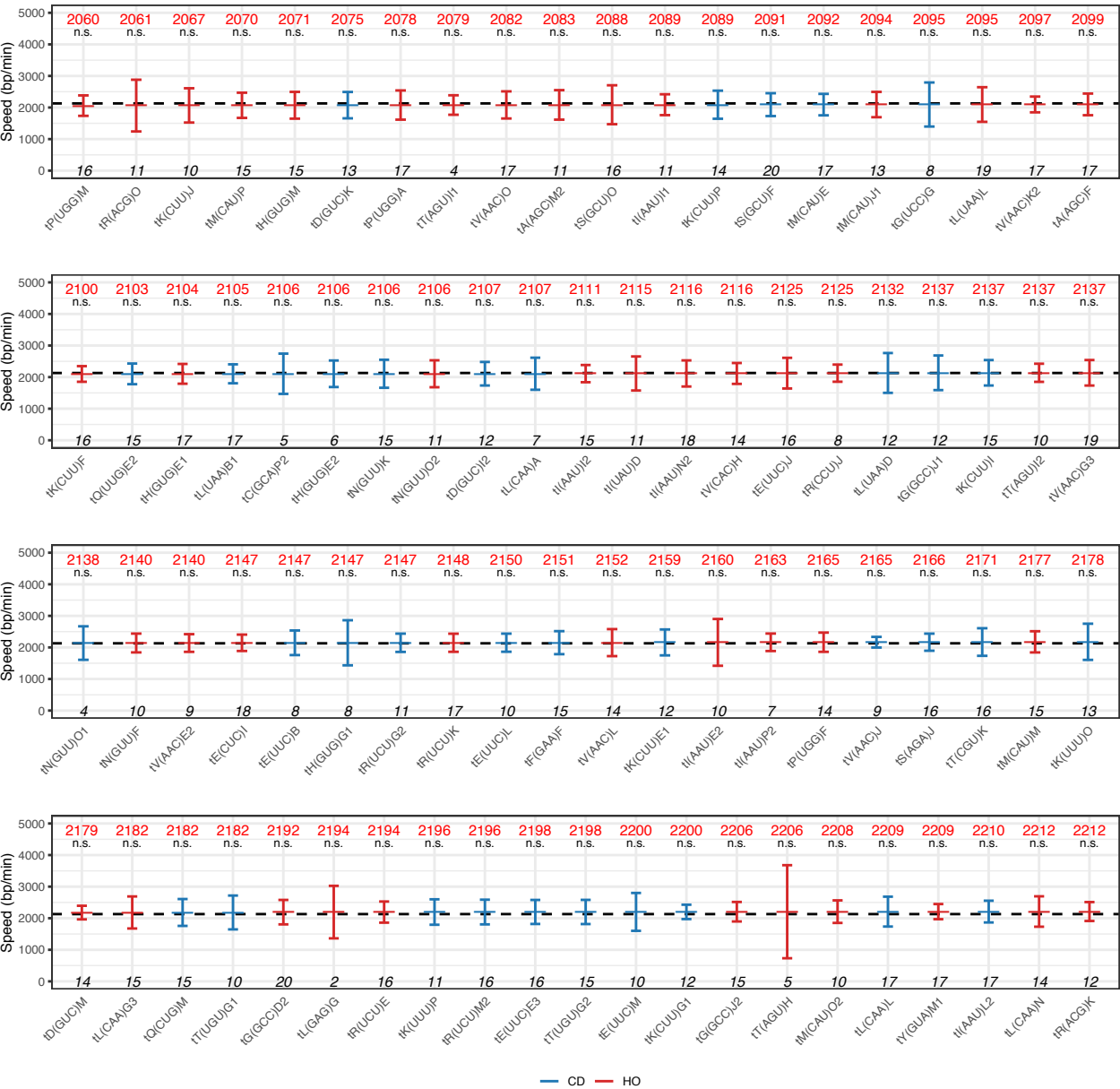
Supplementary Figure 12, continued.
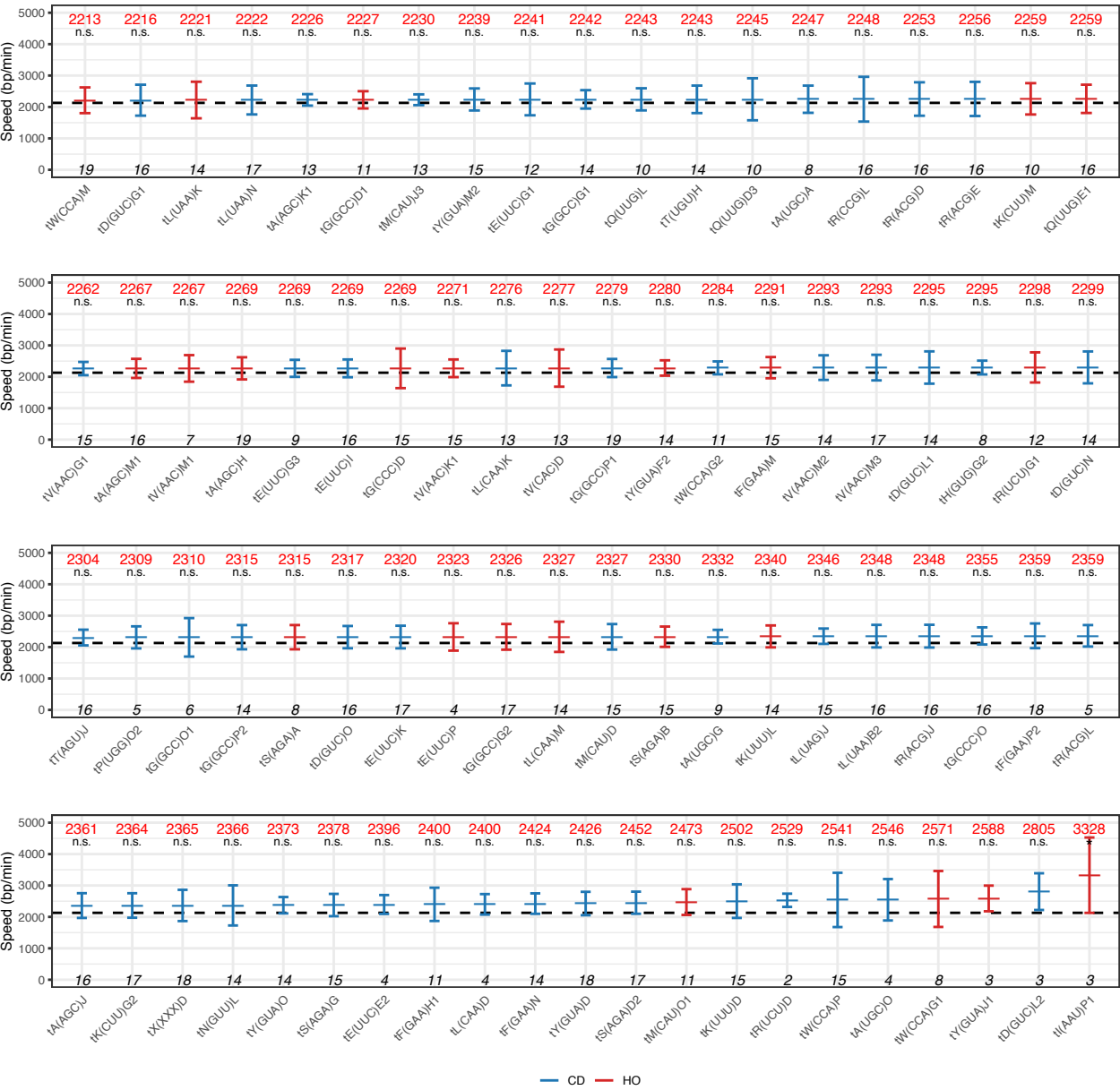
Supplementary Figure 12, continued.
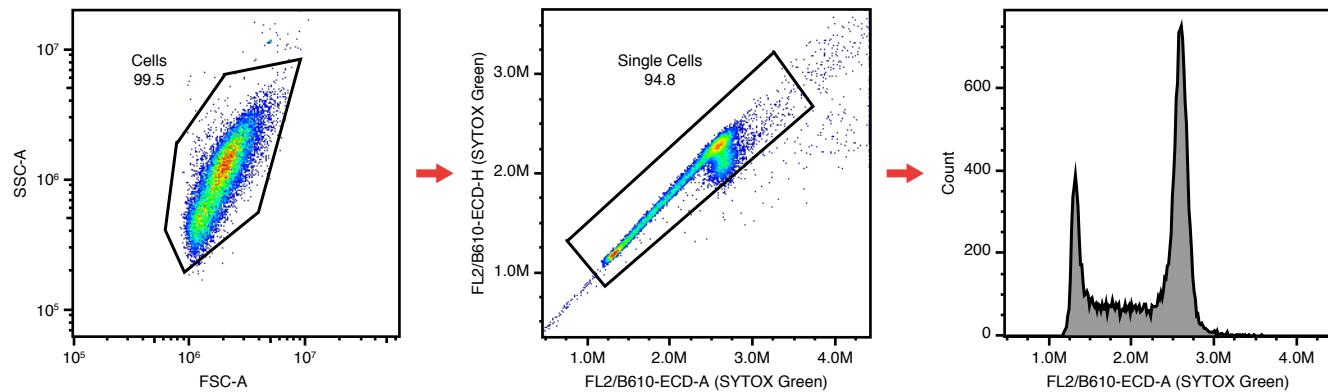
Supplementary Figure 13. Gating strategy for flow cytometry analysis. Cells are fixed in ethanol and DNA is counterstained with SYTOX Green prior to flow cytometry analysis. Cells are initially gated using the FSC-Area versus SSC-Area plot to remove debris (left panel), then interrogated by the ratios of area (FL2/B610-ECD-A) to height (FL2/B610-ECD-H) of the SYTOX Green signal to gate out cell doublets (middle panel). A histogram of the FL2/B610-ECD-Area (right panel) shows DNA content after gating.

Supplementary Table 1. Strains and plasmids used in this study. CO, codon-optimized for expression in yeast.

Yeast strain	Genotype	Reference
W303 (*URA3 bar1Δ*)	*MATa trp1-1 leu2-3,112 his3-11,15 bar1::LEU2*	This study
BT1	*MATa trp1-1 leu2-3,112 his3-11,15 bar1::LEU2 ura3-1::URA3-GPD-hsvTK$_{CO}$-ADH-hENT1$_{CO}$(5x)*	This study
BT2	*MATa ade2-1 trp1-1 leu2-3,112 his3-11,15 can1-100 bar1::LEU2 ura3-1::URA3-GPD-hsvTK(7x) aur1::AUR1C-ADH-hENT1$_{CO}$(3x)*	This study
BT3	*MATa trp1-1 leu2-3,112 his3-11,15 bar1::LEU2 ura3-1::URA3-GPD-hsvTK-ADH-hENT1(5x)*	This study
BT4	BT1 *rtt109::KANMX*	This study
BT5	BT1 *sml1::KANMX*	This study
BT6	BT1 *csm3::KANMX*	This study
BT7	BT1 *tof1::KANMX*	This study
BT8	BT1 *mrc1::KANMX*	This study
MCM869	*MATa ade2-1 trp1-1 leu2-3,112 his3-11,15 can1-100 bar1::LEU2 ura3-1::URA3-GPD-hsvTK(7x) aur1::AUR1C-ADH-hENT1$_{CO}$(3x) cdc21::KANMX*	Ma et al, 2012

Plasmid	Description	Reference
p306-BrdU-Inc	Integrating plasmid for incorporation of exogenous thymidine or thymidine analogs	Viggiani et al, 2006
pBL-hsvTK$_{CO}$-hENT1$_{CO}$	Integrating plasmid for incorporation of exogenous thymidine or thymidine analogs (transgenes codon-optimized for expression in yeast)	This study

Supplementary Table 2. Statistical analyses. "other", rest of the genome; n, sample size for each class tested; df, degree of freedom; nr, not relevant; adjusted p values from the same panel, according to Benjamini & Hochberg.

Figure	Comparison	Alternative hypothesis	Sample size	Test	p-value	Adjusted p-value
	0 and 1	effect difference	n = 22, n = 2 | df = 29	Contrast test in linear model	1e-4	1e-4
	0 and 2.5	effect difference	n = 22, n = 2 | df = 29	Contrast test in linear model	2e-15	2e-15
	0 and 5	effect difference	n = 22, n = 2 | df = 29	Contrast test in linear model	4e-18	6e-18
4A	0 and 10	effect difference	n = 22, n = 2 | df = 29	Contrast test in linear model	8e-21	1e-20
	0 and 25	effect difference	n = 22, n = 2 | df = 29	Contrast test in linear model	1e-26	3e-26
	0 and 50	effect difference	n = 22, n = 2 | df = 29	Contrast test in linear model	1e-28	5e-28
	0 and 100	effect difference	n = 22, n = 2 | df = 29	Contrast test in linear model	4e-31	3e-30
	BT1 and csm3	effect difference	n = 22, n = 2 | df = 26	Contrast test in linear model	1e-32	2e-32
	BT1 and mrc1	effect difference	n = 22, n = 2 | df = 26	Contrast test in linear model	4e-42	2e-41
4B	BT1 and rtt109	effect difference	n = 22, n = 2 | df = 26	Contrast test in linear model	4e-24	5e-24
	BT1 and sml1	effect difference	n = 22, n = 2 | df = 26	Contrast test in linear model	8e-21	8e-21
	BT1 and tof1	effect difference	n = 22, n = 2 | df = 26	Contrast test in linear model	2e-33	4e-33
	other and centromere	effect difference	n = 22, n = 22 | df = 81	Contrast test in linear model	1e-4	3e-4
6A	other and rDNA	effect difference	n = 22, n = 19 | df = 81	Contrast test in linear model	1e-11	5e-11
	other and telomere	effect difference	n = 22, n = 22 | df = 81	Contrast test in linear model	8e-4	0.001
	other and tRNA	effect difference	n = 22, n = 22 | df = 81	Contrast test in linear model	0.77	0.77
6B	CD and HO	effect difference	n = 22, n = 22 | df = 21	Contrast test in linear model	3e-8	*nr*
6C	Leading and lagging	effect difference	n = 22, n = 22 | df = 21	Contrast test in linear model	0.04	*nr*
	other and CEN1	effect difference	n = 22, n = 13 | df = 207	Contrast test in linear model	6e-4	2e-3
	other and CEN2	effect difference	n = 22, n = 4 | df = 207	Contrast test in linear model	0.41	0.41
	other and CEN3	effect difference	n = 22, n = 13 | df = 207	Contrast test in linear model	0.02	0.03
	other and CEN4	effect difference	n = 22, n = 13 | df = 207	Contrast test in linear model	0.03	0.04
	other and CEN5	effect difference	n = 22, n = 16 | df = 207	Contrast test in linear model	5e-4	0.002
	other and CEN6	effect difference	n = 22, n = 14 | df = 207	Contrast test in linear model	9e-4	0.002
	other and CEN7	effect difference	n = 22, n = 8 | df = 207	Contrast test in linear model	0.005	0.01
S10A	other and CEN8	effect difference	n = 22, n = 18 | df = 207	Contrast test in linear model	0.29	0.31
	other and CEN9	effect difference	n = 22, n = 11 | df = 207	Contrast test in linear model	2e-5	4e-4
	other and CEN10	effect difference	n = 22, n = 16 | df = 207	Contrast test in linear model	0.03	0.04
	other and CEN11	effect difference	n = 22, n = 18 | df = 207	Contrast test in linear model	0.01	0.02
	other and CEN12	effect difference	n = 22, n = 15 | df = 207	Contrast test in linear model	6e-5	5e-4
	other and CEN13	effect difference	n = 22, n = 16 | df = 207	Contrast test in linear model	0.09	0.10
	other and CEN14	effect difference	n = 22, n = 16 | df = 207	Contrast test in linear model	5e-4	2e-3
	other and CEN15	effect difference	n = 22, n = 18 | df = 207	Contrast test in linear model	0.02	0.03
	other and CEN16	effect difference	n = 22, n = 14 | df = 207	Contrast test in linear model	0.02	0.03
	other and TEL02L	effect difference	n = 22, n = 17 | df = 245	Contrast test in linear model	0.03	0.04
	other and TEL04R	effect difference	n = 22, n = 12 | df = 245	Contrast test in linear model	0.008	0.02
	other and TEL05L	effect difference	n = 22, n = 16 | df = 245	Contrast test in linear model	0.006	0.02
	other and TEL05R	effect difference	n = 22, n = 13 | df = 245	Contrast test in linear model	0.03	0.04
	other and TEL06L	effect difference	n = 22, n = 12 | df = 245	Contrast test in linear model	0.02	0.03
	other and TEL07R	effect difference	n = 22, n = 21 | df = 245	Contrast test in linear model	0.01	0.03
	other and TEL08L	effect difference	n = 22, n = 15 | df = 245	Contrast test in linear model	0.02	0.03
S10B	other and TEL08R	effect difference	n = 22, n = 9 | df = 245	Contrast test in linear model	2e-4	0.001
	other and TEL09L	effect difference	n = 22, n = 19 | df = 245	Contrast test in linear model	0.10	0.11
	other and TEL10L	effect difference	n = 22, n = 20 | df = 245	Contrast test in linear model	0.43	0.43
	other and TEL12L	effect difference	n = 22, n = 18 | df = 245	Contrast test in linear model	0.001	0.005
	other and TEL12R	effect difference	n = 22, n = 20 | df = 245	Contrast test in linear model	9e-5	8e-4
	other and TEL14L	effect difference	n = 22, n = 19 | df = 245	Contrast test in linear model	0.004	0.01
	other and TEL15R	effect difference	n = 22, n = 18 | df = 245	Contrast test in linear model	0.03	0.04
	other and TEL16L	effect difference	n = 22, n = 19 | df = 245	Contrast test in linear model	0.10	0.11
	other and TEL16R	effect difference	n = 22, n = 13 | df = 245	Contrast test in linear model	9e-5	8e-4

Figure	Comparison	Alternative hypothesis	Sample size	Test	p-value	Adjusted p-value
	other and tA(AGC)D	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.01	0.13
	other and tA(AGC)F	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.82	0.97
	other and tA(AGC)G	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.19	0.68
	other and tA(AGC)H	effect difference	n = 22, n = 19 | df = 2728	Contrast test in linear model	0.30	0.78
	other and tA(AGC)J	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.11	0.55
	other and tA(AGC)K1	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.53	0.86
	other and tA(AGC)M1	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.35	0.78
	other and tA(AGC)M2	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.81	0.97
	other and tA(UGC)A	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.55	0.87
	other and tA(UGC)E	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.18	0.65
	other and tA(UGC)G	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.23	0.72
	other and tA(UGC)L	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.50	0.84
	other and tA(UGC)O	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.08	0.50
	other and tC(GCA)B	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	2e-16	5e-14
	other and tC(GCA)P1	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.16	0.64
	other and tC(GCA)P2	effect difference	n = 22, n = 5 | df = 2728	Contrast test in linear model	0.95	0.97
	other and tD(GUC)G1	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.55	0.88
	other and tD(GUC)G2	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.17	0.65
	other and tD(GUC)I1	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	2e-5	4e-4
	other and tD(GUC)I2	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.88	0.97
	other and tD(GUC)K	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.73	0.95
	other and tD(GUC)L1	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.27	0.76
	other and tD(GUC)L2	effect difference	n = 22, n = 3 | df = 2728	Contrast test in linear model	0.01	0.14
	other and tD(GUC)M	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.77	0.97
	other and tD(GUC)N	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.25	0.73
	other and tD(GUC)O	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.20	0.68
	other and tE(CUC)D	effect difference	n = 22, n = 3 | df = 2728	Contrast test in linear model	0.44	0.82
	other and tE(CUC)I	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.90	0.97
	other and tE(UUC)B	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.97	0.98
	other and tE(UUC)C	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	2e-9	6e-8
	other and tE(UUC)E1	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.53	0.86
	other and tE(UUC)E2	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.26	0.74
	other and tE(UUC)E3	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.63	0.92
	other and tE(UUC)G1	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.49	0.84
	other and tE(UUC)G2	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.14	0.59
	other and tE(UUC)G3	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.42	0.82
	other and tE(UUC)I	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.33	0.78
	other and tE(UUC)J	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.97	0.98
S12	other and tE(UUC)K	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.18	0.65
	other and tE(UUC)L	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.90	0.97
	other and tE(UUC)M	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.69	0.94
	other and tE(UUC)P	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.41	0.82
	other and tF(GAA)B	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.44	0.82
	other and tF(GAA)D	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	2e-15	3e-13
	other and tF(GAA)F	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.90	0.97
	other and tF(GAA)G	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.23	0.72
	other and tF(GAA)H1	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.09	0.53
	other and tF(GAA)H2	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.09	0.53
	other and tF(GAA)M	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.28	0.78
	other and tF(GAA)N	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.05	0.39
	other and tF(GAA)P1	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.13	0.57
	other and tF(GAA)P2	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.10	0.54
	other and tG(CCC)D	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.36	0.78
	other and tG(CCC)O	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.12	0.57
	other and tG(GCC)C	effect difference	n = 22, n = 19 | df = 2728	Contrast test in linear model	0.51	0.86
	other and tG(GCC)D1	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.57	0.89
	other and tG(GCC)D2	effect difference	n = 22, n = 20 | df = 2728	Contrast test in linear model	0.65	0.92
	other and tG(GCC)E	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.22	0.72
	other and tG(GCC)F1	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.05	0.39
	other and tG(GCC)F2	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.66	0.93
	other and tG(GCC)G1	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.46	0.83
	other and tG(GCC)G2	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.16	0.64
	other and tG(GCC)J1	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.94	0.97
	other and tG(GCC)J2	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.61	0.91
	other and tG(GCC)O1	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	0.37	0.79
	other and tG(GCC)O2	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.46	0.83
	other and tG(GCC)P1	effect difference	n = 22, n = 19 | df = 2728	Contrast test in linear model	0.28	0.78
	other and tG(GCC)P2	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.21	0.71
	other and tG(UCC)G	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.86	0.97
	other and tG(UCC)O	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.47	0.84
	other and tH(GUG)E1	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.85	0.97
	other and tH(GUG)E2	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	0.94	0.97
	other and tH(GUG)G1	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.92	0.97
	other and tH(GUG)G2	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.33	0.78
	other and tH(GUG)H	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.30	0.78
	other and tH(GUG)K	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.44	0.82
	other and tH(GUG)M	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.71	0.95

Figure	Comparison	Alternative hypothesis	Sample size	Test	p-value	Adjusted p-value
	other and tI(AAU)D	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	6e-10	2e-8
	other and tI(AAU)E1	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	6e-4	0.01
	other and tI(AAU)E2	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.84	0.97
	other and tI(AAU)G	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	0.02	0.16
	other and tI(AAU)I1	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.82	0.97
	other and tI(AAU)I2	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.89	0.97
	other and tI(AAU)L1	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.42	0.82
	other and tI(AAU)L2	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.58	0.89
	other and tI(AAU)N1	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.64	0.92
	other and tI(AAU)N2	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.94	0.97
	other and tI(AAU)P1	effect difference	n = 22, n = 3 | df = 2728	Contrast test in linear model	8e-6	2e-4
	other and tI(AAU)P2	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.80	0.97
	other and tI(UAU)D	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.93	0.97
	other and tI(UAU)L	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.003	0.047
	other and tK(CUU)C	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	1e-7	3e-6
	other and tK(CUU)D1	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.49	0.84
	other and tK(CUU)D2	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	0.36	0.78
	other and tK(CUU)E1	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.85	0.97
	other and tK(CUU)E2	effect difference	n = 22, n = 3 | df = 2728	Contrast test in linear model	0.003	0.047
	other and tK(CUU)F	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.84	0.97
	other and tK(CUU)G1	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.64	0.92
	other and tK(CUU)G2	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.10	0.53
	other and tK(CUU)G3	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.64	0.92
	other and tK(CUU)I	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.94	0.97
	other and tK(CUU)J	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.71	0.95
	other and tK(CUU)K	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	2e-11	9e-10
	other and tK(CUU)M	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.42	0.82
	other and tK(CUU)P	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.76	0.97
	other and tK(UUU)D	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.01	0.13
	other and tK(UUU)G1	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.30	0.78
	other and tK(UUU)G2	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	4e-9	1e-7
	other and tK(UUU)K	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	2e-12	1e-10
	other and tK(UUU)L	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.15	0.63
	other and tK(UUU)O	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.74	0.96
	other and tK(UUU)P	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.67	0.94
	other and tL(CAA)A	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.89	0.97
	other and tL(CAA)C	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.33	0.78
	other and tL(CAA)D	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.24	0.73
S12 (continued)	other and tL(CAA)G1	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.18	0.65
	other and tL(CAA)G2	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.36	0.78
	other and tL(CAA)G3	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.73	0.95
	other and tL(CAA)K	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.36	0.78
	other and tL(CAA)L	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.57	0.89
	other and tL(CAA)M	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.20	0.68
	other and tL(CAA)N	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.57	0.89
	other and tL(GAG)G	effect difference	n = 22, n = 2 | df = 2728	Contrast test in linear model	0.79	0.97
	other and tL(UAA)B1	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.87	0.97
	other and tL(UAA)B2	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.14	0.58
	other and tL(UAA)D	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.98	0.98
	other and tL(UAA)J	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.12	0.56
	other and tL(UAA)K	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.54	0.87
	other and tL(UAA)L	effect difference	n = 22, n = 19 | df = 2728	Contrast test in linear model	0.79	0.97
	other and tL(UAA)N	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.52	0.86
	other and tL(UAG)J	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.14	0.59
	other and tL(UAG)L2	effect difference	n = 22, n = 5 | df = 2728	Contrast test in linear model	0.68	0.94
	other and tM(CAU)D	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.17	0.65
	other and tM(CAU)E	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.81	0.97
	other and tM(CAU)J1	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.81	0.97
	other and tM(CAU)J2	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.31	0.78
	other and tM(CAU)J3	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.52	0.86
	other and tM(CAU)M	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.74	0.96
	other and tM(CAU)O1	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.03	0.29
	other and tM(CAU)O2	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.65	0.92
	other and tM(CAU)P	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.70	0.95
	other and tN(GUU)C	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.02	0.21
	other and tN(GUU)F	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.91	0.97
	other and tN(GUU)K	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.87	0.97
	other and tN(GUU)L	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.12	0.56
	other and tN(GUU)N2	effect difference	n = 22, n = 2 | df = 2728	Contrast test in linear model	0.01	0.13
	other and tN(GUU)O1	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.95	0.97
	other and tN(GUU)O2	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.87	0.97
	other and tN(GUU)P	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.24	0.73
	other and tP(AGG)C	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.13	0.58
	other and tP(AGG)N	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.05	0.39
	other and tP(UGG)A	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.72	0.95
	other and tP(UGG)F	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.82	0.97
	other and tP(UGG)H	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.09	0.53

Figure	Comparison	Alternative hypothesis	Sample size	Test	p-value	Adjusted p-value
	other and tP(UGG)L	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.35	0.78
	other and tP(UGG)M	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.62	0.92
	other and tP(UGG)N1	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	0.35	0.78
	other and tP(UGG)N2	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.61	0.91
	other and tP(UGG)O1	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.12	0.56
	other and tP(UGG)O2	effect difference	n = 22, n = 5 | df = 2728	Contrast test in linear model	0.39	0.81
	other and tP(UGG)O3	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.06	0.41
	other and tQ(CUG)M	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.70	0.95
	other and tQ(UUG)B	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.24	0.73
	other and tQ(UUG)C	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	2e-8	5e-7
	other and tQ(UUG)D1	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.35	0.78
	other and tQ(UUG)D3	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.48	0.84
	other and tQ(UUG)E1	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.36	0.78
	other and tQ(UUG)E2	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.86	0.97
	other and tQ(UUG)H	effect difference	n = 22, n = 3 | df = 2728	Contrast test in linear model	0.02	0.16
	other and tQ(UUG)L	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.49	0.84
	other and tR(ACG)D	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.40	0.82
	other and tR(ACG)E	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.38	0.81
	other and tR(ACG)J	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.13	0.58
	other and tR(ACG)K	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.59	0.89
	other and tR(ACG)L	effect difference	n = 22, n = 5 | df = 2728	Contrast test in linear model	0.33	0.78
	other and tR(ACG)O	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.69	0.94
	other and tR(CCG)L	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.43	0.82
	other and tR(CCU)J	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.99	0.99
	other and tR(UCU)D	effect difference	n = 22, n = 2 | df = 2728	Contrast test in linear model	0.20	0.68
	other and tR(UCU)E	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.67	0.94
	other and tR(UCU)G1	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.31	0.78
	other and tR(UCU)G2	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.88	0.97
	other and tR(UCU)K	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.89	0.97
	other and tR(UCU)M1	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	9e-11	4e-9
	other and tR(UCU)M2	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.65	0.92
	other and tS(AGA)A	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.32	0.78
	other and tS(AGA)B	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.16	0.64
	other and tS(AGA)D2	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.02	0.21
	other and tS(AGA)D3	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.40	0.82
	other and tS(AGA)E	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.12	0.57
	other and tS(AGA)G	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.10	0.53
	other and tS(AGA)H	effect difference	n = 22, n = 2 | df = 2728	Contrast test in linear model	0.05	0.39
S12 (continued)	other and tS(AGA)J	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.78	0.97
	other and tS(AGA)L	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	4e-7	8e-6
	other and tS(AGA)M	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.40	0.82
	other and tS(CGA)C	effect difference	n = 22, n = 12 | df = 2728	Contrast test in linear model	0.50	0.84
	other and tS(GCU)F	effect difference	n = 22, n = 20 | df = 2728	Contrast test in linear model	0.77	0.97
	other and tS(GCU)L	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.49	0.84
	other and tS(GCU)O	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.77	0.97
	other and tS(UGA)E	effect difference	n = 22, n = 6 | df = 2728	Contrast test in linear model	0.22	0.72
	other and tS(UGA)I	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.57	0.89
	other and tS(UGA)P	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.42	0.82
	other and tT(AGU)C	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.43	0.82
	other and tT(AGU)D	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.02	0.16
	other and tT(AGU)H	effect difference	n = 22, n = 5 | df = 2728	Contrast test in linear model	0.70	0.95
	other and tT(AGU)I1	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.87	0.97
	other and tT(AGU)I2	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.94	0.97
	other and tT(AGU)J	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.23	0.72
	other and tT(AGU)N2	effect difference	n = 22, n = 4 | df = 2728	Contrast test in linear model	0.09	0.53
	other and tT(AGU)O1	effect difference	n = 22, n = 1 | df = 2728	Contrast test in linear model	0.84	0.97
	other and tT(AGU)O2	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.52	0.86
	other and tT(CGU)K	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.78	0.97
	other and tT(UGU)G1	effect difference	n = 22, n = 10 | df = 2728	Contrast test in linear model	0.72	0.95
	other and tT(UGU)G2	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.63	0.92
	other and tT(UGU)H	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.45	0.82
	other and tT(UGU)P	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.35	0.78
	other and tV(AAC)E2	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.92	0.97
	other and tV(AAC)G1	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.36	0.78
	other and tV(AAC)G3	effect difference	n = 22, n = 19 | df = 2728	Contrast test in linear model	0.96	0.98
	other and tV(AAC)J	effect difference	n = 22, n = 9 | df = 2728	Contrast test in linear model	0.81	0.97
	other and tV(AAC)K1	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.34	0.78
	other and tV(AAC)K2	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.81	0.97
	other and tV(AAC)L	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.88	0.97
	other and tV(AAC)M1	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.45	0.82
	other and tV(AAC)M2	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.27	0.77
	other and tV(AAC)M3	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.25	0.74
	other and tV(AAC)O	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.75	0.96
	other and tV(CAC)D	effect difference	n = 22, n = 13 | df = 2728	Contrast test in linear model	0.34	0.78
	other and tV(CAC)H	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.93	0.97
	other and tV(UAC)B	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	2e-8	5e-7
	other and tV(UAC)D	effect difference	n = 22, n = 16 | df = 2728	Contrast test in linear model	0.43	0.82

Figure	Comparison	Alternative hypothesis	Sample size	Test	p-value	Adjusted p-value
	other and tW(CCA)G1	effect difference	n = 22, n = 8 | df = 2728	Contrast test in linear model	0.02	0.16
	other and tW(CCA)G2	effect difference	n = 22, n = 11 | df = 2728	Contrast test in linear model	0.33	0.78
	other and tW(CCA)K	effect difference	n = 22, n = 5 | df = 2728	Contrast test in linear model	0.07	0.44
	other and tW(CCA)M	effect difference	n = 22, n = 19 | df = 2728	Contrast test in linear model	0.54	0.87
	other and tW(CCA)P	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.005	0.07
S12 (continued)	other and tX(XXX)D	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.10	0.53
	other and tY(GUA)D	effect difference	n = 22, n = 18 | df = 2728	Contrast test in linear model	0.04	0.30
	other and tY(GUA)F2	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.32	0.78
	other and tY(GUA)J1	effect difference	n = 22, n = 3 | df = 2728	Contrast test in linear model	0.08	0.53
	other and tY(GUA)J2	effect difference	n = 22, n = 7 | df = 2728	Contrast test in linear model	0.21	0.69
	other and tY(GUA)M1	effect difference	n = 22, n = 17 | df = 2728	Contrast test in linear model	0.58	0.89
	other and tY(GUA)M2	effect difference	n = 22, n = 15 | df = 2728	Contrast test in linear model	0.47	0.83
	other and tY(GUA)O	effect difference	n = 22, n = 14 | df = 2728	Contrast test in linear model	0.11	0.55

EXP name	temp (°C)	B_pulse (µM)	t_pulse (min)	T_chase (µM)	Strain	mutant	HU (mM)	Sequencing Type	MinKnow	MinKnow Core	n_reads total	n_reads no alignment	percent not aligned	n_reads (>5kb)	n_reads chrM	n_reads	sumlength (nt)	b2a.thr	n_reads_RDP3	n_reads_forks	med_len (reads with forks) (nt)	n_forks	speed_med (bp/min)	dY_med	nb_init	nb_ter	fork_dens (per Mb)	init_dens (per Mb)	ter_dens (per Mb)	file_name	file_md5
BT1_B0	30	0		0	BT1	WT	0	Minion	20.10.3	4.1.2	1011314	116795	11.5%	524315	139714	384601	1.12E+10	0.02	739	48	28225	55	588	0.143	2	5	0.0049	0.0002	0.0004	BT1_B0.tar.gz	03e2c7bfc8dd036ccae3670661fa805c
W303_B100	30	100		1000	W303	WT	0	Gridion	21.02.5	4.2.5	441196	5165	1.2%	326366	63582	262784	1.01E+10	0.02	697	22	36059	24	622	0.133	1	1	0.0024	0.0001	0.0001	W303_B100.tar.gz	4576f8ac52c4729a526300df77b5592