

# Alignment of Synchronized Time-Series Data Using the Characterizing Loss of Cell Cycle Synchrony Model for Cross-Experiment Comparisons

Sophia A. Campione<sup>1</sup>, Christina M. Kelliher<sup>2</sup>, David A. Orlando<sup>3</sup>, Trung Q. Tran<sup>4</sup>, Steven B. Haase<sup>1</sup>

<sup>1</sup> Department of Biology, Duke University <sup>2</sup> Department of Biology, University of Massachusetts <sup>3</sup> Orlando Data Science LLC <sup>4</sup> Department of Computer Science, Duke University

## Corresponding Author

Steven B. Haase

steve.haase@duke.edu

## Citation

Campione, S.A., Kelliher, C.M., Orlando, D.A., Tran, T.Q., Haase, S.B. Alignment of Synchronized Time-Series Data Using the Characterizing Loss of Cell Cycle Synchrony Model for Cross-Experiment Comparisons. *J. Vis. Exp.* (196), e65466, doi:10.3791/65466 (2023).

## Date Published

June 9, 2023

## DOI

10.3791/65466

## URL

jove.com/video/65466

## Abstract

Investigating the cell cycle often depends on synchronizing cell populations to measure various parameters in a time series as the cells traverse the cell cycle. However, even under similar conditions, replicate experiments display differences in the time required to recover from synchrony and to traverse the cell cycle, thus preventing direct comparisons at each time point. The problem of comparing dynamic measurements across experiments is exacerbated in mutant populations or in alternative growth conditions that affect the synchrony recovery time and/or the cell-cycle period.

We have previously published a parametric mathematical model named Characterizing Loss of Cell Cycle Synchrony (CLOCCS) that monitors how synchronous populations of cells release from synchrony and progress through the cell cycle. The learned parameters from the model can then be used to convert experimental time points from synchronized time-series experiments into a normalized time scale (lifeline points). Rather than representing the elapsed time in minutes from the start of the experiment, the lifeline scale represents the progression from synchrony to cell-cycle entry and then through the phases of the cell cycle. Since lifeline points correspond to the phase of the average cell within the synchronized population, this normalized time scale allows for direct comparisons between experiments, including those with varying periods and recovery times. Furthermore, the model has been used to align cell-cycle experiments between different species (e.g., *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe*), thus enabling direct comparison of cell-cycle measurements, which may reveal evolutionary similarities and differences.

## Introduction

Time-series measurements made on synchronized populations of cells as they progress through the cell cycle is a standard method for investigating the mechanisms that control cell-cycle progression<sup>1,2,3,4,5,6,7,8</sup>. The ability to make comparisons across synchrony/release time-series experiments is vital to our understanding of these dynamic processes. The use of replicate experiments to corroborate findings can increase the confidence in the reproducibility of the conclusions. Furthermore, comparisons between environmental conditions, across mutants, and even between species can uncover many new insights into cell-cycle regulation. However, interexperimental variability in the recovery from synchrony and in the speed of cell-cycle progression impairs the ability to make time-point-to-time-point comparisons across replicates or between experiments with altered cell-cycle timing. Due to these challenges, replicates are often not included for the full time series (e.g., Spellman et al.<sup>4</sup>). When replicates for the entire time series are gathered, the data cannot be analyzed in aggregate, but rather a single replicate is used for analysis, and other replicates are often relegated to supplemental figures (e.g., Orlando et al.<sup>8</sup>). Furthermore, comparisons between experiments with different recovery or cell-cycle progression characteristics are difficult. The measurements of smaller intervals between an event of interest and a cell-cycle landmark (e.g., bud emergence, S-phase entry, or anaphase onset) can help reduce errors if these landmark events are tracked<sup>1,2,3,9,10,11,12</sup>. However, subtle but important differences may remain undetected or obscured using these *ad hoc* methods. Finally, single-cell analyses allow for analyzing cell-cycle progression without relying on synchronization or alignment<sup>13</sup>, though large-scale

measurements in single-cell studies can be challenging and costly.

To overcome these difficulties, we developed the Characterizing Loss of Cell Cycle Synchrony (CLOCCS) model to aid the analysis of time-series measurements made on synchronized populations<sup>14,15</sup>. CLOCCS is a flexible mathematical model that describes the distribution of synchronized cells across cell-cycle phases as they are released from synchrony and progress through the cell cycle. The branching process framework enables the model to account for the asymmetric qualities of mother and daughter cells after division, as observed in *S. cerevisiae*, while still being useful for organisms that divide by fission, such as *S. pombe*. The model can take inputs from a diverse set of measurement types to specify the cell-cycle phase. It can ingest budding cell-cycle phase data, which includes measurements of the percent budded cells over time, allowing for the estimation of the number of cells outside of the unbudded G1 phase<sup>14,15</sup>. The model can also ingest flow cytometric data that measures the DNA content, thus enabling the assessment of landmark transitions from G1 to S, S to G2, and M to G1<sup>15</sup>. Fluorescent morphological markers can also be used to identify the cell-cycle phase. The fluorescent labeling of myosin rings, nuclei, and spindle pole bodies (SPBs) can be used to determine the cell-cycle phase, and these were incorporated into the CLOCCS model<sup>11</sup>; however, these measurements will not be described in this protocol. Additionally, the septation index was used as an input for modeling data from *S. pombe*<sup>14</sup>. Thus, the model can be used for cell-cycle analyses in a variety of organisms and can be further expanded.

CLOCCS is a parametric model that allows for the full Bayesian inference of multiple parameters from the input data (e.g., budding percentage, DNA content). These parameters include the recovery time from synchrony, the length of the cell-cycle period (estimated separately for mother and daughter cells), and the average cell-cycle position of the cells at each time point. These parameters represent the behavior of the average cell in the population, enabling the researcher to map each time point to a cell-cycle position expressed as a lifeline point. The conversion to lifeline points depends on the CLOCCS parameters  $\lambda$  and  $\mu_0$  ( $\mu_0$ )<sup>14,15</sup>. The parameter  $\lambda$  corresponds to the average cell-cycle period of the mother cells. However, due to the mother-daughter delay<sup>14,15</sup>, this is not the average cell-cycle period of the full population that includes both the mother and daughter cells. CLOCCS additionally infers the parameter  $\delta$  ( $\delta$ ), which corresponds to the mother-daughter delay and, thus, allows for the calculation of the average cell-cycle period of the full population. Finally, because each experiment begins after release from cell-cycle synchronization, the time required to recover from the synchronization method is represented by the CLOCCS parameter  $\mu_0$ . CLOCCS fits a model to the input cell-cycle phase data and then infers these parameters using a random walk Markov chain Monte Carlo algorithm<sup>14,15</sup>. By mapping multiple experiments to a common cell-cycle lifeline time scale, direct phase-specific comparisons can be made between replicates or experiments where the recovery time or cell-cycle periods are not identical<sup>8,14,15</sup>.

As synchronized populations lose synchrony at some rate over the course of the time series<sup>14,15,16,17</sup>, variability in the rate of synchrony loss can also impede quantitative comparisons across experiments. By identifying the location of populations and the variance in their distributions, CLOCCS accounts for differences in rates of synchrony

loss. This powerful tool allows for specific and detailed comparisons across experiments, thus providing the ability to directly make relevant comparisons not only between replicates but also between environmental conditions, mutants, and even species that have dramatically different cell-cycle timing<sup>14,15</sup>.

This paper describes a method using CLOCCS to estimate parameters by fitting data from synchrony/release time-series experiments, map the data to a common lifeline scale, and then make relevant comparisons between replicates or experiments. Lifeline alignment allows for direct phase-specific comparisons across these experiments, which allows for the aggregation and comparison of replicates and for making more relevant comparisons across experiments with different recovery timings and cell-cycle periods.

## Protocol

### 1. Collecting cell-cycle phase and experimental data

1. Synchronize the cells with respect to the cell cycle using the desired synchronization method (e.g., centrifugal elutriation as described in Leman et al.<sup>18</sup> or mating pheromone arrest as described in Rosebrock<sup>19</sup>; both Leman et al.<sup>18</sup> and Rosebrock<sup>19</sup> also include methods for the release from synchrony). Begin sampling throughout the time series, ensuring that the time series is at least two full cell-cycle periods in length, and optimally, collect at least 10 samples per cell cycle. At each time point, collect a sample for cell-cycle phase data (budding or flow cytometry) and a sample for experimental data, as described below.

2. If using budding data as the cell-cycle phase data, collect data on budding for the CLOCCS alignment.

1. Sample throughout the time series. For each time point, collect cells, and fix them by mixing 200  $\mu$ L of sonicated cell culture with 200  $\mu$ L of fixative solution, as described in Leman et al.<sup>18</sup>.

2. For standard budding, count at least 200 cells per time point using a transmitted light microscope with a 40x objective and a hemocytometer. Add the cell sample from step 1.2.1 to the hemocytometer, and dilute if the density prevents counting. Record the number of budded and unbudded cells at each time point. Calculate the percent of budded cells, and plot for each time point in a budding curve.

**NOTE:** Other methods of specifying the cell-cycle phase information are available, but these are not described in this protocol. The other methods are described in the CLOCCS readme and in a previous work<sup>11</sup>.

3. If using flow-cytometric DNA content data as the cell-cycle phase data, collect flow cytometry DNA staining data for the flow-cytometric CLOCCS alignment.

1. Sample throughout the time series. For each time point, collect cells, and fix them as described in Haase and Reed<sup>20</sup>.

2. Stain the DNA, and analyze using standard flow cytometric analysis. A recommended staining protocol for *S. cerevisiae* is described in Haase and Reed<sup>20</sup>.

4. Collect associated omics or related experimental data. For standard transcriptomic data, collect as described in Leman et al.<sup>18</sup> and Kelliher et al.<sup>21,22</sup>. Ensure that

the data are associated with time points containing cell-cycle phase data to allow for downstream alignment. For optimal alignment, ensure that each time point containing experimental data also has phase data associated with it.

**NOTE:** The experimental data can take many forms. Traditionally, we use the alignment method described for aligning time-series transcriptomic experiments. However, any type of data associated with time points can be aligned (i.e., proteomics<sup>22</sup>).

## 2. Installing the required software

**NOTE:** This section assumes that Conda, Java 19, and Git are already installed (**Table of Materials**).

1. Download the CLOCCS\_alignment repo by entering the following command into the terminal:

```
git clone https://gitlab.com/haase-lab-group/cloccs_alignment.git
```

2. Create a Conda environment using the conda\_req.yml file by entering the following command into the terminal in the folder where the CLOCCS\_alignment repo was cloned:

```
conda env create -f conda_req.yml
```

## 3. Using CLOCCS to parameterize the experiments

1. Double-click on the cloccs\_v2023.jar file in the CLOCCS folder in the CLOCCS\_alignment repo, and wait for a graphical user interface to open. This screen allows for inputting options for the CLOCCS run and displays the results once run.

2. Input the general settings.

1. Set **Sim Anneal**, **Burn In**, and **Iterations** by typing in the associated text input boxes. **Sim Anneal**

(simulated annealing) identifies good starting parameter values, **Burn In** searches for posterior modes, and the **final stage** allows for all posterior inferences to be drawn. Higher values increase the run-time but also increase the accuracy.

2. Input the experimental conditions by specifying the **temperature** in **Celsius** and the **synchronization method** using the text box labeled **Temperature** and the dropdown menu **Synchro. Method**, respectively.
3. **Optionally** configure the advanced settings in the **Advanced Settings menu**. The advanced settings allow for priors to be set for each of the parameters ("**mu0**", "**sigma0**", "**sigmav**", "**lambda**", "**bud.start**", "**bud.end**").

**NOTE:** More information regarding the advanced settings can be found in the readme.txt in the CLOCCS folder of the CLOCCS\_alignment repo.

3. Input the settings for use with the budding data.
  1. Choose the appropriate selection from the **Model Type** dropdown menu. The default option **Bud** is for standard budding information for budding yeast.

**NOTE:** Other more advanced options also exist in the dropdown menu: **Mutant** for budding information for mutants that undergo multiple budding cycles without division, **BudSSLSMR** for budding information and additional spindle pole body and myosin ring information, and **BudNucDivNeck** for budding information and additional dividing and bud neck nuclei information. These advanced options are described in the CLOCCS readme and in previous work<sup>11,14,15</sup>.

2. Import the data using the **Data Import panel** by typing into the text input boxes or by uploading a file by clicking on the **Select File** button. The first column specifies the time points. The remaining two columns specify the budding data and can take any of the following options: the number of unbudded cells (**No Bud**), the number of budded cells (**Bud**), or the total number of cells (**Total**).

4. Input the settings for use with the flow cytometric data. For each experiment, run **either** step 3.3 or step 3.4.

**NOTE:** Flow cytometric data and budding data can be used together. Though previously we described running them together<sup>15</sup>, for this tool, they must be run independently and then compared.

1. Convert the .fcs files into the correct CLOCCS input format for flow cytometry by following the instructions in **Supplemental File 1** (also found in the CLOCCS\_alignment repo as CLOCCS/flow\_cytometry\_conversion\_instructions.txt).
2. Select the **Flow** selection from the **Model Type** dropdown menu.
3. Import the data using the **Data Import panel**. Click on **Select File**, and select the file generated in step 3.4.1.
4. Select the time points for which a flow cytometric CLOCCS fit should be plotted by selecting the time points in the **Times for Fitting** box.
5. Once all the inputs have been selected for either budding or flow cytometry, click on the **Apply** button, and then click on the **Sample** button at the top of the screen.

6. View the budding curve or flow cytometry plots with the predicted fits by selecting the **Predicted Fits** tab. This tab opens by default immediately after the previous step.
7. View the parameter histograms for each parameter by selecting the **Parameter Histograms** tab and then selecting the sub-tab that corresponds to the parameter of interest from the following options:  **$\mu_0$** ,  **$\delta$** ,  **$\sigma_0$** ,  **$\sigma_{\text{avg}}$** ,  **$\lambda$** ,  **$\text{bud.start}$** ,  **$\text{bud.end}$** , etc.
8. View the posterior score plot by selecting the **Posterior Score** tab.
9. View the settings, and further alter them by selecting the **Settings** tab; view the log of the previous runs by selecting the **Log** tab.
10. Obtain the CLOCCS parameters from the fit by selecting the **Posterior Parameters** tab. The resulting table will have the following form: each row consists of a parameter, with the final row being the posterior. The columns consist of the predicted parameter for the mean, the 2.5% lower confidence interval, the 97.5% upper confidence interval, and the acceptance rate.
  1. Record the parameters used for alignment for each experiment: the **recovery time from synchrony ( $\mu_0$ )** and the **average cell-cycle period of the mother cells ( $\lambda$ )**.
  2. Calculate the **cell-cycle period** by calculating the **average** of the **mother cell period ( $\lambda$ )** and the **daughter cell period ( $\lambda + \delta$ )**, where  **$\delta$**  is the **daughter-specific delay**.

**NOTE:** Repeat section 3 with all the experiments to be included in the comparisons.

#### 4. Conversion of time points to lifeline points using the Python conversion functions and the CLOCCS parameters

**NOTE:** Conversion between time points and lifeline points requires two conversion formulas<sup>21</sup>. A Python implementation for conversion and data visualization are available in the CLOCCS\_alignment repo and described below.

1. Activate the Conda environment by entering the following command into the terminal: `conda activate CLOCCS_alignment`
2. Open an interactive Python notebook by typing the following command into the terminal: `jupyter notebook`
3. Create a new Python notebook in the desired folder.
 

**NOTE:** An example notebook has been included to demonstrate standard use and can be found in `Alignment/JOVE_example.ipynb` in the CLOCCS\_alignment repo.
4. Import the Python file containing the alignment functions by running the following command in the first cell:
 

```
%run path_to_repo/cloccs_alignment/Alignment/
utilities.py
```

  1. Substitute the path to the CLOCCS\_alignment repo for `path_to_repo`.
5. If using budding data as the cell-cycle phase data, import a data frame containing the percent budded at each time point by running the following command in a new cell:
 

```
budding_df = pd.read_csv("path_to_folder/
budding_filename.tsv", sep = "\t", index_col=0)
```

  1. Substitute the appropriate file path and filename. If the file is a .csv file, remove `sep = "\t"`

6. If using budding data as the cell-cycle phase data, align the budding data to a lifeline point time scale by entering the following function into a new cell:

```
aligned_budding_df =  
df_conversion_from_parameters(budding_df,  
timepoints, param_mu0, param_lambda)
```

1. For timepoints, substitute a list of the time points to be the index of the budding\_df data frame.
2. For param\_mu0 and param\_lambda, substitute the learned parameters from the budding CLOCCS run in section 3 for the experiment.

7. If using flow cytometry data, import the flow cytometry data by running the following command in a new cell:

```
flow_samples =  
flow_cytometry_import(flow_input_folder)
```

1. For flow\_input\_folder, substitute the appropriate path to the folder containing the flow cytometry .fcs files.

8. If using flow cytometry data, generate a conversion table between the time points and lifeline points for each experiment by typing the following command into a new cell:

```
flow_converter = convert_tp_to_ll(timepoints,  
param_mu0, param_lambda)
```

1. For timepoints, substitute a list of the time points from the flow cytometry data.
  2. For param\_mu0 and param\_lambda, substitute the learned parameters from the flow cytometry CLOCCS run in section 3 for the experiment.
9. Import the data frame containing the experimental data into the notebook by running the following command in a new cell:

```
data_df = pd.read_csv("path_to_folder/  
exp_data_filename.tsv", sep = "\t", index_col=0)
```

1. Substitute the appropriate file path and filename. If the file is a .csv file, remove sep = "\t".

**NOTE:** This can be done for any tabular data. The experimental data must simply have the time points as either the columns or the index of the data frame. Example data can be found in the CLOCCS\_alignment repo.

10. Align the experimental data to a lifeline point time scale by entering the following function into a new cell:

```
lifeline_aligned_df =  
df_conversion_from_parameters(data_df, timepoints,  
param_mu0, param_lambda, interpolate, lowerll, upperll)
```

1. For timepoints, substitute a list of the time points as the index or the columns of the experimental data\_df from the previous step.
2. For param\_mu0 and param\_lambda, substitute the values obtained in section 3 from CLOCCS.

**NOTE:** The parameters can come from any CLOCCS run performed on any of the accepted cell-cycle phase data types.

3. Optionally, substitute interpolate with **True** or **False**, or leave blank (the default is **False**).

**NOTE:** When set to **False**, the data will not be interpolated. When set to **True**, the lifeline points will be rounded and interpolated to fill in the values between the lifeline points, such that there is a point per integer in the range of the lifeline points. This allows for better comparison across datasets.

4. Optionally, substitute **lowerll** and **upperll** with **None** or **integer values**.

**NOTE:** When set to **None**, all of the lifeline points after interpolation are kept. When **integers** are supplied, this truncates the data so that the lifeline points range from the **lowerll** to the **upperll**. This allows for comparison across datasets with a different lowerll or upperll.

11. Download the lifeline-aligned dataset by entering the following command into a new cell:  
`lifeline_aligned_df.to_csv("path_to_desired_location/name_of_file.tsv", sep = "\t")`
12. Repeat steps 4.5-4.11 with all the experiments to be included in the comparisons.

## 5. Comparing budding curves and flow cytometry data

1. Plot the budding curves prior to alignment using the Python utilities function by entering the following command into a new cell:  
`plot_budding_curves(list_of_budding_curves, list_for_legend = leg_list, point_type = str_type, title = str_title)`
  1. Substitute a list containing the data frames of all the desired budding curves for plotting for `list_of_budding_curves`-[`bud_df1`, `bud_df2`, `bud_df3`].
  2. Substitute a list of the labels for the legend-[**Experiment 1, Experiment 2, Mutant**] for `leg_list` if desired. If not, **exclude** or substitute **None**.
  3. Substitute **time** for `str_type`.
  4. Substitute a string title **Comparison Budding Curves** for `str_title` if desired. If not, substitute **None**, or **exclude**.

2. Plot the budding curves after alignment using the Python utilities function by following the instructions in step 5.1, but with a list of **aligned** budding curves substituted for `list_of_budding_curves` and with **lifeline** for `point_type` instead of **time**.
3. To plot the flow cytometry data, plot the associated data from the .fcs files at the corresponding lifeline points using the converter generated in step 4.8.
4. Convert the lifeline points to the cell-cycle phase by using the converter table (**Table 1**).  
**NOTE:** This can also be plotted by following the instructions in step 5.1, but with **phase** for `point_type` instead of **time**.

## 6. Comparing the experimental data

1. Determine the gene list to be plotted in the line graphs based on literature information or the genes of interest for the research.
2. Use the provided `plot_linegraph_comparison` in the Python utilities file to perform line graph comparisons on the original, aligned, or aligned and interpolated data frame by typing the following command into a new cell:  
`plot_linegraph_comparison(list_of_dfs, list_for_legend, genelist, point_type = str_type, title = str_title)`
  1. Substitute a list of the data frames of the experiments to be compared for `list_of_dfs`.  
**NOTE:** The data frames can be unaligned or aligned; however, the corresponding `point_type` must be input in step 6.2.4.
  2. Substitute a list of the titles for each data frame in the same order as the list of data frames for `list_for_legend`.

3. Substitute a list of the gene names (which must be included in the index of the data frames) to be plotted for `genelist`.
  4. Substitute the point type for `str_type`. Use **lifeline** (the default is lifeline point scale) or **phase** (the cell-cycle phase lifeline scale) for the aligned data frames in step 6.2.1 or **time** for the unaligned data frames in step 6.2.1.
  5. Substitute an optional string title for `str_title`.
3. Determine the gene list to be included in the heatmap using the literature or algorithms to determine the top periodic genes.
- NOTE:** For proper heatmap comparisons, the data should be aligned, interpolated, and timescale-adjusted in step 6.2; it should have the same starting and ending lifeline value for each experiment.
1. Run periodicity algorithms to determine the top periodic genes<sup>23,24</sup>, or use the desired alternative methods to determine the gene list (i.e., literature results).
  2. Import a .csv or .tsv gene list file into the notebook using the following command in a new cell:
 

```
sort_df = pd.read_csv("path_to_folder/sorting_filename.tsv", sep="\t", index_col=0)
```
  3. Substitute the appropriate file path and filename. If the file is a .csv file, remove `sep="\t"`.
  4. Use the provided function `plot_heatmap_comparison` in the Python utilities file to perform a heatmap comparison on the aligned, interpolated, and phase-aligned data frame by typing the following command into a new cell:
 

```
plot_heatmap_comparison(list_of_dfs, list_for_legend, genelist, title = str_title)
```

1. Substitute a list of the **aligned** data frames of the experiments to be compared for `list_of_dfs`.
2. Substitute a list of the titles for each data frame in the same order as the list of data frames for `list_for_legend`.
3. Substitute a list of the gene names (which must be included in the index of the data frames) to be plotted for `genelist`.
4. Substitute an optional string title for `str_title`.

**NOTE:** The first data frame in the list is the one that will be used for ordering the genes in the heatmap. The genes will be ordered by the maximum in the first period for that data frame, and the same order will be used for the subsequent data frames in the list.

## Representative Results

The steps described in the above protocol and in the workflow in **Figure 1** were applied to five cell-cycle synchronized time-series experiments to demonstrate two representative comparisons: between replicates with different synchrony methods (mating pheromone and centrifugal elutriation<sup>18</sup>) and sequencing platforms (RNA-sequencing [RNA-seq] and microarray), as well as across experimental conditions. Multiple experiments were performed with *S. cerevisiae*, and cell-cycle phase and experimental data were collected for each experiment. The workflow involves using CLOCCS to parameterize the various synchrony/release time-series experiments, using these parameters to align the experiments to a common comparable lifeline scale, and then using these aligned experiments for the two representative comparisons.

To demonstrate the representative comparison across replicates, we selected three experiments performed with the

same strain and in the same experimental conditions, called Condition 1. Two of these experiments were direct replicates of each other, and both were analyzed *via* microarray analysis and synchronized *via* centrifugal elutriation. The third experiment was analyzed using RNA-seq analysis and synchronized *via* alpha factor mating pheromone arrest. To demonstrate the second comparison across experiments with varying cell-cycle periods, the Condition 1 RNA-seq experiment (cell-cycle period: 71 min) from above was compared with Condition 2 (cell-cycle period: 82 min), and Condition 3 (cell-cycle period: 110 min) (**Table 2**). For each experiment, the cells were grown in their respective conditions, synchronized, released, and then sampled throughout two or more cell-cycle periods. The budding and/or flow cytometry data were collected to provide information on the cell-cycle phase, and either microarray or RNA-seq time-series transcriptomic data were collected as described in Leman et al.<sup>18</sup> (**Supplemental Table S1**).

For each experiment, the data took the forms described in **Figure 2**, which presents the Condition 2 experiment as an example for demonstration. Each dataset had a budding curve, which allowed for the inference of the cell-cycle phase. This curve comprised a budding percent value for each time point in the time series, which was then plotted to produce a budding curve displaying multiple cell-cycle oscillations (**Figure 2**). The cell-cycle phase data also took the form of flow-cytometric DNA content staining data for each time point in the time series. Select time points for Condition 2 were plotted (**Figure 2**). The flow cytometry files were combined into a single table comprising the cells in each log fluorescence bin for each time point for inputting into the CLOCCS using the `flow_cytometry_CLOCCS_file_from_fcs` function in the Python utilities. Each dataset also contained experimental data. In this case, the data were transcriptomic

data, and the data were organized into rows of genes, each with a value for the abundance of RNA at each time point in the experiment (**Figure 2**).

We have demonstrated the use of CLOCCS and the conversion to lifeline points for the Condition 2 RNA-seq dataset; however, the process was identical for the other experiments as well. The budding information was input into the CLOCCS algorithm as described in protocol section 3 and as shown in **Figure 3A**. The default values for **Sim Anneal**, **Burn In**, **Iterations**, and **Advanced Settings** were used. The appropriate experimental conditions were selected. The model type of "Bud" was used for the budding data. The resulting CLOCCS budding fits were viewed to ensure that the budding curves were properly fit, as demonstrated by the data points overlaying the corresponding fit curve with a small 95% confidence band (**Figure 3B** and **Supplemental Figure S1**). The parameters  $\mu_0$  and  $\lambda$  from the **posterior parameters table** (**Figure 3C**) were recorded for use in the alignment. The flow cytometry data for Condition 2 were separately input into CLOCCS, as described in protocol section 3. Currently, CLOCCS expects flow cytometers to produce 10 bit data with 1,024 channels; however, modern flow cytometers can have more channels. Since our flow cytometer produces data with more than 1,024 channels, the data were binned into 1,024 bins. With flow cytometry cell-cycle phase data, CLOCCS produces a CLOCCS fit for each selected time point (**Figure 3D** and **Supplemental Figure S2**) and supplies a **posterior parameters table** similar to the **budding posterior parameters table** in **Figure 3C**. The parameters for budding that CLOCCS runs for each of the other experiments are described in **Table 2**, and the parameters for the flow cytometry that CLOCCS runs are described in **Supplemental Table S2**.

The CLOCCS parameters corresponding to the cell-cycle period of the mother cells ( $\lambda$ ) and the recovery time ( $\mu_0$ ) were used for the lifeline alignment. It is important to note that  $\lambda$  does not necessarily represent the average cell-cycle period of the cell population. In cases where the cells undergo a full division, there are an equal number of mother and daughter cells, so the average cell-cycle period is the average between the cell-cycle period of the mother cells ( $\lambda$ ) and the cell-cycle period of the daughter cells ( $\lambda + \delta$ ); specifically, delta ( $\delta$ ) is the length of the daughter-specific delay. This is the calculation that we used for the cell-cycle period for each experiment (Table 2). For each experiment, the corresponding parameters  $\lambda$  and  $\mu_0$  were then used in the conversion function, `df_conversion_from_parameters`, supplied in the Python utilities file, as demonstrated for Condition 2 (Figure 4A). For the budding curves, the data were not interpolated. However, for experimental data, the lifeline-aligned datasets were resampled using interpolation such that each lifeline point contained interpolated data for improved plotting. To ensure that the lifeline-aligned datasets had the same range of lifeline points, lower and upper lifeline limits were set to truncate the data at those points. These `lowerll` and `upperll` parameters were input into the `df_conversion_from_parameters` function when the interpolation was set to **True**. For the Condition 1 comparison, they were set to 44 and 270, respectively, for all the datasets, and for the comparison across environmental conditions, they were set to 50 and 300, respectively. An example use of these functions for alignment and comparison can be seen in the example Python notebook `JOVE_example.ipynb`, and the code used for generating the figures can be seen in the `JOVE_Figures.ipynb` notebook in the `CLOCCS_alignment` repo.

This conversion from time points to lifeline points depends on two formulas<sup>21</sup> (Figure 4A) using  $\mu_0$  (recovery time) and  $\lambda$  (mother period). The first formula,  $\frac{TP}{\mu_0} \times 100$ , is the recovery phase formula (Figure 4A). This formula is used only for time points within the recovery phase, which consists of the time points up to and including  $\mu_0$ , since  $\mu_0$  corresponds to the recovery time. The time points are then converted to a lifeline scale range ending with 100 lifeline points (Table 1), marking the end of the recovery phase and the beginning of the first cell cycle. The post-recovery phase uses the second formula,  $\frac{TP - \mu_0}{\lambda} \times 100 + 100$  (Figure 4A), which converts each subsequent post-recovery time point into a lifeline point after 100. Each subsequent 100 lifeline points correspond to a new cell cycle, with the first cycle corresponding to lifeline points 100 to 200, the second cycle corresponding to lifeline points 200 to 300, and so on (Table 1). The conversion from time points to lifeline points is applied to each dataset individually using the corresponding CLOCCS parameters for that dataset. After each dataset is converted to the lifeline scale, the cell-cycle phases are aligned, which allows for phase-specific comparisons across datasets.

Table 3 shows the conversion of select time points into their respective lifeline points for the representative conversion of the Condition 2 dataset using parameters from the budding CLOCCS run. The budding data collected from the Condition 2 RNA-seq were plotted in a budding curve showing the percent budded over time for both the unaligned time scale in minutes (Figure 4B) and the aligned timescale in lifeline points (Figure 4C) using the Python function `plot_budding_curves` in a Python notebook. The lifeline points could be easily converted into experimental and cell-cycle phase information (Table 1), and the recovery phase and first to third cell cycles were color-coded by hand accordingly

(**Figure 4B,C**). Since each lifeline point corresponded to a cell-cycle phase, individual flow cytometry plots could be labeled *via* the Python functions using the cell-cycle phase determined by the lifeline alignment. These phases matched with the phases determined *via* flow-cytometric analysis for Condition 2. The flow cytometry data collected for the Condition 2 dataset were plotted for select time points and labeled using the cell-cycle phase determined from the flow cytometry lifeline alignment. In each case, the data matched the phase determined by the alignment (**Figure 4D**).

It is important to note that the expression level of each gene for each sample remains the same, but the labeling of the time points is altered from time in minutes to lifeline points. However, the conversion is not linear. The recovery phase, highlighted in gray, occupies a higher percentage of the experimental time once the conversion to lifeline points has been performed (**Figure 4B,C**). The advantage of the lifeline scale is that it allows for detailed phase information and phase comparisons across experiments. The phase information is contained in the lifeline points, as described above and displayed in **Table 1**. Furthermore, G1 is contained in the first 15.5 lifeline points of each cell cycle, S in the next 20 lifeline points, and G2/M in the next 64.5 lifeline points (**Table 1**). However, this artificially constrains the recovery time to the same time span of each consecutive cell cycle, even if the recovery phase appears very short in the original time point scale. This does not obscure the comparisons, because the phases of each experiment are aligned. In most cases, it is more relevant to compare the data at points that occur at the same experimental and biological phase rather than at time points that occur at the same time in minutes.

Once all the experiments have been converted to the aligned lifeline scale using the provided Python functions

in the Python utilities file, they can be compared. Here, we demonstrate two common comparisons between experiments: one between replicates of a similar experiment across platforms and synchronization methods (**Figure 5**) and one between different experimental conditions with a changing period length (**Figure 6** and **Figure 7**). As described above, the first comparison is across two elutriated microarray replicates and one alpha factor synchronized RNA-seq experiment. Before alignment, the two microarray replicates showed similar synchrony and cell-cycle dynamics, but the Condition 1 Microarray 2 replicate appeared slightly delayed (**Figure 5A**). The most striking difference was found when comparing the unaligned datasets; the Condition 1 RNA-seq second cycle appeared aligned with the first cycle of the two microarray experiments. The difference was likely not related to the different transcriptomic platforms but rather the different synchronization methods. The cell populations in the microarray experiments were synchronized by centrifugal elutriation, while the population for the RNA-seq experiment was synchronized by a mating pheromone treatment. Indeed, synchronization with mating pheromone substantially reduced the recovery time compared to elutriation (**Figure 5A** and **Table 2**).

Despite the obvious differences between replicates when plotted in terms of the elapsed time, after the lifeline alignment, the curves were almost identical, and more detailed and relevant comparisons across replicates were made possible (**Figure 5B**). The recovery phase was aligned so that each experiment began at the same lifeline point, and the variations in period were normalized by lifeline alignment. Due to the alignment, experimental values at the same lifeline point across replicates occurred in the same cell-cycle phase, thus enabling calculations of the experimental variance across replicates. The recovery and

cell-cycle phases are labeled in **Figure 5B** to provide additional information about cell-cycle phases in each of the experiments. This lifeline alignment could then be applied to the experimental dataset (**Figure 5C,D**) using the Python function `df_conversion_from_parameters` provided in the utilities file, as described above.

In **Figure 5D**, the transcriptomic data were aligned, and the expression dynamics for the *CDC20* gene were plotted using the `plot_linegraph_comparison` Python function in a Python notebook. Before alignment, it appeared as if the first peak expression of the microarray experiments aligned with the second peak of the RNA-seq experiment (**Figure 5C**); however, after alignment, the first cell-cycle peaks of each dataset aligned properly (**Figure 5D**). Furthermore, the peak width of the experiments appeared to differ between the RNA-seq dataset and the microarray datasets, but after alignment, the peak width was more aligned (**Figure 5C,D**).

The second comparison is between experiments in different environmental conditions with different cell-cycle periods (**Figure 6**). As described above, here, we compared *S. cerevisiae* datasets in Condition 1 to Condition 2 and Condition 3, which correspond to cell-cycle periods of 71, 82, and 110 min, respectively. These differences in the cell-cycle period introduced uncertainty when comparing across experiments prior to cell-cycle phase alignment, as shown in the unaligned budding curves. The period differences are visible in the unaligned budding curves (**Figure 6A**). However, when they were CLOCCS aligned using this protocol, the three curves looked remarkably similar, thus making comparisons of experimental data possible (**Figure 6B**).

Using the flow cytometry CLOCCS parameters, Condition 1 and Condition 2 were aligned to a common lifeline scale,

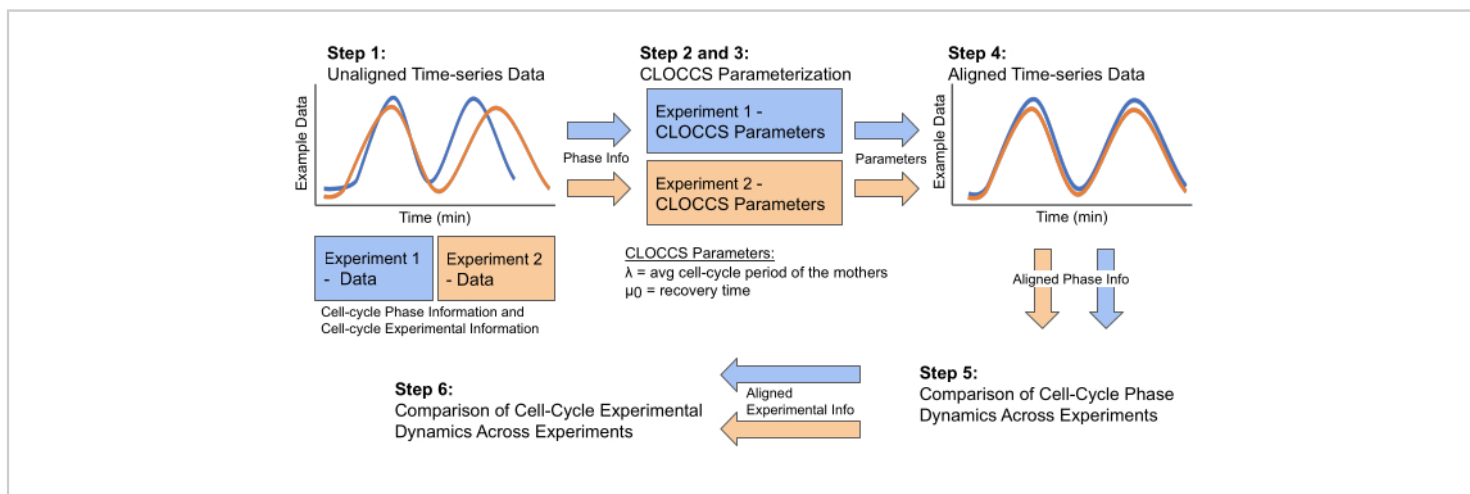
and DNA content histograms were plotted in Condition 2 and at equivalent lifeline points in Condition 1. Flow cytometric measurements of the DNA content across lifeline points were compared (**Figure 6C**). As the DNA content measurements were not continuous and not easily interpolated, we could only compare the nearest lifeline points. The cell-cycle phase data for each comparable lifeline point was not identical between the two conditions (**Figure 6C**), which indicates that the CLOCCS fits and resulting parameters were likely slightly misaligned for Condition 1. This was likely due to the poorer CLOCCS fit to the flow cytometric data for Condition 1 compared to Condition 2 (**Supplemental Figure 2**). However, the alignment only deviated in one sample and, thus, still allows for improved phase-specific comparisons.

The budding lifeline alignment was then applied to the experimental data for the RNA-seq experiments in Condition 1, Condition 2, and Condition 3 (**Figure 7**) by using the budding CLOCCS parameters in the `df_conversion_from_parameters` function on the experimental data. The transcriptomic data were aligned, and the gene expression of the gene *CDC20* for each time series was shown for the three experiments. Prior to alignment, the transcript dynamics of *CDC20* were non-overlapping (**Figure 7A**). After the alignment, the first and second peaks of the *CDC20* gene expression were much more closely aligned for all three datasets. After alignment, it became clear that the peaks occurred in the same cell-cycle phase, but the shapes of the curves were different (**Figure 7B**). Condition 3 had a lower and broader first peak compared to the other two conditions, even after accounting for the differences in the cell-cycle period, suggesting that these differences were likely related to the experimental conditions being tested (**Figure 7B**).

Large-scale transcriptomic comparisons could also be made. For these comparisons, 278 genes were selected by running the periodicity algorithm JTK\_CYCLE<sup>23</sup> on each dataset and taking the intersection of the top periodic genes. However, genes can be selected using any desired method or from the literature. These genes were plotted in the same order for all three conditions both for the unaligned (**Figure 7C**) and the aligned (**Figure 7D**) heatmaps using the `plot_heatmap_comparison` Python function in a Python notebook. These heatmaps allow for hundreds of gene-level comparisons to be made simultaneously. Comparisons across unaligned experiments could be made regarding the change in curve dynamics, the peak time relative to neighboring genes, and the period length, etc. (**Figure 7C**). However, detailed phase-specific comparisons could not be made because the time points do not necessarily correlate to the same cell-cycle phase across conditions. Although the second cycles appeared similar after alignment, the first cycles were slightly shifted between the conditions (**Figure 7D**). This shift may reflect the fact that the budding cell-cycle phase information was of lower quality for Condition

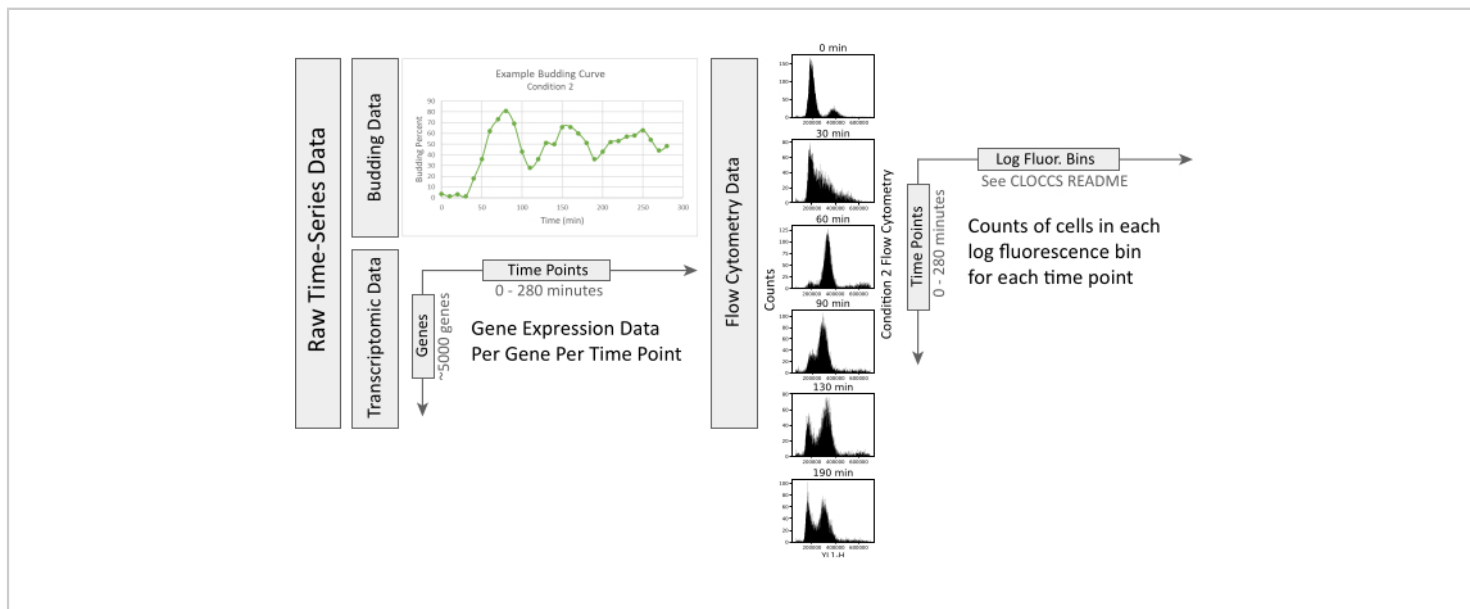
3. Nonetheless, the alignment of the experiments for the three conditions allowed for an improved phase-specific comparison. Prior to alignment, it was unclear whether the first peak of expression in each condition would occur at the same cell-cycle phase (**Figure 7C**); however, after alignment, the experiments could be compared in a phase-specific manner (**Figure 7D**). Prior to alignment, the peaks in Condition 3 appeared much broader than in the other two conditions (**Figure 7C**); however, after alignment, it became clear that the peaks in Condition 3 were of similar width to the other conditions when aligned (**Figure 7D**).

These representative results demonstrate the process for the use of CLOCCS to align experiments to a common time scale. Prior to alignment, direct time point comparisons often do not correlate to a similar cell-cycle phase. The conversion of the elapsed experimental time in minutes to lifeline points that represent the cell-cycle phase allows for phase-specific and biologically relevant comparisons between experiments at the same point in the cell cycle.

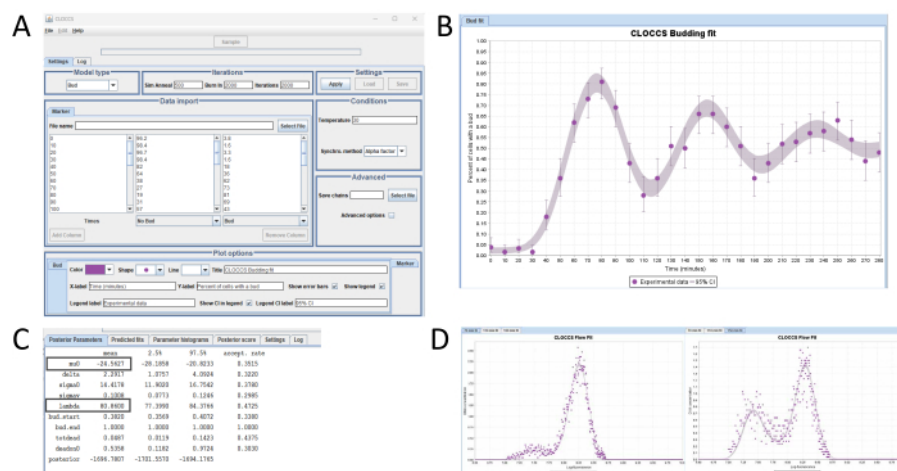


**Figure 1: CLOCCS lifeline alignment workflow overview.** The experimental workflow for the alignment of two example datasets using CLOCCS, followed by representative comparisons between the datasets. The major steps from the protocol

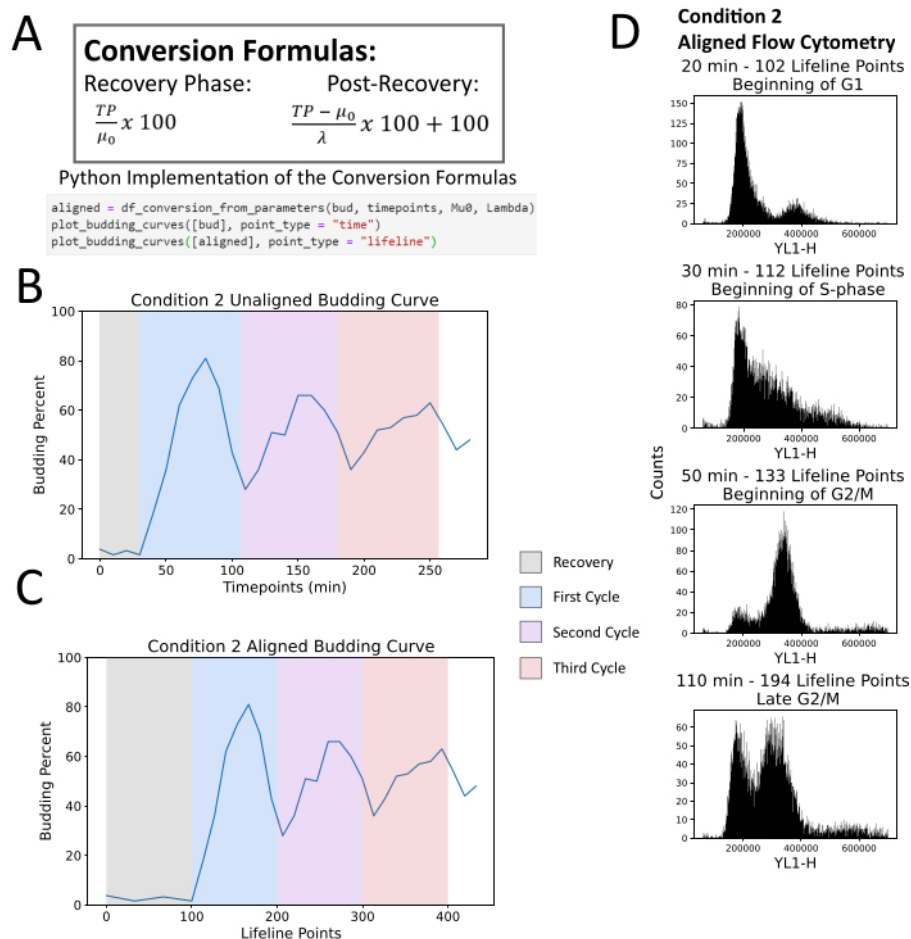
are illustrated: the collection of unaligned cell-cycle phase and experimental data for each of the datasets (step 1), the use of CLOCCS for the parameterization of each dataset (step 2 and step 3), the alignment of the datasets to a common lifeline (step 4), and finally, the comparison of the cell-cycle phase and experimental dynamics (step 5 and step 6). The unaligned cell-cycle phase data are input into CLOCCS to provide learned parameters, which are then used for alignment to a common lifeline scale. These aligned datasets are then compared. Abbreviation: CLOCCS = Characterizing Loss of Cell Cycle Synchrony. [Please click here to view a larger version of this figure.](#)



**Figure 2: Format of the cell-cycle phase and experimental data required for the workflow.** The data required for the workflow consist of two main components: cell-cycle phase data and cell-cycle experimental data. The cell-cycle phase data can consist of cell-cycle budding data or flow-cytometric DNA content data for each time point in the time series. The experimental data can take many forms, but in this case, are transcriptomic data, which consist of gene expression data for each gene for every time point in the time series. [Please click here to view a larger version of this figure.](#)

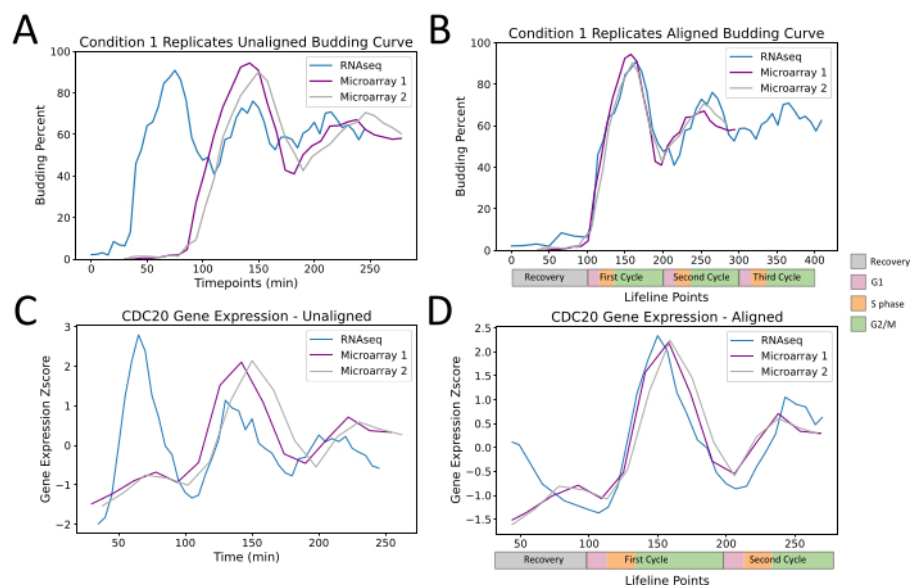


**Figure 3: Example of results from running CLOCCS on an *S. cerevisiae* cell-cycle dataset. (A)** A screenshot of the CLOCCS graphical user interface with the input values and settings supplied for Condition 2 budding data. The times, the number of unbudded cells, and the number of budded cells are input, as well as the model type, iterations, and conditions, etc. **(B)** A screenshot of the resulting CLOCCS budding fit for Condition 2 under the "Predicted Fit" tab of the results. Each datapoint has an associated sampling error bar corresponding to the 95% binomial proportion confidence intervals of the data (for each time point, at least 200 cells were counted [between 204 and 295 cells]). The resulting budding fit curve shows the confidence band for the 95% confidence interval of the CLOCCS fit in purple. **(C)** A screenshot of the resulting "Posterior Parameters" table for the Condition 2 budding CLOCCS run consisting of the CLOCCS parameters at the mean, the 2.5% confidence interval, and the 97.5% confidence interval. The posterior and acceptance rates are also shown. **(D)** A screenshot of the flow cytometry CLOCCS fits for Condition 2 at 70 min and 150 min. [Please click here to view a larger version of this figure.](#)



**Figure 4: Example of the conversion process from time points to aligned lifeline points for the Condition 2 dataset.**

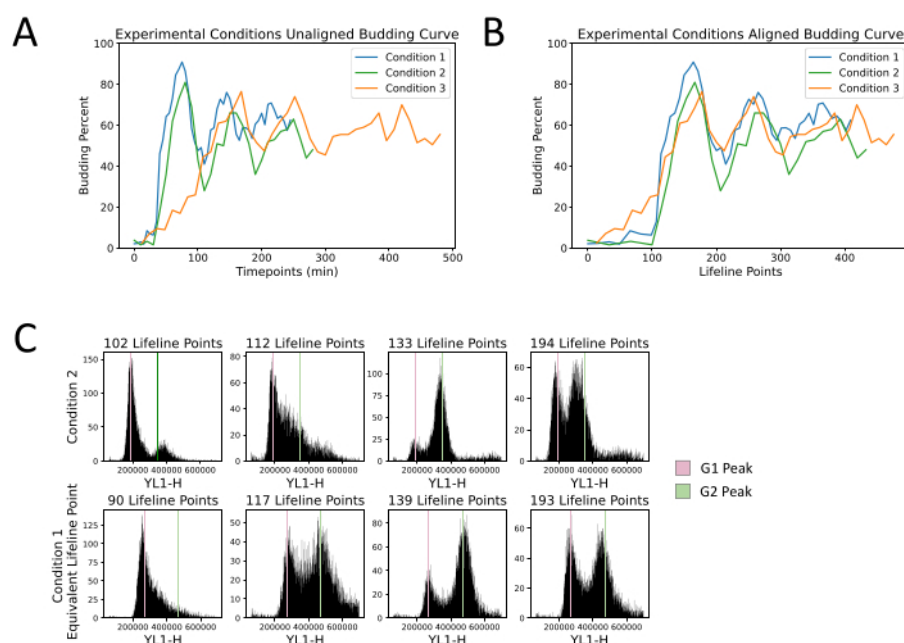
(A) The conversion formulas used to convert from time points to lifeline points. A screenshot of the Python functions in the Python notebook for conversion and plotting the budding curves. (B) The unaligned Condition 2 budding curve showing the budding percent for each time point in minutes. The cell-cycle and recovery phases are highlighted as follows: recovery (gray), first cell cycle (blue), second cell cycle (magenta), and third cell cycle (salmon). (C) The aligned Condition 2 budding curve showing the same budding percentages but plotted on the lifeline-aligned scale. The cell-cycle and recovery phases are highlighted as in panel C. (D) The aligned flow cytometry plots for select time points from Condition 2 corresponding to distinct cell-cycle phases based on the lifeline scale: the beginning of G1, the beginning of S-phase, the beginning of G2/M, and late G2/M. [Please click here to view a larger version of this figure.](#)



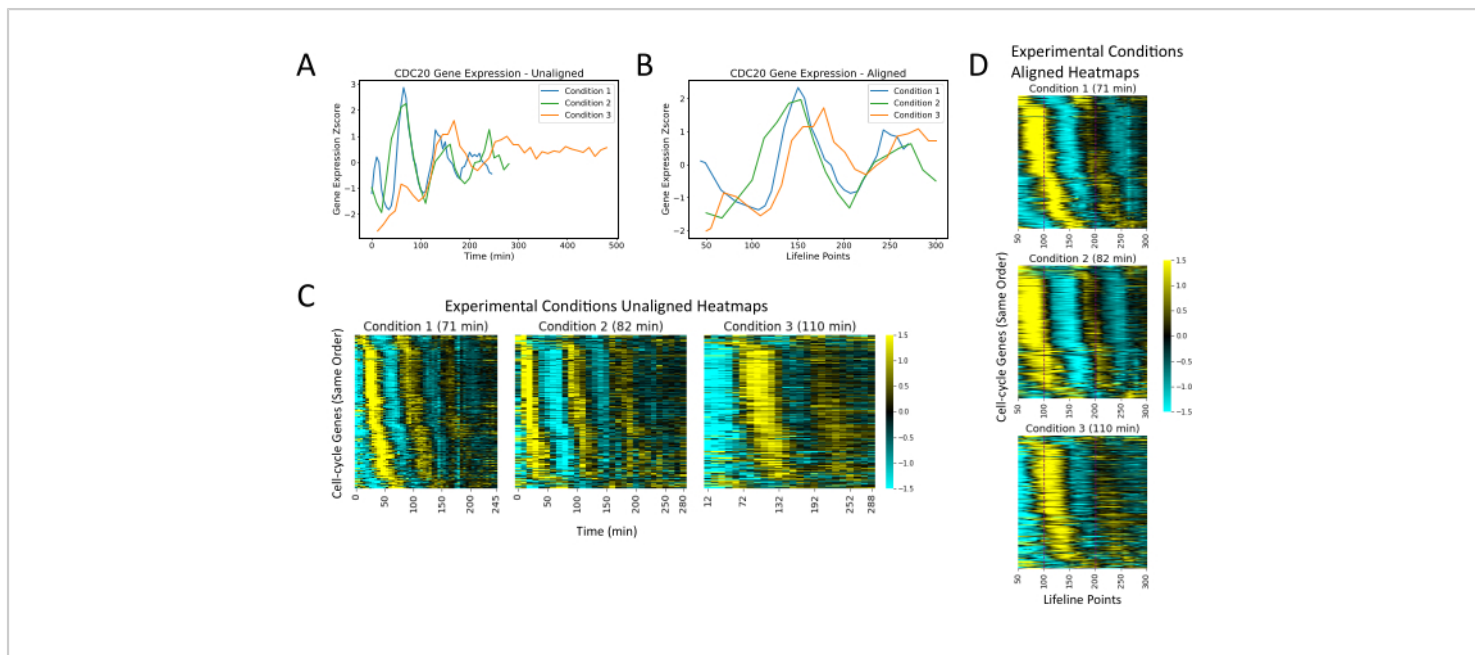
**Figure 5: Representative results for the comparison of the aligned and unaligned Condition 1 replicate experiments.**

Comparison of the Condition 1 replicates: Condition 1 RNA-seq (blue), Condition 1 microarray 1 (purple), and Condition 1 microarray 2 (gray). **(A)** The unaligned budding curve for the Condition 1 datasets. **(B)** The aligned budding curve for the Condition 1 datasets. The lifeline points have been converted to the cell-cycle phase and are color-coded below the x-axis.

**(C)** The unaligned gene expression of a representative gene, *CDC20*, for the Condition 1 datasets. **(D)** The aligned gene expression of a representative gene, *CDC20*, for the Condition 1 datasets. [Please click here to view a larger version of this figure.](#)



**Figure 6: Representative results for the comparison of aligned and unaligned cell-cycle phase data across experiments with varying periods.** Comparison of the cell-cycle phase data for datasets with three different environmental conditions and, thus, three different cell-cycle periods: Condition 1 RNA-seq (cell-cycle period: 71 min), Condition 2 RNA-seq (cell-cycle period: 82 min), and Condition 3 RNA-seq (cell-cycle period: 110 min). **(A)** The unaligned budding curve for the datasets. **(B)** The aligned budding curve for the datasets. **(C)** The flow-cytometric DNA content histograms for Condition 2 (top row) compared to the equivalent lifeline points in Condition 1 (bottom row). [Please click here to view a larger version of this figure.](#)



**Figure 7: Representative results for the comparison of the aligned and unaligned transcriptomic data across experiments with varying periods.** Comparison of the transcriptomic data associated with the datasets in Figure 6: Condition 1 RNA-seq, Condition 2, and Condition 3. **(A)** The unaligned gene expression of a representative gene, *CDC20*, for the Condition 1, Condition 2, and Condition 3 RNA-seq datasets. **(B)** The aligned gene expression of *CDC20* for the datasets. **(C)** The unaligned heatmap of the top cell-cycle periodic genes in the same order for each dataset. **(D)** The lifeline-aligned heatmaps of the same cell-cycle periodic genes from panel C in the same order. The dashed purple lines correspond to the lifeline points 100 and 200. [Please click here to view a larger version of this figure.](#)

**Table 1: Lifeline point to cell-cycle phase conversion.**

The conversion key between the lifeline point scale and the corresponding phase in the experiment. Lifeline points 0-100 correspond to recovery from synchrony. Each subsequent 100 lifeline points correspond to a new cell cycle, with the first 15.5 lifeline points corresponding to G1, the next 20 corresponding to S-phase, and the remaining lifeline points corresponding to G2/M. [Please click here to download this Table.](#)

**Table 2: Budding CLOCCS parameters.** The resulting budding CLOCCS parameters "lambda" and "mu0" for each experiment from the representative results. Additionally, the

daughter-specific delay "Delta" and the calculated cell-cycle period are shown for each experiment. [Please click here to download this Table.](#)

**Table 3: Conversion table showing the conversion between time points in minutes and their respective corresponding lifeline points for Condition 2.** [Please click here to download this Table.](#)

**Supplemental Figure S1: CLOCCS budding fits for Condition 1 and Condition 3.** Screenshot of the resulting CLOCCS budding fit for **(A)** the Condition 1 RNA seq budding data, **(B)** the Condition 1 microarray 1 budding data, **(C)**

the Condition 1 microarray 2 budding data, and for (D) the Condition 3 budding data. The CLOCCS budding fit for Condition 2 can be seen in **Figure 3B**. The 95% confidence band and the sampling error bars are as described in the CLOCCS documentation<sup>14,15</sup> and in **Figure 3**. For each time point for each time series, approximately 200 cells were counted. [Please click here to download this File.](#)

**Supplemental Figure S2: CLOCCS flow cytometry fits for Condition 1 and Condition 2.** Screenshot of the flow cytometry CLOCCS fits for the samples shown in **Figure 6C** for Condition 2 (top row: **A-D**) and Condition 1 (bottom row: **E,F**). [Please click here to download this File.](#)

**Supplemental Figure S3: Sensitivity of the alignment to variations in the CLOCCS parameters.** Comparison of the alignment of the Condition 1 RNA-Seq dataset using (**A-C**) variations in the CLOCCS parameters  $\lambda$  and  $\mu_0$  within the confidence interval of the CLOCCS fit and (**D,E**) with large variations in the parameters. Comparison between the mean value with the 2.5% and 97.5% confidence values output in the parameter table by CLOCCS for (**A**) the parameter  $\mu_0$ , (**B**) the parameter  $\lambda$ , and (**C**) for both parameters  $\mu_0$  and  $\lambda$ . (**D**) Comparison between the alignment using the mean value for  $\mu_0$  compared to large variations in the  $\mu_0$  parameter (200% to 0.25% of  $\mu_0$ ). (**E**) Comparison between the alignment using the mean value for  $\lambda$  compared to large variations in the  $\lambda$  parameter (200% to 0.25% of  $\lambda$ ). [Please click here to download this File.](#)

**Supplemental Table S1: Description of the data collection for each experiment.** For each experiment, this table provides a description of the budding data, flow cytometry data, transcriptomic data, and synchronization method. [Please click here to download this File.](#)

**Supplemental Table S2: CLOCCS parameters from the flow-cytometric CLOCCS runs.** The CLOCCS parameters " $\mu_0$ " and " $\lambda$ " for the Condition 1 and Condition 2 flow cytometry CLOCCS runs. [Please click here to download this File.](#)

**Supplemental File 1: Instructions for the conversion of the flow-cytometric data into CLOCCS input format.** For the use of CLOCCS with flow-cytometric data, a specific input format is required. This file provides more detailed instructions regarding protocol step 3.4.1 to explain how to use the Python utility functions to perform this conversion. [Please click here to download this File.](#)

## Discussion

This paper presents a method for more accurately and quantitatively assessing data from time-series experiments on synchronized populations of cells. The method utilizes learned parameters from CLOCCS, a Bayesian inference model that uses input cell-cycle phase data, such as budding data and flow-cytometric DNA content data, to parameterize each experiment<sup>14,15</sup>. CLOCCS uses the input cell-cycle phase data to infer the parameters for each experiment, which are then used for alignment to a common lifeline scale. Converting multiple synchrony/release time-series experiments to a single lifeline-aligned time scale allows for phase-specific and relevant comparisons between experiments and the aggregation of multiple replicate experiments, which were previously difficult or impossible.

The critical steps of this protocol include gathering the data, running CLOCCS, aligning the datasets, and comparing across the datasets. First, data must be gathered for use in this protocol. The data must consist of both experimental data-containing information regarding the question of interest (i.e., transcriptomic data, gene-expression data, proteomic

data)-and cell-cycle phase data-containing information on the phase of the cell cycle (i.e., budding data, flow-cytometric DNA content data). Then, the cell-cycle phase data can be used in CLOCCS to gather the parameter information for each experiment. The parameters  $\mu_0$  (recovery phase length) and  $\lambda$  (mother cell-cycle period) are used to convert the time points into lifeline points. The lifeline point alignment allows for the aligned time series to be directly compared.

One limitation of the method is that proper alignment is dependent on identifying a good fit to the data. Achieving the best CLOCCS fit relies on the quality of the cell-cycle phase data and the use of the correct input settings for the experiment in CLOCCS. The fit to the cell-cycle phase data determines the accuracy of the learned parameters and, thus, greatly impacts the accuracy of the alignment, because it depends on the use of these parameters. As broad changes in the parameters would greatly affect the alignment, the changes remain minimal within the confidence interval supplied in the CLOCCS output (**Supplemental Figure S3**). It is important to note that this sensitivity to variations in the parameters is also what allows for alignment between datasets with varying cell-cycle timing.

The accuracy of the CLOCCS fit can be determined using the resulting CLOCCS fit curve and the corresponding error bars and error band (**Figure 3B,D**, **Supplemental Figure S1**, and **Supplemental Figure S2**). The CLOCCS fit tab shows the original data points, as well as the CLOCCS fit curve with the confidence band corresponding to the confidence interval of the CLOCCS fit and the error bars corresponding to the 95% binomial proportion confidence interval of the data, since the counts are assumed to be independent binomial random variables<sup>14</sup>. For example, the confidence bars on the budding

data measure the confidence in the proportion of budded cells for a given sample.

One method for determining the quality of the CLOCCS fit involves determining whether the error bars of the data overlap with the confidence interval band of the CLOCCS fit. Another indication is the broadness of the 95% confidence band of the CLOCCS fit. In general, the width of the band decreases with increased goodness of fit. An indication of poor alignment is if the cell-cycle phase of the original data does not match with the cell-cycle phase inferred from the alignment. Each alignment can be double-checked by confirming that for, each time point, the phase indicated by the cell-cycle phase information data matches with the cell-cycle phase assigned by the alignment.

A poor CLOCCS fits or poor alignment could be the result of low-quality cell-cycle phase data. High-quality budding data will have a very low budding percentage immediately after arrest and a very high budding percentage at the first peak. The subsequent peaks and troughs will lose synchrony but should be distinct and evenly spaced. Since the lifeline points represent the average cell-cycle phase of the population, poor synchronization can impede proper alignment as well. High-quality flow-cytometric DNA content data will have distinct 1C and 2C peaks for each time point corresponding to the appropriate cell-cycle phase. Additionally, insufficient cell-cycle phase data introduces parameter identifiability problems. In the case of sufficient data, the parameters can be inferred and do not change substantially between CLOCCS runs. However, the parameters described in this protocol ( $\lambda$ ,  $\delta$ ,  $\mu_0$ ) cannot be disentangled when the cell-cycle phase data contain only one full cell cycle. To allow for improved parameter estimation, sufficient and well-constructed cell-cycle data should be used for the

CLOCCS fits<sup>14,15</sup>. Furthermore, the CLOCCS model uses prior information as described in Orlando et al.<sup>15</sup>, but this information can be adjusted to better suit the experimental conditions used.

If the quality of the cell-cycle phase data is good, then re-adjusting the CLOCCS settings may help produce a more accurate fit. For example, the number of iterations selected could be increased to improve accuracy. Confirming that the correct synchronization method was selected in CLOCCS can also be useful, since alpha factor arrest is associated with a shorter recovery time compared to elutriation.

This method is also limited in terms of the types of cell-cycle phase data currently supported. However, CLOCCS is flexible and can be adapted to support other types of data. For example, CLOCCS has previously been adapted to support the cell-cycle fluorescent labeling of spindle pole bodies, myosin rings, and nuclei<sup>11</sup> for use as cell-cycle phase identifiers. Furthermore, the use of CLOCCS with species other than *S. cerevisiae* has been made possible. CLOCCS accepts septation indices as a marker for the cell-cycle phase in *S. pombe*<sup>14</sup>, as well as flow-cytometric DNA content data, which are easily collectable for many species<sup>15</sup>. This allows for the comparison of experimental data at the same phase of the cell cycle for two completely different species and can give insights into changes in the cell cycle across evolution.

Though only supported forms of cell-cycle phase data can be used with this lifeline alignment method, this method is agnostic to the type of time-series experimental data used. In this protocol, we have demonstrated its use in aligning the gene expression of an individual gene, as well as time-series transcriptomic data for hundreds of genes in tandem. We have shown that this method can be used to compare across platforms and, thus, make comparisons

between RNA-seq datasets and microarray datasets taken in similar conditions. We have also shown that this method can be used to align datasets with different synchronization methods by comparing between a dataset that was elutriated (Condition 1 Microarray) with a dataset that was alpha factor arrested (Condition 1 RNA-seq). Previously, CLOCCS has also been used to align time-series transcriptomic and time-series proteomic data using budding cell-cycle phase data<sup>22</sup>, which allowed for direct comparisons between the mRNA dynamics and the dynamics of the corresponding protein. CLOCCS has also been used to align time-series data across species, such as for alignment between *S. cerevisiae* and *S. pombe*<sup>14</sup> and between the first cycle of *S. cerevisiae* and the pathogenic yeast *Cryptococcus neoformans*<sup>21</sup>. Finally, CLOCCS alignment is currently specific for cell-cycle time-series data and has not yet been adapted for use with other types of rhythmic processes. One area where this would be of particular interest is for circadian rhythms, where circadian time (CT) is conventionally used to align experiments, though its implementation is not consistently applied. Another area of interest is for investigating developmental rhythms, such as those of the malaria parasite. For example, the alignment of *Plasmodium falciparum* strains with different periods, as described in Smith et al.<sup>25</sup>, would allow for more detailed comparisons across strains. The alignment of these periodic processes for comparison would allow for a better understanding of these important rhythmic biological functions. These types of cell-cycle comparisons have been made possible by using CLOCCS for lifeline alignment, as described in this protocol.

## Disclosures

The authors have no conflicts of interest to disclose.

## Acknowledgments

S. Campione and S. Haase were supported by funding from the National Science Foundation (DMS-1839288) and the National Institutes of Health (5R01GM126555). Additionally, the authors would like to thank Huarui Zhou (Duke University) for comments on the manuscript and for beta testing the protocol. We also thank Francis Motta (Florida Atlantic University) and Joshua Robinson for their help with the Java code.

## References

1. Tyers, M., Tokiwa, G., Futcher, B. Comparison of the *Saccharomyces cerevisiae* G1 cyclins: Cln3 may be an upstream activator of Cln1, Cln2 and other cyclins. *EMBO Journal*. **12** (5), 1955-1968 (1993).
2. Schwob, E., Nasmyth, K. CLB5 and CLB6, a new pair of B cyclins involved in DNA replication in *Saccharomyces cerevisiae*. *Genes and Development*. **7** (7A), 1160-1175 (1993).
3. Polymenis, M., Schmidt, E. V. Coupling of cell division to cell growth by translational control of the G1 cyclin CLN3 in yeast. *Genes and Development*. **11** (19), 2522-2531 (1997).
4. Spellman, P. T. et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*. **9** (12), 3273-3297 (1998).
5. Cho, R. J. et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*. **2** (1), 65-73 (1998).
6. Bar-Joseph, Z. Analyzing time series gene expression data. *Bioinformatics*. **20** (16), 2493-2503 (2004).
7. Pramila, T., Wu, W., Miles, S., Noble, W. S., Breeden, L. L. The Forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S-phase gap in the transcriptional circuitry of the cell cycle. *Genes and Development*. **20** (16), 2266-2278 (2006).
8. Orlando, D. A. et al. Global control of cell-cycle transcription by coupled CDK and network oscillators. *Nature*. **453** (7197), 944-947 (2008).
9. Nash, R., Tokiwa, G., Anand, S., Erickson, K., Futcher, A. B. The WHI1+ gene of *Saccharomyces cerevisiae* tethers cell division to cell size and is a cyclin homolog. *EMBO Journal*. **7** (13), 4335-4346 (1988).
10. Basco, R. D., Segal, M. D., Reed, S. I. Negative regulation of G1 and G2 by S-phase cyclins of *Saccharomyces cerevisiae*. *Molecular and Cellular Biology*. **15** (9), 5030-5042 (1995).
11. Mayhew, M. B., Robinson, J. W., Jung, B., Haase, S. B., Hartemink, A. J. A generalized model for multi-marker analysis of cell cycle progression in synchrony experiments. *Bioinformatics*. **27** (13), i295-i303 (2011).
12. Qu, Y. et al. Cell cycle inhibitor Whi5 records environmental information to coordinate growth and division in yeast. *Cell Reports*. **29** (4), 987-994.e5 (2019).
13. Di Talia, S., Skotheim, J. M., Bean, J. M., Siggia, E. D., Cross, F. R. The effects of molecular noise and size control on variability in the budding yeast cell cycle. *Nature*. **448** (7156), 947-951 (2007).
14. Orlando, D. A. et al. A probabilistic model for cell cycle distributions in synchrony experiments. *Cell Cycle*. **6** (4), 478-488 (2007).
15. Orlando, D. A., Iversen, E. S., Jr., Hartemink, A. J., Haase, S. B. A branching process model for flow

- p>cytometry and budding index measurements in cell synchrony experiments.
- Annals of Applied Statistics*
- .
- 3**
- (4), 1521-1541 (2009).
16. Duan, F., Zhang, H. Correcting the loss of cell-cycle synchrony in clustering analysis of microarray data using weights. *Bioinformatics*. **20** (11), 1766-1771 (2004).
17. Darzynkiewicz, Z., Halicka, H. D., Zhao, H. Cell synchronization by inhibitors of DNA replication induces replication stress and DNA damage response: analysis by flow cytometry. *Methods in Molecular Biology*. **761**, 85-96 (2011).
18. Leman, A. R., Bristow, S. L., Haase, S. B. Analyzing transcription dynamics during the budding yeast cell cycle. *Methods in Molecular Biology*. **1170**, 295-312 (2014).
19. Rosebrock, A. P. Synchronization and arrest of the budding yeast cell cycle using chemical and genetic methods. *Cold Spring Harbor Protocols*. **2017** (1) (2017).
20. Haase, S. B., Reed, S. I. Improved flow cytometric analysis of the budding yeast cell cycle. *Cell Cycle*. **1** (2), 132-136 (2002).
21. Kelliher, C. M., Leman, A. R., Sierra, C. S., Haase, S. B. Investigating conservation of the cell-cycle-regulated transcriptional program in the fungal pathogen, *Cryptococcus neoformans*. *PLoS Genetics*. **12** (12), e1006453 (2016).
22. Kelliher, C. M. et al. Layers of regulation of cell-cycle gene expression in the budding yeast *Saccharomyces cerevisiae*. *Molecular Biology of the Cell*. **29** (22), 2644-2655 (2018).
23. Hughes, M. E., Hogenesch, J. B., Kornacker, K. JTK\_CYCLE: An efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets. *Journal of Biological Rhythms*. **25** (5), 372-380 (2010).
24. Deckard, A., Anafi, R. C., Hogenesch, J. B., Haase, S. B., Harer, J. Design and analysis of large-scale biological rhythm studies: A comparison of algorithms for detecting periodic signals in biological data. *Bioinformatics*. **29** (24), 3174-3180 (2013).
25. Smith, L. M. et al. An intrinsic oscillator drives the blood stage cycle of the malaria parasite *Plasmodium falciparum*. *Science*. **368** (6492), 754-759 (2020).