



Microfluidics for single-cell lineage tracking over time to characterize transmission of phenotypes in *Saccharomyces cerevisiae*

Poonam Bheda, Diana Aguilar-Gomez, Igor Kukhtevich, Johannes Becker, Gilles Charvin, Antonis Kirmizis, Robert Schneider

► To cite this version:

Poonam Bheda, Diana Aguilar-Gomez, Igor Kukhtevich, Johannes Becker, Gilles Charvin, et al.. Microfluidics for single-cell lineage tracking over time to characterize transmission of phenotypes in *Saccharomyces cerevisiae*. STAR Protocols, 2020, 1 (3), pp.100228. 10.1016/j.xpro.2020.100228 . hal-03666939

HAL Id: hal-03666939

<https://hal.science/hal-03666939>

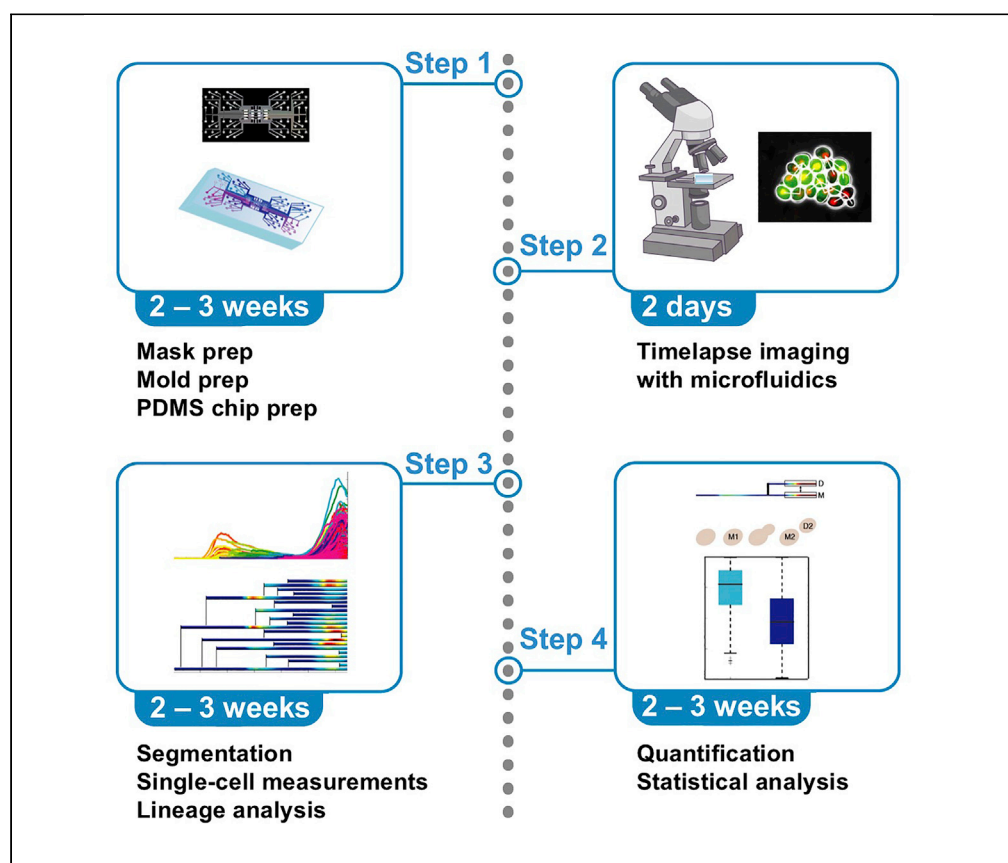
Submitted on 12 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Protocol

Microfluidics for single-cell lineage tracking over time to characterize transmission of phenotypes in *Saccharomyces cerevisiae*



The budding yeast *Saccharomyces cerevisiae* is an excellent model organism to dissect the maintenance and inheritance of phenotypes due to its asymmetric division. This requires following individual cells over time as they go through divisions to define pedigrees. Here, we provide a detailed protocol for collecting and analyzing time-lapse imaging data of yeast cells. The microfluidics protocol can achieve improved time resolution for single-cell tracking to enable characterization of maintenance and inheritance of phenotypes.

Poonam Bheda,
Diana
Aguilar-Gómez, Igor
Kukhtevich,
Johannes Becker,
Gilles Charvin,
Antonis Kirmizis,
Robert Schneider

poonam.bheda@
helmholtz-muenchen.de,
poonam.j.bheda@gmail.
com (P.B.)
robert.schneider@
helmholtz-muenchen.de
(R.S.)

HIGHLIGHTS

Single-cell-tracking
microfluidics and
imaging allow
phenotype and
lineage analysis

Details to prepare
custom microfluidics
chips

Custom software for
cell segmentation

Detailed
quantifications and
statistical analyses of
single-cell lineage
data

Bheda et al., STAR Protocols 1,
100228
December 18, 2020 © 2020
The Author(s).



Protocol

Microfluidics for single-cell lineage tracking over time to characterize transmission of phenotypes in *Saccharomyces cerevisiae*

Poonam Bheda,^{1,8,9,*} Diana Aguilar-Gómez,^{2,8} Igor Kukhtevich,^{1,8} Johannes Becker,³ Gilles Charvin,^{4,8} Antonis Kirmizis,⁵ and Robert Schneider^{1,6,7,*}

¹Institute of Functional Epigenetics, Helmholtz Zentrum München, 85764 Neuherberg, Germany

²Center for Computational Biology, UC Berkeley, Berkeley, CA 94720, USA

³Institute of Bioengineering, School of Engineering, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

⁴Development and Stem Cells, IGBMC, 67400 Illkirch, France

⁵Department of Biological Sciences, University of Cyprus, 2109 Nicosia, Cyprus

⁶German Center for Diabetes Research (DZD), 85764 Neuherberg, Germany

⁷Faculty of Biology, Ludwig-Maximilians Universität München, 80333 Munich, Germany

⁸Technical Contact

⁹Lead Contact

*Correspondence: poonam.bheda@helmholtz-muenchen.de or poonam.j.bheda@gmail.com (P.B.), robert.schneider@helmholtz-muenchen.de (R.S.)
<https://doi.org/10.1016/j.xpro.2020.100228>

SUMMARY

The budding yeast *Saccharomyces cerevisiae* is an excellent model organism to dissect the maintenance and inheritance of phenotypes due to its asymmetric division. This requires following individual cells over time as they go through divisions to define pedigrees. Here, we provide a detailed protocol for collecting and analyzing time-lapse imaging data of yeast cells. The microfluidics protocol can achieve improved time resolution for single-cell tracking to enable characterization of maintenance and inheritance of phenotypes. For complete details on the use and execution of this protocol, please refer to Bheda et al. (2020a).

BEFORE YOU BEGIN

Preparation of yeast reporter strains

⌚ Timing: 1–2 weeks

1. Prepare yeast reporter strains. For fluorescence reporters, yeast can be transformed with self-propagating plasmids or with linear DNA for integration with the appropriate selection markers (Figures 1A–1C, Methods Video S1) (Brachmann et al., 1998; Gietz and Woods, 2002).

Note: In Bheda et al. (2020a) we created a reporter strain in the Y7092 background (Tong and Boone, 2007) by fusing GFP at the C terminus of the endogenous *GAL1* open reading frame (ORF) on chromosome 2 (*chrom II*) by homologous recombination (Figure 1A). The reporter cassette included a superfolder GFP (sfGFP) fused to a degron sequence from *CLN2* (*CLN2-PEST*) to destabilize the reporter, followed by a terminator sequence from *ADH1* (*ADH1-term*), which was cloned into a plasmid. The reporter cassette was directly followed by the *kanMX* selection cassette, which is made of a *TEF1* promoter sequence (*TEF1-prom*), the G418-resistance sequence (G418^R), and a *TEF1* terminator (*TEF1-term*). The reporter +



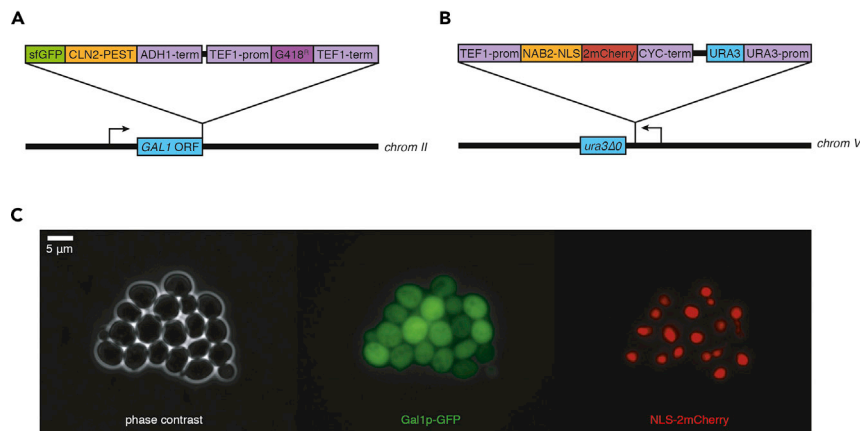


Figure 1. *S. cerevisiae* strain construction with integrated reporter and lineage marker

(A) In Bheda et al. (2020a), we created a reporter strain by fusing GFP at the C terminus of the endogenous *GAL1* open reading frame (ORF) on chromosome 2 (*chrom II*) by homologous recombination. The reporter cassette included a superfolder GFP (sfGFP) fused to a degenon sequence from *CLN2* (*CLN2-PEST*), followed by a terminator sequence from *ADH1* (*ADH1-term*) and a kanMX selection cassette, which is made of a *TEF1* promoter sequence (*TEF1-prom*), the G418-resistance sequence (*G418^R*), and a *TEF1* terminator (*TEF1-term*).

(B) Additionally we created a lineage marker by expressing NLS-2mCherry under the control of the constitutively active *TEF1* promoter (*TEF1-prom*) and a *CYC1* terminator (*CYC1-term*) with a *URA3* selection marker.

(C) Snapshot from Methods Video S1. Left: phase contrast. Middle: Galp-GFP. Right: nuclear 2mCherry.

selection cassette was PCR amplified with oligonucleotides bearing homology to the C terminus of *GAL1* and transformed into yeast. Transformants were selected on YPD plates supplemented with G418 and reporter insertions were tested by junction PCRs and validated by sequencing.

Note: For lineage analyses, we recommend including a second fluorescent reporter to facilitate pedigree mapping. Although some segmentation algorithms, such as the custom software PhyloCell that we used for segmentation and lineage definition (see “Custom software for cell-tracking microfluidics data processing” and the Key resources table for details), are capable of reconstructing lineages without a lineage marker, using one reduces errors. As yeast nuclear envelopes do not break down during cell division, a nuclear fluorescent reporter clearly shows the intact nucleus dividing and separating in two cells – i.e., a mother cell and its daughter cell (Methods Video S1), facilitating pedigree definition. In Bheda et al. (2020a), we used for this a nuclear localization signal (NLS) sequence fused to mCherry (Figures 1B and 1C, Methods Video S1) in order to sequester the fluorescent protein in the nucleus and aid in lineage definition. Specifically we created this lineage marker by expressing 2 copies of monomeric mCherry (2mCherry) with an NLS fusion to the N terminus under the control of the constitutively active *TEF1* promoter (*TEF1-prom*) and a *CYC1* terminator (*CYC1-term*). This lineage marker cassette is located on an integrating plasmid with a *URA3* selection marker. The plasmid was linearized in the *URA3* promoter, transformed in a background strain with a *URA3* deletion (*ura3 Δ 0*) on chromosome V (*chrom V*), and transformants were selected on synthetic drop-out plates lacking uracil. Other suitable lineage markers include the nuclear histone H2B gene *Htb2p* (Goulev et al., 2017; Meitinger et al., 2014; Morlot et al., 2019) or the budneck markers *Cdc10p* and *Myo1p* (Charvin et al., 2008).

Preparation of yeast media

⌚ Timing: 1–2 days

2. Prepare a suitable amount of synthetic media prior to the experiment. See [Materials and equipment](#) for media recipes used in [Bheda et al. \(2020a\)](#).

Note: We recommend preparing twice as much media to allow for potential flow rate problems. Make sure to account for each set of media tubing, flow rate, and length of time lapse. For example, for a chip with 10 independent channels, a flow rate of 10 $\mu\text{L}/\text{min}$, and a 10 h experiment, prepare 120 mL media.

$$10 \text{ channels} \times \frac{10 \mu\text{L}}{\text{min}} \times 600 \text{ min} \times 2 = 120 \text{ mL}$$

Note: Synthetic media is preferable over rich media (e.g., YP) to minimize cellular autofluorescence.

3. On the day of the microfluidics experiment, combine media components and sterile filter through a 0.22 μm filter to remove particulate matter or precipitates from older media that could clog the microfluidics microchambers/microchannels.

Microfluidic chip prep work

⌚ Timing: ~2 weeks

4. Purchase or design and fabricate a microfluidics chip that is appropriate for your needs.

Note: A chip made from polydimethylsiloxane (PDMS) with a #1.5 glass coverslip is recommended. The height of the microchannels where the cells will be trapped should be ~3–3.3 μm for haploid yeast with normal sizes. Mother cells typically have a long axis diameter (as yeast cells are slightly oblong) between 4–6 μm ([Kukhtevich et al., 2020](#)).

The use of custom-made PDMS microfluidics chips for inheritance analyses of *S. cerevisiae* has been previously described ([Bheda et al., 2020a](#); [Fehrmann et al., 2013](#); [Garmendia-Torres et al., 2018](#); [Goulev et al., 2017](#); [Paoletti et al., 2016](#); [Xu et al., 2015](#)). Our custom-designed PDMS microfluidics chip allows haploid yeast cells to be trapped and tracked for ~8 generations ([Figure 2A](#) and see “Limitations”) ([Bheda et al., 2020a](#); [Goulev et al., 2017](#)).

To prepare a custom-made PDMS chip as in [Bheda et al. \(2020a\)](#) start with mask fabrication of the desired design ([Figure 2B](#)):

5. Prepare a high-resolution computer-aided-design (CAD) file with the desired microfluidics chip design.

Note: Our design includes a 3- μm layer for trapping yeast and an ~40- μm layer for the media supply channels. The CAD file for the microfluidic chip design is available on Mendeley Data ([Bheda et al., 2020b](#)).

6. Prepare photomasks of the microfluidics chip by printing the high-resolution CAD files for each layer onto a polymer slide.

Note: We have outsourced this step (Selba, Switzerland).

Using these masks, prepare a replica mold as follows ([Figure 2B](#)):

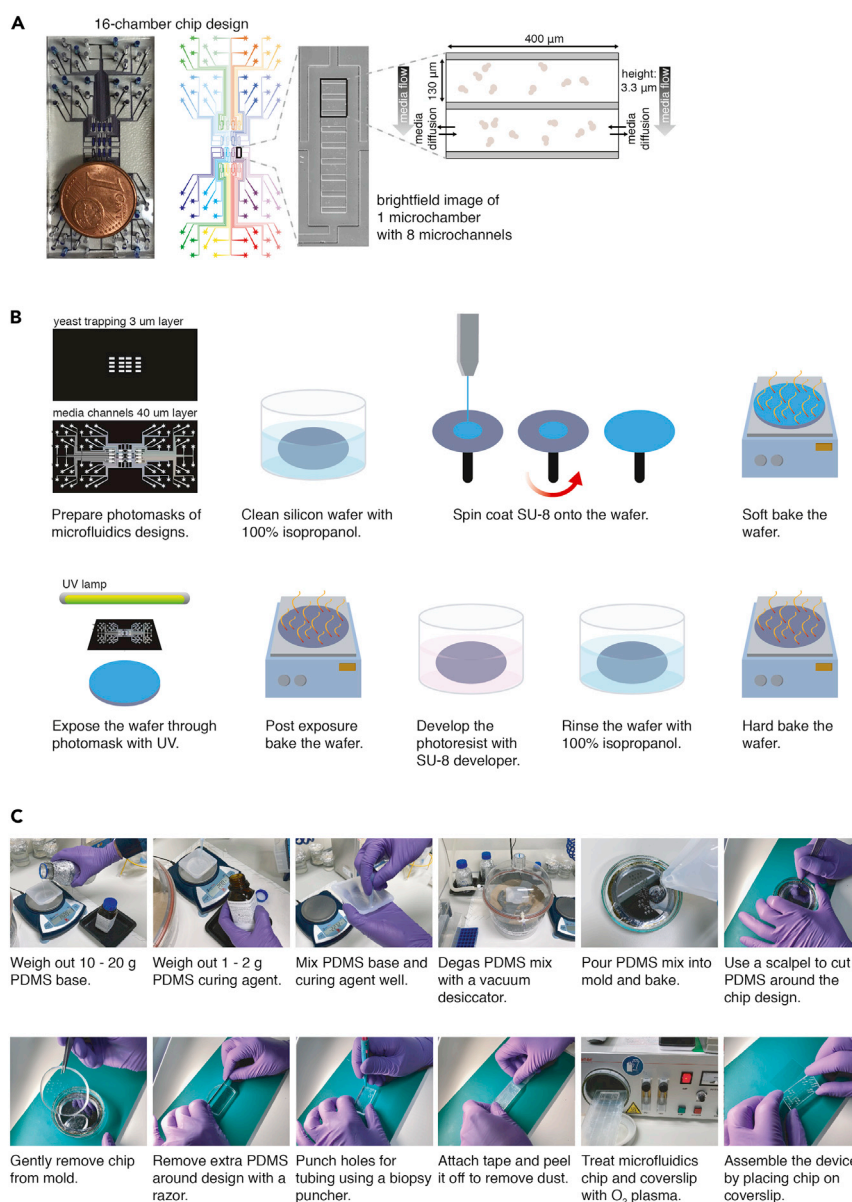


Figure 2. Cell-tracking microfluidics chip

(A) Cell- and lineage-tracking custom microfluidics design (figure modified from Figure 1A in [Bheda et al., 2020a](#)). The chip is designed with 16 independent microchambers, with each having its own media and cell inlet and outlet channels (represented by different colors), where different strains or conditions can be tested simultaneously. Each microchamber has 8 microchannels for trapping the yeast such that 8 regions containing cells of interest can be imaged per strain/condition ([Goulev et al., 2017](#)).

(B) Mold fabrication using photomasks and SU-8 photoresist. Photomasks are made from CAD files designed for each layer of the microfluidics chip, then printed onto slides. The mold is made by 2-layer photolithography using a silicon wafer. The process for each layer involves using a spin coater to evenly spread SU-8 photoresist on the wafer and UV treatment through each photomask to transfer the design onto the wafer. This process results in a negative replica mold that can be used repeatedly to prepare PDMS microfluidics chips.

(C) Preparation of a PDMS chip stepwise from left to right. Liquid PDMS mix is poured into the replica mold and baked. The solidified PDMS is then assembled into a microfluidics chip by punching holes, treating with O_2 plasma, and attaching to a coverslip. For details see text.

7. Start the process with the 3- μ m layer for trapping yeast by cleaning a 4-inch silicon wafer with 100% isopropanol, then baking it at 120°C for ~15 min to dry it out.
8. Use a spin coater to evenly spread 3 mL of SU-8 2002 photoresist onto the wafer at 400 rpm.
9. Soft bake the wafer for 2 min at 95°C.
10. Expose the SU-8 2002 on the wafer through the photomask with 100 mJ/cm² UV at 365 nm.
11. For the post exposure bake, bake the wafer for 2 min at 95°C.
12. Develop the SU-8 2002 for 15–30 s in a bath of ~50 mL SU-8 developer.
13. Rinse the wafer with 100% isopropanol and dry with compressed air.
14. Perform a hard bake by baking the wafer for 15 min at 150°C.
15. For the second, 40- μ m layer for the media supply channels, use the spin coater to spread 3 mL of SU-8 2025 on top of the first layer at 2,500 rpm.
16. Soft bake the wafer for 7 min at 95°C.
17. Align the second photomask with the first layer manually or with a mask aligner.
18. Expose the SU-8 2025 on the wafer through the second photomask with 160 mJ/cm² UV at 365 nm.
19. For the post exposure bake, bake the wafer for 2 min at 65°C, followed by a 6 min bake at 95°C.
20. Develop the SU-8 2025 for 45–60 s in a bath of ~50 mL SU-8 developer.
21. Rinse the wafer with 100% isopropanol and dry with compressed air.
22. Perform a hard bake by baking the wafer for 15 min at 150°C.

△ CRITICAL: This process is best carried out in a microelectronics cleanroom.

▮▮ Pause Point: The replica mold can now be stored and reused indefinitely.

On the day of the experiment, prepare the PDMS microfluidics chip using the replica mold (Figure 2C):

23. Weigh out 10:1 PDMS to curing agent (included in the PDMS kit) and mix well.

Note: 10 g PDMS + 1 g curing agent is sufficient for 1 chip. If cutting a larger area around the chip, prepare up to 20 g PDMS + 2 g curing agent.

24. Degas PDMS mix using a vacuum pumping unit connected to a vacuum desiccator for 10–30 min with 15–20 mbar pressure until no air bubbles remain.
25. Pour the PDMS mix into the mold and let sit for 10 min to remove any remaining air bubbles.

Alternatives: the PDMS mix can be first poured into the mold and then degassed using the vacuum desiccator.

26. Bake the microfluidics chip in the mold at 100°C in an incubator for 1 h to crosslink the PDMS.

Note: PDMS can be baked in the mold ahead of time and with different lengths of time, depending on temperature. For every 10°C less, double the time (e.g., 2 h at 90°C, 4 h at 80°C, 8 h at 70°C, or 16 h at 60°C).

△ CRITICAL: Keep microfluidics chip in mold until use.

▮▮ Pause Point: The PDMS chip in the replica mold can be stored indefinitely.

27. On the day of the cell-tracking microfluidics data collection, cut out the PDMS replica from the mold using a scalpel.

Caution: Use caution when cutting the chip out of the mold. Cut deep enough to remove chip, while minimizing scratches to the mold.

Note: If you cut out a large area around the chip to remove from the mold, the edges should be trimmed to the size of the coverslip that it will be attached to, using a scalpel or preferably a long razor blade.

Caution: Avoid touching the surface of the microfluidics chip as the microchannels can easily be damaged.

28. Place the microfluidics chip with the microchannels side up and punch out holes for connecting tubing using a 1–1.5 mm biopsy puncher.

Note: The choice of biopsy puncher size depends on the type of tubing and connection to the chip. If using Tygon tubing and bent needles with the specifications as in [Bheda et al. \(2020a\)](#) and in the Media supplementation section and [Key resources table](#), we suggest using the 1 mm biopsy puncher. We have also successfully used Teflon tubing directly inserted in the chip. In this case, a larger hole is preferable; therefore, for this type of tubing connection to the chip, we suggest using a 1.5 mm biopsy puncher.

29. Dust particles on the surface of the chip should be cleaned by gently attaching a piece of Scotch tape evenly across the surface and peeling the tape away.

Caution: Dust particles on the surface of the chip can block media flow through the microchannels.

30. Using forceps, wash a 24 × 50 mm #1.5 glass coverslip first by completely immersing in 100% ethanol for ~5 s, followed by immersing the coverslip in ddH₂O for ~5 s, and then dry with compressed air.

Alternatives: Soak coverslips for at least 1 h in H₂O₂, rinse with ddH₂O, and dry with compressed air.

31. Treat the microfluidics chip (microchannel side up) and coverslip with O₂ plasma cleaner surface activation.

Note: Using the Zepto B model plasma cleaner from Diener, the microfluidics chip and chamber can be treated with 6.0 power with the timer set to 0–40 (~2 min) under 0.8–1 mbar pressure with 100% O₂.

32. Assemble the chip by carefully placing the PDMS (microchannel side down) on top of the coverslip. Carefully press the chip and coverslip together to achieve full contact. Allow the chip to bond to the glass for 10 min at 20°C–25°C.

⚠ CRITICAL: Do not push too hard on the chip otherwise the microchannels may be crushed. You should be able to see a contrast between where the PDMS is in contact with the glass and where it is still in contact with air. With careful pressing, the air front should spread and disappear.

⏸ Pause Point: It is best to plasma clean and assemble the microfluidics chip just prior to setting up the cell-tracking microfluidics experiment to retain hydrophilicity in the microchannels that results from O₂ plasma treatment, however the assembled chip can be stored for days to weeks.

Preparation of cell and media tubings

⌚ Timing: 1 day

To prepare tubing for media supplementation and to inject cells into the microfluidics chip as in Bheda et al. (2020a):

33. Pull out the needle (18 gauge, 90° bent, blunt edge) from plastic used for syringe attachment.
34. Insert needle into one end of Tygon tubing with 0.02" inner dimension × 0.060" outer dimension.

⏸ Pause Point: Cell and media tubings can be stored and reused indefinitely.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Chemicals, peptides, and recombinant proteins		
Drop-out Mix Complete w/o yeast nitrogen Bbase	US Biological	Cat# D9515
Yeast Nitrogen Base without amino acids	Becton-Dickinson	Cat# 291940
D-(+)-Raffinose	Sigma-Aldrich	Cat# 83400
D-(+)-Galactose	Sigma-Aldrich	Cat# G6404
D-(+)-Glucose	Sigma-Aldrich	Cat# G8270
Polydimethylsiloxane (PDMS) kit Sylgard 184	Dow Corning	Cat# 2646340
SU-8 2002 photoresist	MicroChem	Cat# SU-8 2002
SU-8 2025 photoresist	MicroChem	Cat# SU-8 2025
SU-8 developer	MicroChem	Cat# Y020100
Experimental models: organisms/strains		
<i>Saccharomyces cerevisiae</i>	n/a	n/a
Software and algorithms		
PhyloCell	Gilles Charvin	https://github.com/gcharvin/Goulev et al., 2017
Autotrack	Gilles Charvin	https://github.com/gcharvin/Goulev et al., 2017
Single-cell lineage analysis sample code	Diana Aguilar-Gómez	https://github.com/aguilar-gomez/SingleCellLineageAnalysis
MATLAB	MathWorks	versions R2009a, R2016a, and R2016b
Python	Python Software Foundation	versions 2.7 and 3.6.3
R	R Foundation for Statistical Computing	version 3.4.4
Nikon NIS Elements	Nikon Microscope Solutions	version 4.51.c1
Other		
1 mm biopsy puncher	Integra Miltex, USA	Cat# 33-31AA-P/25
1.5 mm biopsy puncher	Integra Miltex, USA	Cat# 33-31A-P/25
Vacuum pumping unit	Vacuubrand, Germany	Cat# PC 3004 VARIO
Vacuum desiccator	Nalgene, USA	Cat# 5311-0250

(Continued on next page)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Incubator	Binder, Germany	Cat# ED 56
Scotch Magic invisible tape	3M, USA	Cat# 810
Steritop 0.22 μ m PES filter	Millipore, USA	Cat# 2GPT02RE
1 mL syringe	B. Braun, Germany	Cat# 9166017V
22 gauge 0.5 inch blunt edge needle	Adhesive Dispensing, UK	Cat# AD722050
Thermoelectric modules	RS, France	Cat# 490-1317
Temperature regulator	Oven Industries, USA	Cat# 5R6-900
Spin coater	Mikasa, Japan	Cat#MS-B100
O ₂ plasma cleaner	Diener Electronic, Germany	Cat# Zepto B
24 × 50 mm #1.5 glass coverslip	Knittel Glass, Germany	Cat# VD12450Y1A.01
Tygon tubing 0.02 inch ID × 0.060 inch OD	Cole-Parmer, Germany	Cat# GZ-06419-01
18 gauge 0.5 inch bent 90° blunt needle	Techcon, USA	Cat# TE718050B90PK
Microfluidic flow controller	Elveflow, France	Cat# OB1 MK3+
Microfluidic flow sensor	Elveflow, France	Cat# MFS 3
Port selector valve	Idex, USA	Cat# MXX778-605
Peristaltic pump	Ismatec, Switzerland	Cat# ISM 931
Objective heater	Okolab, Italy	n/a
Heating chip holder	custom-built	n/a
Epifluorescence microscope	Nikon, Japan	Cat# Eclipse Ti-E
Objective	Nikon, Japan	Cat# plan-apo λ 100 \times /1.45 NA Ph3 oil immersion
Light engine	Lumencor, USA	Cat# SPECTRA X
EMCCD camera	Andor, UK	Cat# iXon Ultra 888
CAD file for mask fabrication	Bheda et al., 2020b	https://data.mendeley.com/datasets/yr2nr5syx/1

MATERIALS AND EQUIPMENT

Equipment

Media supplementation equipment

For automated media supplementation in Bheda et al. (2020a), we used a pressure microfluidic flow controller with a flow sensor. Automated media changes were controlled by a port selector valve system (Figure 3A). Media is supplied via tubing prepared with bent needles (see connected to the microfluidics chip (Figure 3B, left).

Inverted epifluorescence microscope

Phase contrast and fluorescence images in Bheda et al. (2020a) were captured with Nikon NIS Elements software from an inverted Nikon Eclipse Ti-E epifluorescence microscope using a plan-apo λ 100 \times /1.45 NA Ph3 oil immersion objective, SPECTRA X light engine illumination, and an Andor iXon Ultra 888 EMCCD camera.

Temperature control equipment

An objective heater is important to maintain the temperature (normally 30°C for yeast cells) during a microfluidics experiment. We used an objective heater and also a custom-built microfluidic chip holder with standard thermoelectric modules controlled by temperature regulation devices (Figures 3A and 3B) (Charvin et al., 2008). The heating chip holder is designed with holes so that bars or holders can be screwed in to hold the chip in place, which is important to prevent movement so

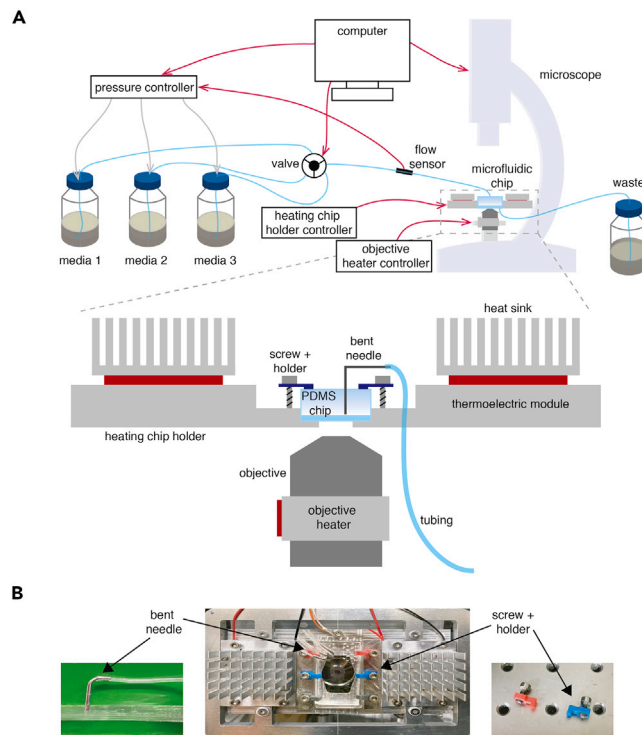


Figure 3. Cell-tracking microfluidics setup

(A) Schematic of microfluidics setup. Top: red lines indicate regulation by controlling units. Gray lines indicate flow of air. Blue lines indicate tubing for media flow. Software on the computer controls the pressure controller, valve, and microscope. The pressure controller also receives input from the flow sensor, allowing it to adjust the pressure in case flow is too high or low. The heating chip holder and the objective heater are regulated by independent controllers. Bottom: zoomed in schematic of custom-built heating chip holder with thermoelectric modules, objective heater, tubing with bent needles, and screws + holders.

(B) Left: photo of tubing with bent needle inserted in PDMS. Middle: photo of heating chip holder. Right: photo of 3D-printed holders.

that cells of interest are continually tracked and remain in focus. We have used custom 3D-printed holders (Figure 3B, right). We have found that the objective heater is the most important temperature regulation equipment – the heating chip holder is not necessary, but helps to buffer major temperature changes.

Recipes

Synthetic complete (SC) media

We recommend first preparing the media components by autoclaving the amino acid mix plus yeast nitrogen base together and separately sterile filtering the carbon source (sugar). To prepare media as in [Bheda et al. \(2020a\)](#):

1.25 × SC stock

Reagent	Final concentration	Per liter
Drop-out Mix Complete w/o Yeast Nitrogen Base	1.25×	2.75 g
Yeast Nitrogen Base without Amino Acids	1.25×	8.375 g
autoclaved ddH ₂ O		up to 1 L

Note: autoclave on the same day that it is prepared

20% raffinose

Reagent	Final concentration	Per liter
raffinose	20%	200 g
autoclaved ddH ₂ O		up to 1 L

Note: sterile filter through a 0.22 µm filter on the same day that it is prepared

20% glucose

Reagent	Final concentration	Per liter
glucose	20%	200 g
autoclaved ddH ₂ O		up to 1 L

Note: sterile filter through a 0.22 µm filter on the same day that it is prepared

15% galactose/15% raffinose

Reagent	Final concentration	Per liter
galactose	15%	150 g
raffinose	15%	150 g
autoclaved ddH ₂ O		up to 1 L

Note: sterile filter through a 0.22 µm filter on the same day that it is prepared

On the day of the microfluidics experiment, combine media components and sterile filter to remove particulate matter or precipitates from older media that could clog the microfluidics microchambers/ microchannels.

SC+2% raffinose

Reagent	Final concentration	Per liter
1.25× SC stock	1×	800 mL
20% raffinose	2%	100 mL
autoclaved ddH ₂ O		100 mL

Note: prepare and sterile filter through a 0.22 µm filter on the day of the experiment

SC+2% glucose

Reagent	Final concentration	Per liter
1.25× SC stock	1×	800 mL
20% glucose	2%	100 mL
autoclaved ddH ₂ O		100 mL

Note: prepare and sterile filter through a 0.22 µm filter on the day of the experiment

SC+1.5% galactose/1.5% raffinose

Reagent	Final concentration	Per liter
1.25× SC stock	1×	800 mL
15% galactose/15% raffinose	1.5%/1.5%	100 mL
autoclaved ddH ₂ O		100 mL

Note: prepare and sterile filter through a 0.22 μm filter on the day of the experiment

Custom software for cell-tracking microfluidics data processing

Cells were segmented and mapped from frame to frame and nuclei were segmented to build pedigrees from the single-cell tracking microfluidics data in Bheda et al. (2020a) using the custom software PhyloCell and Autotrack (<https://github.com/gcharvin>) that operates in MATLAB. These software have been previously implemented for lineage analyses of *S. cerevisiae* (Bheda et al., 2020a; Fehrmann et al., 2013; Garmendia-Torres et al., 2018; Goulev et al., 2017; Paoletti et al., 2016; Xu et al., 2015). For help using these software, there are available files on Github as HTML web pages (<https://github.com/gcharvin/phyloCell/help/html/index.html>). In addition, there is a demo project that can be launched from PhyloCell to try out the features of the software (<https://github.com/gcharvin/phyloCell/demo>).

Alternatives

Microfluidic chip

Commercially available microfluidics chips for yeast, such as CellASIC ONIX plates from Millipore, are available.

Microscope

Other inverted epifluorescence microscopes are compatible with time-lapse microfluidics imaging. We have also successfully used a Zeiss AxioObserver.

Media supplementation

Other media flow systems can be used with time-lapse microfluidics imaging. We have also successfully used a peristaltic pump instead of a pressure flow controller.

Temperature control equipment

We have also used a heated box around the microscope for experiments where rapid increases in temperature are important, such as temperature-sensitive mutants that are grown under temperature-permissive conditions and then changed to temperature-restrictive conditions. The heated box helps to facilitate the temperature change, but this equipment is not absolutely required.

Segmentation software

There are many alternative segmentation software programs for budding yeast cells, some of which can also track cells and lineages, such as CellStar (Versari et al., 2017).

STEP-BY-STEP METHOD DETAILS

Cell-tracking microfluidics data collection

⌚ Timing: 36 h/1.5 days

This step describes how to collect microfluidics data for single-cell tracing.

Day 1

1. Set up yeast cultures by inoculating 3 mL synthetic media of choice in a 15-mL tube with yeast from a freshly streaked plate (less than 1 month old) and grow at 30°C with shaking at 250–300 rpm for 10–15 h.

Day 2

2. Filter media through a 0.22 μm filter to remove particulate matter or precipitates from older media that could clog the microfluidics microchambers/microchannels.

3. Dilute cells by inoculating a fresh culture of 3 mL media with 25–50 μL culture from step 1 and grow at 30°C with shaking at 250–300 rpm for ~ 4 h (~ 2 doublings) until cells reach low log phase ($\sim 3\text{--}9 \times 10^6$ cells/mL).

Note: Many strains of *S. cerevisiae* grown in typical growth media are in log phase with a concentration of $\sim 0.3\text{--}3 \times 10^7$ cells/mL. In most spectrophotometers, this corresponds to an optical density (OD) from 0.1 to 1; however, this should be previously determined. Using spectrophotometers where *S. cerevisiae* log phase is 0.1–1, low log phase corresponds to OD readings between 0.1 and 0.3.

4. Place microfluidics chip in the heating chip holder and secure the chip with screws + holders.

△ CRITICAL: The chip should be sufficiently tightly secured such that movement is not possible during the time-lapse experiment; however, take care not to apply too much pressure on the chip, as this can cause the coverslip to break. Securing the chip is important to prevent movement so that cells of interest are continually tracked and remain in focus.

5. Connect all tubings (media and cell) to the chip and prime with media.

Note: We recommend using ~ 10 cm of tubing for cell inlet and outlet. The tubing lengths for media inlet and outlet will depend on your setup, but ~ 50 cm tubing should be sufficient for most setups.

Note: We recommend priming the chip and tubings with enough media until a continuous flow is observed into the waste (see [Figure 3A](#)).

Optional: Tape the tubings onto the stage. Tubings can move during the experiment and affect the focus of the cells.

6. Carefully inject cells obtained from the low log phase culture into each microchamber using a 1 mL syringe and a 22 gauge blunt edge needle inserted into 1 of the 2 cell tubes attached to each microchamber (the other cell tube allows uncaptured cells and media to flow out of the chip) such that each microchannel has very few cells.

△ CRITICAL: It is best that the starting cells that will be captured and tracked have a bud. This helps to avoid capturing old/sick/arrested/dying cells. In case unwanted cells are captured, they can be flushed out by injecting more culture with a slightly harder push on the syringe.

Note: Choose the number of starting cells depending on the purpose and length of the experiment. For example, in a memory experiment as in [Bheda et al. \(2020a\)](#), we started the experiment with ~ 2 budding cells in each region of interest to capture ~ 8 generations.

7. Start pumping media through chip.

Note: We recommend a flow rate of 10–50 $\mu\text{L}/\text{min}$. Higher flow rates can lead to loss of cells (see “[Troubleshooting](#)”).

8. Choose positions to image. Set focus for each position. Collect images with 3-min resolution with phase contrast and any filters for fluorescent molecules of interest.

Note: For segmentation and analysis by PhyloCell, we recommend the following order of acquisition: (channel 1) phase contrast, (channel 2) fluorescent protein of interest, (channel 3) lineage marker.

Note: Exposure levels and times will need to be determined for each protein of interest such that signal is not saturated and there is minimal photobleaching.

Note: According to our experience, capturing images every 3 min provides enough time resolution for cell mapping and to capture budding events for lineage tracking without resulting in severe phototoxicity (see “[Limitations](#)”).

△ **CRITICAL:** We have successfully used the Perfect Focus System (Nikon) and the Definite Focus (Zeiss) autofocus systems to minimize focus loss. Even with autofocus systems, there can be focus loss due to e.g., tubing movement, channel clogging, or thermal expansion. Therefore, we recommend checking the progress of the experiment after 30 min to 1 h to ensure that cells are still in focus.

Day 3

9. After the experiment is complete, clean media and cell tubings by pumping first ddH₂O, then 80% ethanol, and then drying them completely by letting air run through.

Note: We recommend a flow rate of ~300 µL/min for 20 min for each step, therefore running ~6 mL of each liquid through the tubings.

Cell-tracking microfluidics data processing

This step describes how images collected from the single-cell tracking microfluidics data should be processed prior to quantification and statistical analyses. We have used the custom software PhyloCell (see “[Materials and equipment](#)” above for further details).

10. Segment cells using fluorescence or phase contrast images.

Note: To use PhyloCell for this step, the images should first be imported to generate a segmentation project. We recommend JPEG image files. PhyloCell can generate a segmentation project in a number of ways. For more details, please access the help files available for PhyloCell on Github (see “Loading images and generating segmentation projects with PhyloCell”). For Nikon NIS Elements users, we have a function available to convert the ND2 file format into a PhyloCell project with JPEG images.

Optional: Segment lineage markers.

11. Map cells so that they are tracked from frame to frame.
12. Build pedigrees based on mother-bud proximity or lineage markers (see “[Preparation of yeast reporter strains](#)” in “[Before you begin](#)” above).
13. Correct any segmentation, mapping, or lineage errors.
14. Measure fluorescence in segmented cells.

Note: This information can be used as absolute fluorescence or as normalized to cell area.

15. Subtract cellular autofluorescence and/or background fluorescence to determine fluorescence signal of interest.

Note: Different kinds of media (e.g., synthetic vs rich or glucose vs galactose) result in different autofluorescence levels. Therefore, rather than subtracting autofluorescence signal in the strain of interest from a non-inducing medium, it is preferable to use a strain with the same background lacking the reporter.

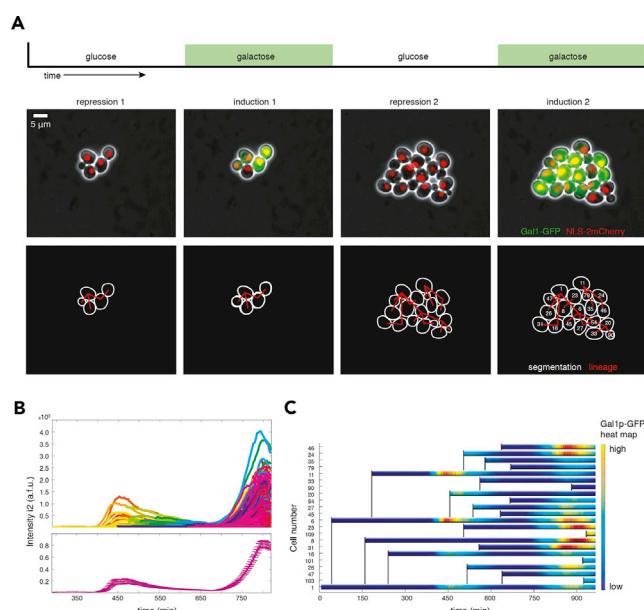


Figure 4. Expected outcomes

(A) Cells captured in the microfluidics chip are tracked over time with live imaging. In Bheda et al. (2020a), reporter cells with constitutively expressed NLS-2mCherry lineage marker were repeatedly exposed to galactose, resulting in induction of Gal1p-GFP.

(B) Top: single-cell traces of Gal1-GFP expression over two inductions during time lapse. Each colored line represents a different single cell. Bottom: mean Gal1-GFP expression, error bars indicate SEM.

(C) Example pedigree of mother cells and their progeny with Gal1-GFP expression depicted by heat map to visualize induction inheritance over time. Vertical black lines indicate daughter cell birthtimes.

Note: Background fluorescence can be calculated using a contour in the image that does not contain any cells.

EXPECTED OUTCOMES

The microfluidics setup described in the protocol above will allow you to capture and image cells to track them using segmentation and mapping over time. With high time resolution and the aid of lineage markers, it is possible not only to track the cells, but also define the pedigree of all the cells (Figure 4). In Bheda et al. (2020a), we tracked a Gal1p-GFP reporter yeast strain that was repeatedly exposed to galactose. Mother cells were present throughout the whole experiment, resulting in two inductions of Gal1p-GFP. These cells were segmented and their lineages (daughter cells) were defined by the constitutively expressed NLS-2mCherry marker (Figure 4A). By measuring the fluorescence in each of the cells, we observed memory by higher expression during a second induction in both mother cells and previously unexposed daughter cells (Figure 4B). Building a pedigree of mother cells and their progeny allowed us to visualize the inheritance of phenotypes (Figure 4C). Together, this provides the basis of the single-cell quantification and statistical analyses in the following sections to dissect maintenance and inheritance of phenotypes.

QUANTIFICATION AND STATISTICAL ANALYSIS

We describe here different possible ways (A to D below) with which data from the single-cell tracking microfluidics experiments can be analyzed for single-cell behavior and/or lineage evaluation, mainly following what has been implemented in Bheda et al. (2020a) (Figure 5A). We have used these calculations to compare memory in individual mother cells to themselves during two inductions (maintenance of memory) or to analyze memory transmittance from mother to daughter cells by using the lineage information (inheritance of memory). Furthermore, we have compared maintenance and

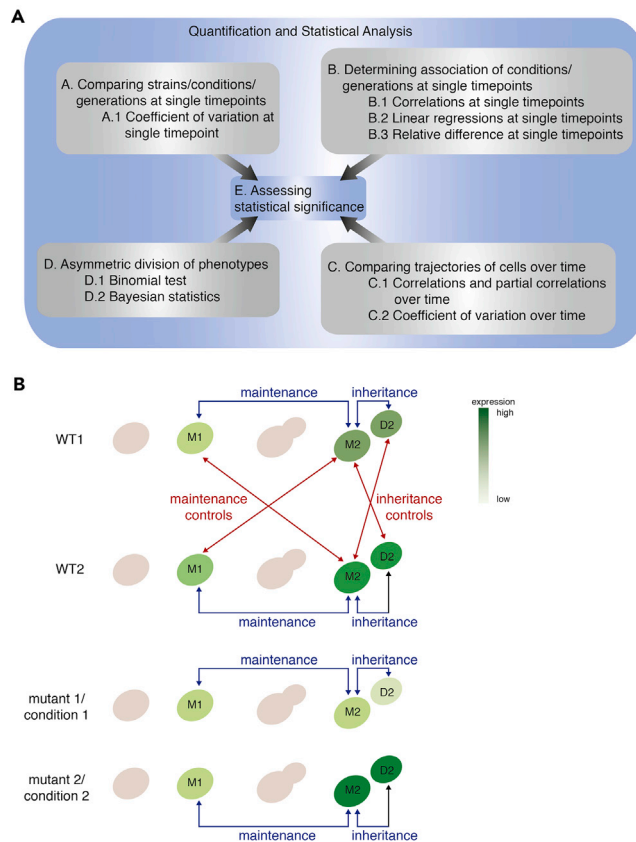


Figure 5. Overview of possible options for single-cell lineage-tracking analysis

(A) Cell-tracking microfluidics data can be quantified and statistically analyzed in myriad ways. Here we depict various possibilities A–D described in the text within the section on Quantification and statistical analysis. These analyses, and the sub-analyses listed beneath are not hierarchical, but can be applied independently of each other, based on the required analysis of the collected data.

(B) The strength of the cell-tracking microfluidics setup implemented is the ability to dissect transmission of phenotypes into maintenance in mother cells (M) and inheritance into their specific daughter cells (D). Maintenance is measured by analyzing differences between a mother cell at an initial time point (M1) and the same cell at a later time point (M2) (top panel). Inheritance is measured between mothers and daughters at the same time point after the birth of the daughter (M2 and D2). Maintenance and inheritance analyses are depicted by blue arrows, which are compared against controls such as shuffled pairs of mothers or mother-daughter pairs as depicted by the red arrows. In addition, maintenance and inheritance in wild-type (WT) cells can be compared to different conditions or mutants (bottom panel). We employed these analyses to dissect maintenance and inheritance of phenotypes during repeated inductions and found e.g., asymmetric inheritance as depicted in mutant 1/condition 1 and gain-of-memory such as in mutant 2/condition 2. Please note, these analyses can be modified to address other important questions on single-cell inheritance in yeast and even other cell types.

inheritance of memory between a wild-type strain and several mutant strains (Figure 5B). However, the calculations and analyses listed below are applicable to a wide variety of topics beyond memory, but are by no means exhaustive and depend on the question(s) of interest.

Overview of the data

Simple plots can be generated for an overall visualization of the microfluidics experiment results.

1. Organize data using the package pandas (McKinney, 2010) in Python which works with dataframe resulting in two-dimensional data structures with a tabular format. Pandas can be used to read tab separated files, among other file formats, into dataframes (e.g., `DataFrame = pandas.read_table("input.tsv")`). Using pandas, data can be easily grouped by a specified variable and this can

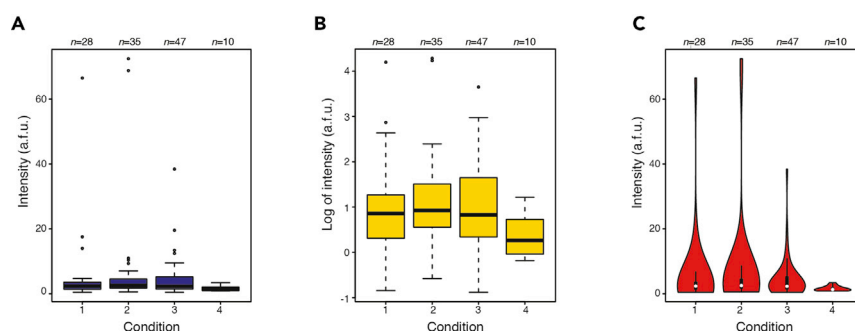


Figure 6. Analysis figures option A: comparing strains/conditions/generations at single time points

(A–C) (A) Boxplots with linear scale, (B) boxplots with log scale, and (C) violinplots to visualize the same data. These figures were generated in R using the functions *boxplot()* and *vioplot()*. Boxplot elements are as follows: center line, median; diamond, mean; box limits, upper and lower quartiles; whiskers, 1.5× interquartile range; crosses, outliers. Violinplot elements are as follows: white circle, median; box limits, upper and lower quartiles; whiskers, 1.5× interquartile range; violin, rotated kernel density.

facilitate statistical analysis or plotting of data. For example, useful functions to do this in pandas are *DataFrame.set_index()* and *DataFrame.groupby()*.

2. Plot single-cell traces to observe the phenotype(s) (e.g., fluorescence signal) over time (Figure 4B, top). In Python, some useful packages for plotting are seaborn (Waskom et al., 2014) and matplotlib (Hunter, 2007). The *lineplot* function from seaborn (available from seaborn v0.10.0 onwards, previously *tsplot()*) allows to plot time series with error bands, confidence intervals, and data ranges, among many other features. An example using this function on microfluidics data over 200 data points, 3 different cell strains, and more than 450 cells, is available in the TimeSeriesPlot directory on Github (<https://github.com/aguiar-gomez/SingleCellLineageAnalysis/>).
3. Calculate the mean or the median of the phenotype(s). Include confidence intervals, standard deviation, or other measurements to represent the variation (Figure 4B, bottom). Useful packages to perform these statistical analyses in Python are scipy (Virtanen et al., 2020) and numpy (Harris et al., 2020).
4. Plot pedigree of mother cells and their progeny according to a heatmap of the phenotype to observe the phenotype(s) over time and its potential inheritance (Figure 4C). This can be done in the PhyloCell GUI with *Plot Pedigree/Parentage* from the *Analyze* tab.

Method A: Comparing strains/conditions/generations at single time points

To compare single-cell phenotypes (e.g., different strains, or the same strain at different time points or in different conditions, or mothers vs daughters), distributions at single time points of interest can be visualized.

1. Plot data for phenotype of interest grouped by e.g., strain or condition using boxplots, violinplots, kernel densities, or histograms (Figures 6A–6C). As mentioned above, the *seaborn* and *matplotlib* packages in Python are useful for data visualization, including the functions *matplotlib.pyplot.boxplot()* and *seaborn.boxplot()*. Examples using these functions are available on the provided Github link (<https://github.com/aguiar-gomez>). An alternative package for plotting data using a language other than Python is *ggplot2* in R (Wickham, 2011).

Note: Depending on the type of data some plots might be favored:

Quantitative (numerical) data: histograms, boxplots, violinplots, and scatterplots might be preferred. Boxplots depict where most of the data lies, as the box represents the 25th to 75th percentiles. Violinplots are very similar to boxplots, but additionally they have a rotated kernel density on each side.

Categorical (ordinal and nominal) data: bar charts and dot charts. The counts of each categorical data can be plotted with bar charts. Dot plots can be used to plot the difference in proportions per category.

2. Perform statistical tests to confirm actual differences between groups.

Note: For more information on choosing statistical tests, refer to section on Assessing statistical significance.

Coefficient of variation at single time point

1. Calculate a coefficient of variation (CV) for all cells within a population at a specific time point to determine how noisy (variable) gene expression is in each population. To obtain the mean and standard deviation, *numpy* functions can be used such as *numpy.mean()* and *numpy.std()*, respectively.

$$CV = \frac{\sigma}{\mu}$$

Where σ indicates the standard deviation and μ indicates the mean. For N cells with the observed phenotypes (x_1, x_2, \dots, x_n) the CV is calculated in the following way:

$$CV(x_1, x_2, \dots, x_n) = \frac{\sqrt{\frac{\sum_{i=1}^n (\mu - x_i)^2}{N}}}{\mu}$$

2. Bootstrap cells in each time point to create a distribution of CVs and to calculate error bars by the standard error of the mean (SEM) or the standard deviation (SD). For our applications we used 1,000 replicates, but for time or memory consuming processes, 100 may be enough to get a rough estimate (Pattengale et al., 2010). Although there are no strict rules regarding the number of bootstrap replicates that should be applied, in general 599 bootstrap replicates are recommended for a wide range of applications (Wilcox, 2010). The functions *bootstrap()* and *errorbars()* that have been used are available in the Python notebook (SLCV.ipynb) within the directory *Sublineage_CV* on Github (<https://github.com/aguiar-gomez/SingleCellLineageAnalysis>).
3. Perform statistical tests to confirm actual differences between groups.

Method B: Determining association of conditions/generations at single time points

To determine whether two phenotypes that have been observed are related, the data for one phenotype can be plotted as a function/relative to the second phenotype (e.g., different strains, or the same strain at different time points or in different conditions, or mothers vs daughters).

1. Plot the phenotypes as a scatter plot against each other, with one observation on the x axis and the other along the y axis (Figure 7A). A function called *matplotlib.pyplot.scatter()* can be used to generate scatterplots in Python, or alternatively the *geom_point()* function within *ggplot2* in R.

Correlations at single time points

1. Calculate correlations to determine how closely associated (linear) two phenotypes (x and y) are (Figure 7A).

Note: We used Pearson's correlation in Bheda et al. (2020a), but there are other types such as Spearman's.

Pearson's correlation assumes a linear relationship (constant rate and same direction) between the variables. Spearman's rank correlation assumes monotonic relationship (same direction not necessarily same rate), but is in fact a special case of Pearson's correlation.

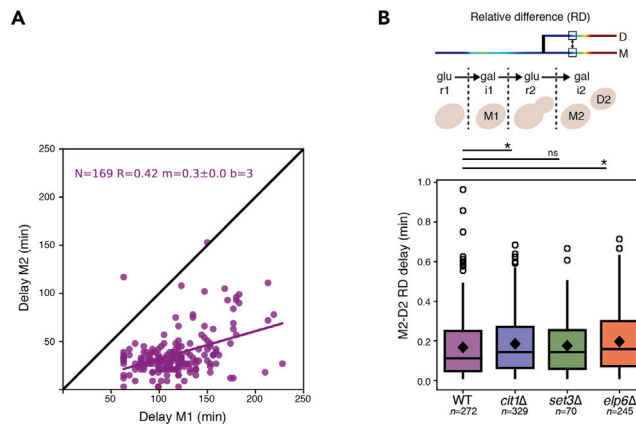


Figure 7. Analysis figures option B: determining association of conditions/generations at single time points

(A) Scatter plot with correlation coefficient and linear regression. In Bheda et al. (2020a), we plotted the delay until detectable Gal1-GFP expression for mother cells to themselves in a first versus second induction and calculated a corresponding correlation coefficient of $R = 0.42$ and a linear regression with a slope $m = 0.3$.

(B) Relative difference calculations. We also used relative difference (RD) as a measure for relatedness by the difference in delay until detectable Gal1-GFP expression in mother-daughter pairs for different strains (from Figure 4A in Bheda et al., 2020a). These figures were generated in Python using the functions library Matplotlib. Boxplot elements are as follows: center line, median; diamond, mean; box limits, upper and lower quartiles; whiskers, $1.5 \times$ interquartile range; crosses, outliers.

$$R = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y}$$

The cov is the covariance between x and y and σ indicates the standard deviation. R indicates how well correlated the variables are, where 1 equals a perfect correlation. R^2 , which is also known as the coefficient of determination, indicates the proportion of the variance in the variable that is predictable from the explanatory variables. Pearson's coefficient of correlation could be calculated using the function `pearsonr()` found within the statistical functions library `scipy.stats` in Python.

Linear regressions at single time points

1. Calculate linear regressions to determine whether one phenotype can be predicted by (or is dependent on) another (single, Figure 7A) or several explanatory variables (multiple).

For n cells with two phenotypes or n pairs of cells, there are n datapoints: $\{(x_i, y_i), i = 1, \dots, n\}$, where x_i is the one phenotype of cell i and y_i is another phenotype of the same cell (or of the other cell in the pair). The function that relates x_i and y_i is:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

Linear regressions seek to find a linear model that fits best all of the cells, minimizing ϵ_i , which is the residue of each datapoint to the line of the model.

$$y = \alpha + \beta x$$

The offset (α) reflects the independent scaling difference between the variables. The slope (β) reflects the relation of proportion between variables and often provides the most interesting information about their relationship. For example, a positive slope indicates a directly proportional relationship, which means that if one of the variables increases, the other one does as well. Linear regression using least squares can be performed in Python by importing `numpy` and using the function `numpy.linalg.lstsq()`, which takes x and y as parameters and returns the coefficients (including the offset) and the residuals.

Relative difference at single time points

1. Calculate relative differences (RD) as a measure for how similar or different two single-cell phenotypes are (Figure 7B).

$$RD = \frac{|x - y|}{|f(x, y)|}$$

To avoid bias due to the values themselves, the difference in the phenotype values is divided by some reference value $f(x, y)$, which, for example, can be the mean (μ) so that both cells contribute equally, and is then a similar measurement to CV (dividing the RD by 2, or rescaling the values on a scale from 0 to 1 is equivalent to calculating a CV for two values).

2. Perform statistical tests to confirm actual differences between groups.

Method C: Comparing trajectories of cells over time

With time-lapse data, correlations can be calculated over multiple time points, e.g., to compare a single cell to itself during two inductions or using the lineage information for analyzing inheritance from pairs of mother and daughter cells.

1. As previously, plot the phenotypes as a scatter plot against each other, with one observation on the x axis and the other along the y axis, and now use a color scale for the time (Figure 8A) or make a 3D plot where the z axis is time and each cell is plotted with a different color/symbol (Figure 8B). These figures can be generated in MATLAB using the function for plotting in 3D axis `scatter3D()`; an example of this application is provided within `scatter3d_Compare2intervals.m` under the directory `PHYLOCELLcomplement` on Github (<https://github.com/aguiar-gomez/SingleCellLineageAnalysis>).

Correlations and partial correlations over time

1. Calculate correlations or partial correlations for each cell to itself over two time periods (e.g., two inductions) or pairs of cells (mothers and daughters) over their time trajectories to obtain a distribution of correlations.
2. Plot as e.g., boxplots or violinplots (Figures 8C and 8D).
3. Perform statistical tests to confirm actual differences between groups.

Note: When the phenotype of all of the cells follows the same trend through the experiment (such as steadily increasing gene expression during an induction), this results in extremely high values of correlation as the overall trend plays a greater role in determining similarity than other underlying variables (e.g., relatedness, age, cell cycle, etc.) (Figure 8C). To minimize this effect, calculate partial correlations (Figure 8D).

Partial correlation measures the degree of association (strength and direction) between two continuous variables, while removing the effect of a third controlling variable. In order to subtract this effect, calculate an average of the phenotype in the population through time, and use it as the controlling variable. For each of the continuous variables that you want to compare, perform a linear regression to the controlling variable and obtain the residuals.

For a cell x_k , the linear regression (a_k) is calculated by testing models (a) where prediction (which is calculated as the scalar product between the model and the population average \bar{x}), is most similar to x_k , meaning the residues of the subtraction are minimized:

$$a_k = \operatorname{argmin} \left\{ \sum_{i=1}^N (x_i^k - (a \cdot \bar{x}))^2 \right\}$$

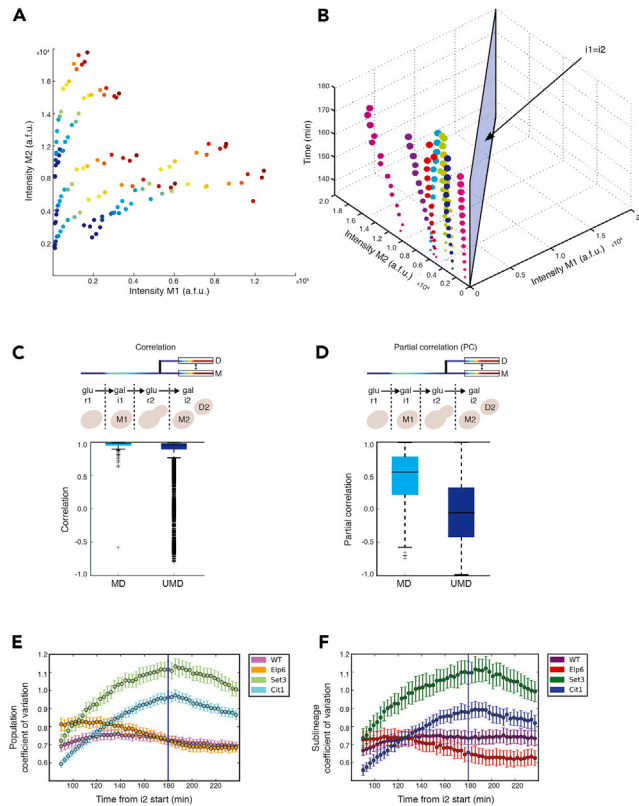


Figure 8. Analysis figures option C: comparing trajectories of cells over time

(A and B) (A) Single-cell phenotypes plotted on the x axis and y axis, and time with color scale or (B) time on the z axis. Each dot represents a single cell at a given time point. (A) and (B) were generated in MATLAB, using the functions `plot(x,y)` and `scatter3()`, respectively.

(C) Cells undergoing similar phenotype changes (e.g., induction) over time will have high correlation, regardless of whether they are related (MD = mother-daughter pairs) or unrelated (UMD = unrelated mother-daughter pairs). Boxplot elements are as follows: center line, median; diamond, mean; box limits, upper and lower quartiles; whiskers, $1.5 \times$ interquartile range; crosses, outliers.

(D) Same dataset as in (C), except calculated with partial correlation. Partial correlation helps to differentiate underlying sources of similarity by subtracting the population effect. The boxplots of (C) and (D) were generated in Python using the Matplotlib function `matplotlib.pyplot.boxplot()`.

(E and F) (E) Comparing different strains for coefficient of variation over time for the population of cells at each time point and (F) within sublineages, which is defined by the lineage of each mother cell at any given time point. Error bars in E and F are calculated from bootstrapping ($\times 1,000$) and they are drawn using the standard deviation of the bootstrap samples. Error bars in (E) and (F) were plotted using the `errorbar()` function in Matplotlib.

Next, the residue, which represents the part of the phenotype that cannot be explained by the population effect, is calculated for each cell:

$$r_k = x_k - (a_k \cdot \bar{x})$$

Finally, Pearson correlation coefficient is calculated using the residues of two cells:

$$\text{Partial correlation}(r_1, r_2) = \frac{\text{cov}(r_1, r_2)}{\sigma_{r_1} \sigma_{r_2}}$$

Note: The analysis implemented in our previous work (Bheda et al., 2020a) is available on Github (<https://github.com/aguiar-gomez/SingleCellLineageAnalysis>) under the directory *Partial Correlation* which contains a jupyter notebook (PartialCorrelations.ipynb) with the function `partial_corr()` used to calculate partial correlations. This function takes three arguments which all must be vectors of the same size, from which two vectors are used for the correlation and a

third used as a controlling variable. This analysis returns the Pearson correlation coefficient of the residuals.

Coefficient of variation over time

1. Calculate coefficients of variation for all cells within the population at each time point to determine how the noise in expression in the population is changing over time (Figure 8E).
2. For data with lineage information, calculate coefficients of variation between sublineages of cells (SLCV). Compare with the population CV to determine whether the population is differentiating on a single-cell level or on a lineage level (Figure 8F) (Ni et al., 2012).

$$SLCV(s, t) = \frac{\sqrt{\frac{N_t}{N_s}} \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} (\bar{x}_i - \mu_s)^2}}{\mu_s}$$

Where s and t are time points, and $s < t$. The second time point t , should be enough time after s to allow cells to divide at least once. N_s is the number of cells at time s , which are defined as mother cells, and each mother cell will divide and have its own sublineage. Therefore, N_s is the number of cells at time s and also the number of sublineages at time t . At time t , N_t is the total number of cells N_s plus their progeny. The phenotype for each cell is denoted as x_i^k ($i = 1, N_s$; $k = 1, n_i$), where i indicates the sublineage of a cell and k the number of cells within that sublineage. Then, \bar{x}_i is the average phenotype among cells of a sublineage and μ_s is the average across all \bar{x}_i .

3. Bootstrap cells in each time point to create a distribution of CVs and to calculate error bars by the standard error of the mean (SEM) or the standard deviation (STD).

Note: A Python notebook is available on Github with an example on how to perform this analysis (https://github.com/aguiar-gomez/SingleCellLineageAnalysis/blob/master/Sublineage_CV/SLCV.ipynb).

Method D: Asymmetric division of phenotypes

To probe asymmetric inheritance of a phenotype, there are methods to determine whether mother and daughter cells behave equally, or if there is a bias toward one or the other.

Binomial test

1. Perform a binomial test as a simple method to determine potential asymmetry of memory inheritance.

Transform the data into a categorical variable by counting the cases k out of the total number of pairs of related cells N (mother-daughter relationships) where the mother has higher/lower values than its daughter. Then, a binomial test can be performed to test the probability (P) that the trait is symmetrically inherited ($p = 50\%$) or higher in mothers or daughters.

$$P = \frac{N!}{k!(N-k)!} p^k (1-p)^{n-k}$$

Note: R base has the function `biom.test(k, N, p)` to perform the test. An example of this analysis is provided on Github (<https://github.com/aguiar-gomez/SingleCellLineageAnalysis>) within the script `AsymmetricDivision.R`.

Bayesian statistics

1. Assess asymmetric inheritance of a trait with Bayesian Statistics (Figure 9). For this you can calculate the posterior distribution $P(p|k, N)$ where p is the probability that the mother has

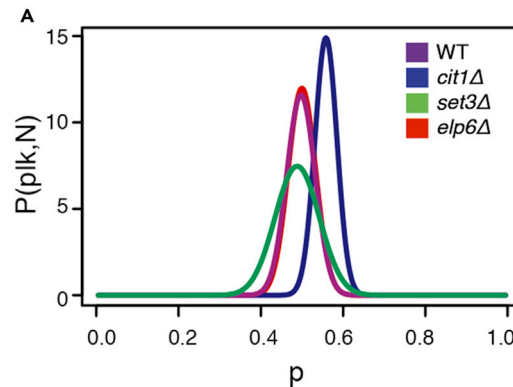


Figure 9. Analysis figures option D: asymmetric division of phenotypes by posterior distributions

Bayesian statistics can be used to determine the probability that mothers or daughters have a certain phenotype. For symmetric inheritance, this will be close to 0.5. It is also possible, for example, to compare the probability that a mutant strain has a different inheritance pattern than a WT. In [Bheda et al. \(2020a\)](#), we implemented these calculations to determine whether there was a bias in mother cells expressing Gal1-GFP before their daughter cells and identified a mutant (*cit1Δ*) with asymmetric inheritance. The x axis shows the probability of mother cells expressing faster than their daughters (p). The y axis shows the posterior probability $P(p|k, N)$, which is the probability of asymmetric inheritance (p) given the observed mother cells that started expressing earlier (k) out of the total pair of cells analyzed (N). This figure was plotted in R using `plot()` and `lines()`.

a certain phenotype (e.g., higher expression or faster induction) than its daughter given the data (k, N):

$$P(p|k, N) = \frac{L(k|p, N) \cdot \pi(p)}{\int L(k|p, N) \cdot \pi(p) dp}$$

Where L is a binomial likelihood, and $\pi(p)$ is a Beta (1, 1) flat prior distribution.

By integrating the distribution in a range $[a, b]$ you can calculate the probability of having a value anywhere in that range. To determine the probability that the mothers express higher more than 50% of the time, calculate the posterior probability that p is larger than 0.5 by integrating $P(p|k, N)$ on the interval $[0.5, 1]$. A probability outcome of 0.5 would then indicate that mothers or daughters express equally ([Figure 9](#)), where CDF is the cumulative distribution function.

$$\begin{aligned} \int_{0.5}^1 P(p|k, N) dp &= \\ 1 - \int_0^{0.5} P(p|k, N) dp &= \\ &= 1 - \text{CDF}(0.5) \end{aligned}$$

To compare strains/conditions to a control/WT, calculate the probability that p of the mutant is larger than p of the wild type (taking into account posterior uncertainty) with $\Delta x = x_{i+1} - x_i$ as the step size.

$$\begin{aligned} \int P_{WT}(p = x) \cdot P_{mutant}(p > x) dx &= \\ \int P_{WT}(p = x) \cdot (1 - \text{CDF}_{mutant}(x)) dx &= \\ \approx \Delta x \sum_{i=0}^N P_{WT}(p = x_i) \cdot (1 - \text{CDF}_{mutant}(x_i)) & \end{aligned}$$

Where P_{WT} and P_{mutant} are the posterior distributions of the WT and the mutant strains, respectively.

Note: The code is available within the *AsymmetricDivision.R* script mentioned above. The function *dbeta()* in R is used to calculate and plot the distributions, and the function *pbeta()* is used to calculate probability values against an expected outcome.

Method E: Assessing statistical significance

Here we suggest a few tests and considerations for data from the protocol and analyses above to compare different strains/conditions/samples and determine whether they are statistically significant from each other.

Sample size

1. Ensure that the number of cells does not differ greatly between the groups and that there are not too many observations in each group.

Note: For comparisons from cell-tracking microfluidics experiments, we have compared ~30–300 single cells/pairs of cells from each group.

Note: Large numbers of data can affect the p-value in a way such that everything appears to be statistically significant, resulting in false positive differences due to e.g., technical variability between experiments and not true biological differences. If you are interested in identifying differences in strains or conditions, one way to account for this is to include multiple technical and biological replicates of a reference (usually WT) strain/condition. This also highlights the importance of considering the effect sizes (Sullivan and Feinn, 2012). For more on this see below.

Control samples

2. Create a control sample to compare related cells by reshuffling pairs of cells (same cell at different time points or in different conditions, or mothers vs daughters) into unrelated pairs to differentiate between population effects and effects on maintenance/inheritance of phenotypes.

Note: Related and unrelated pairs may behave similarly if the phenotype is not inherited or there is not variability in gene expression in the population.

Note: Reshuffling to form unrelated pairs results in a number of combinations that is much greater than the number of related cells. If the total number of cells in the population is N , then we have $\frac{N(N-1)}{2}$ possible pairs. Then if we have x related pairs of cells, we would have $\frac{N(N-1)}{2} - x$ unrelated pairs of cells. To keep sample sizes similar, you can sample the unrelated pairs of cells by randomly selecting the same number of pairs that you have for the related cells.

Test for normality

3. Determine whether your data are normally distributed.

Note: You can use graphical methods like the quantile-quantile plot or a histogram of your data and check whether the shape of the histogram looks like a normal probability curve. However, if you want a formal method you can use a one-sample Kolmogorov-Smirnov test, the Shapiro-Wilk test, or the Anderson-Darling test.

Comparing two samples

4. To determine whether two samples have been drawn from the same distribution you can use the common t-test if your data are continuous variables, normally distributed, and homogenous in variance. If the data are not normally distributed, then non-parametric statistical tests, such as the Mann-Whitney U test or a two-sample Kolmogorov-Smirnov test, are recommended.

Note: All these tests are available in multiple software packages, here we just list a few: R base has the following functions for t-test, Mann-Whitney U (also known as Wilcoxon rank sum test) and Kolmogorov-Smirnov: `t.test()`, `wilcox.test()`, `ks.test()`. In Python the `scipy.stats` module contains `ttest_ind()`, `mannwhitneyu()`, `ks_2samp()` for the equivalent tests. In all of the tests the null hypothesis is that the samples have been drawn from the same distribution. The alternative hypothesis default is “two-sided,” but there is a parameter that can be changed to “greater” or “lower” if you want to test for that specifically. For comparing more than two samples ANOVA can be used, or several other pairwise tests can be performed.

Effect size

5. Consider the size of the effect and whether it is large enough for the effect to be considered meaningful. The effect size is the magnitude of the difference between groups.

Note: The magnitude that would be meaningful requires some domain knowledge of the specific question and the units of the phenotype that is being measured. There are different ways of assessing the effect size. One of the simplest ways is to subtract the means of the different groups and divide by the standard deviation of either group (Cohen’s d). The effect size with this calculation is considered roughly in the following way: small-0.2, medium-0.5, large-0.8, very large-1.3. One of the advantages of the two-sample Kolmogorov-Smirnov tests is that besides calculating a p-value, the tests quantifies a distance between samples (D statistic).

Correcting p values for multiple testing

6. Correct p values for multiple testing when comparing multiple strains/conditions/samples, as the incidence of a falsely significant p-value increases with each test.

Note: There are several methods for multiple correction testing. It is recommended to choose a method of correction that does not favor the results toward a desired conclusion. The Bonferroni correction method can be easily implemented by multiplying the original p values by the number of tests performed. This is the most conservative correction method. Another popular and less stringent correction method is False Discovery Rate (FDR). FDR has greater power than Bonferroni, which means it is better at avoiding type II errors (false negatives). The power comes at a cost of increased number of type I errors (false positives) compared to Bonferroni. R base contains the function `p.adjust()` where the parameter `method` is used to specify which method (Bonferroni, FDR, etc) of correction is desired. Python has a module called `statsmodels.stats.multitest` with a function `multipletests()` that works the same way, having a parameter to specify the correction method.

LIMITATIONS

The microfluidics chip used in [Bheda et al. \(2020a\)](#) is suitable for tracking lineages of cells, but is limited by a maximum number of cells before cell tracking becomes very difficult and prone to numerous errors. If the time lapse is started with ~2 cells in the field of interest, then all cells in ~8 generations can be accounted for and reasonably tracked with 3-min image resolution.

Depending on the question of interest, you may want to use fluorescence reporters as in [Bheda et al. \(2020a\)](#), although there are reports of cell-tracking data using microfluidics where fluorescence reporters are not used and, for example, they rather follow individual cell or colony growth ([Cerulus et al., 2018](#)).

If you employ fluorescence reporters, it is possible that prolonged light exposure from taking multiple phase contrast and fluorescence images with high time resolution could affect the fitness of the cells.

To determine whether this is a problem in your experiments, you can compare the well-being of the cells in your cell-tracking microfluidics experiments to yeast that have not been exposed to excitation light, e.g., subjected to only phase contrast images taken every 15 min to 1 h. For this, you can examine whether there are any noticeable phenotypic differences, e.g., increase in the number of vacuoles or cell death. For a more quantitative test, you can also calculate and compare growth rates for the cells with and without fluorescence excitation ([Schmidt et al., 2018](#)). If you are comparing your data between a wild-type strain and other conditions or mutants, and your growth and imaging conditions are the same, the light incidence is also identical for all strains, and therefore it would be unlikely that your results are affected by the imaging procedure. Additionally, you may want to validate your data with other methods, such as testing bulk expression by RT-qPCR analyses.

TROUBLESHOOTING

Problem

Too many cells are captured in each microchannel at the start of the microfluidics experiment.

Potential solution

Cell cultures that are in low log phase do not need to be diluted prior to injecting into a microchamber. However, if too many cells are being captured in the microchannels, it may help to dilute the cultures in the same media prior to the injection. Additionally, captured cells can be flushed out by injecting more culture with a slightly harder push on the syringe.

Problem

Cells are lost/washed out or images lose focus during the microfluidics experiment.

Potential solution

Flow rates that are too high can cause cells to be lost, especially those growing at the periphery of the colony. For this reason, we recommend using a flow rate between 10–50 μL . Problems with high pressure can also result in sudden, small delaminations of the PDMS chip from the glass coverslip. Small changes in the height of the microfluidics channels can cause cells to be flushed away or imaging to lose focus. This could occur due to clogging of the microchamber/microchannel with dust particles/debris from the media. To eliminate this issue, we recommend filtering the media with a 0.22 μm filter just prior to use, such that particulate matter that could clog the channels are removed.

Problem

There are too many cell mapping errors toward the end of the experiment, making manual correction of cell identities in the segmentation files very time-consuming.

Potential solution

Long cell-tracking microfluidics experiments face the problem that toward the end of the experiment, there are many cells that are all growing and dividing and pushing each other. This makes it difficult for any mapping algorithm to function properly, as they use a limited cellular displacement to aid in mapping. If this is a problem in your experiments, start with fewer cells in each microchannel at the beginning of the experiment.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Poonam Bheda (poonam.bheda@helmholtz-muenchen.de, poonam.j.bheda@gmail.com).

Materials availability

The reagents generated in this study are available upon request from the Lead Contact, Poonam Bheda (poonam.bheda@helmholtz-muenchen.de, poonam.j.bheda@gmail.com). The CAD file for the microfluidics chip design used in Bheda et al. (2020a) is available at Mendeley Data (<https://data.mendeley.com/datasets/yr2nrysyyx/1>) (Bheda et al., 2020b).

Data and code availability

Analysis scripts for cell-tracking microfluidics are available on Github (<https://github.com/aguiar-gomez/SingleCellLineageAnalysis>). On this Github we have some sample datasets available, in the directory *Partial Correlations*, the *fileMotherDaughterWT.txt* contains single-cell data over time for a second galactose induction of yeast cells and includes lineage information (mother cell specified). The directories *Sublineage_CV* and *TimeSeriesPlot* also contain sample datasets of single-cell data over two galactose inductions of the WT strain and unspecified mutants (labeled Mutant1 and Mutant2). Additional sample datasets are available from the authors upon request. PhyloCell and Autotrack software are available on Github (<https://github.com/gcharvin>).

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at <https://doi.org/10.1016/j.xpro.2020.100228>.

ACKNOWLEDGMENTS

We thank Theo Aspert and Youlian Goulev for support on the cell-tracking microfluidics setup. Work in the R.S. laboratory was supported by the DFG through SFB 1064 (project-ID 213249687) and SFB 1309 (project-ID 325871075), Marie Skłodowska-Curie Innovative Training Network (ITN) project "Cell2Cell" (H2020-MSCA-ITN-20219 grant no. 860675), as well as AmPro program (ZT0026) and the Helmholtz Gesellschaft. The A.K. laboratory is co-funded by the European Regional Development Fund and the Republic of Cyprus through the Research & Innovation Foundation (projects: EXCELLENCE/1216/0215 and EXCELLENCE/0918/0105). G.C. was supported by grant no. ANR-10-LABX-0030-INRT, a French State fund managed by the Agence Nationale de la Recherche under the frame program Investissements d'Avenir ANR-10-IDEX-0002-02.

AUTHOR CONTRIBUTIONS

P.B., A.K., and R.S. conceived the study. P.B., D.A.G., and I.K. performed the cell-tracking microfluidics time-lapse experiments and processed the data. G.C. developed the cell-tracking microfluidics and the cell-tracking image analysis pipeline. J.B. identified memory mutant candidates. P.B. and D.A.G. developed and analyzed the cell-tracking microfluidics data for inheritance analyses. P.B. and D.A.G. wrote the manuscript. I.K., A.K., and R.S. critically revised the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Bheda, P., Aguilar-Gomez, D., Becker, N.B., Becker, J., Stavrou, E., Kukhtevich, I., Hofer, T., Maerkl, S., Charvin, G., Marr, C., et al. (2020a). Single-cell tracing dissects regulation of maintenance and inheritance of transcriptional reinduction memory. *Mol. Cell* 78, 915–925.e17.

Bheda, P., Charvin, G., Aguilar-Gomez, D., Kukhtevich, I., Becker, J., Kirmizis, A., and Schneider, R. (2020b). Microfluidics Chip_16 Chamber Design (Mendeley Data), p. V1, <https://doi.org/10.17632/yr2nrysyyx.1>.

Brachmann, C.B., Davies, A., Cost, G.J., Caputo, E., Li, J., Hieter, P., and Boeke, J.D. (1998). Designer deletion strains derived from *Saccharomyces cerevisiae* S288C: a useful set of strains and plasmids for PCR-mediated gene disruption and other applications. *Yeast* 14, 115–132.

- Cerulus, B., Jariani, A., Perez-Samper, G., Vermeersch, L., Pietsch, J.M., Crane, M.M., New, A.M., Gallone, B., Roncoroni, M., Dzialo, M.C., et al. (2018). Transition between fermentation and respiration determines history-dependent behavior in fluctuating carbon sources. *Elife* 7, e39234.
- Charvin, G., Cross, F.R., and Siggia, E.D. (2008). A microfluidic device for temporally controlled gene expression and long-term fluorescent imaging in unperturbed dividing yeast cells. *PLoS One* 3, e1468.
- Fehrmann, S., Paoletti, C., Goulev, Y., Ungureanu, A., Aguilaniu, H., and Charvin, G. (2013). Aging yeast cells undergo a sharp entry into senescence unrelated to the loss of mitochondrial membrane potential. *Cell Rep.* 5, 1589–1599.
- Garmendia-Torres, C., Tassy, O., Matifas, A., Molina, N., and Charvin, G. (2018). Multiple inputs ensure yeast cell size homeostasis during cell cycle progression. *Elife* 7.
- Gietz, R.D., and Woods, R.A. (2002). Transformation of yeast by lithium acetate/single-stranded carrier DNA/polyethylene glycol method. *Methods Enzymol.* 350, 87–96.
- Goulev, Y., Morlot, S., Matifas, A., Huang, B., Molin, M., Toledano, M.B., and Charvin, G. (2017). Nonlinear feedback drives homeostatic plasticity in H₂O₂ stress response. *Elife* 6.
- Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array programming with NumPy. *Nature* 585, 357–362.
- Hunter, J.D. (2007). Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* 9, 90–95.
- Kukhtevich, I.V., Lohrberg, N., Padovani, F., Schneider, R., and Schmoller, K.M. (2020). Cell size sets the diameter of the budding yeast contractile ring. *Nat. Commun.* 11, 2952.
- McKinney, W. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference* 445, 51–56.
- Meitinger, F., Khmelinskii, A., Morlot, S., Kurtulmus, B., Palani, S., Andres-Pons, A., Hub, B., Knop, M., Charvin, G., and Pereira, G. (2014). A memory system of negative polarity cues prevents replicative aging. *Cell* 159, 1056–1069.
- Morlot, S., Song, J., Leger-Silvestre, I., Matifas, A., Gadal, O., and Charvin, G. (2019). Excessive rDNA transcription drives the disruption in nuclear homeostasis during entry into senescence in budding yeast. *Cell Rep.* 28, 408–422.e4.
- Ni, M., Decrulle, A.L., Fontaine, F., Demarez, A., Taddei, F., and Lindner, A.B. (2012). Pre-disposition and epigenetics govern variation in bacterial survival upon stress. *PLoS Genet.* 8, e1003148.
- Paoletti, C., Quintin, S., Matifas, A., and Charvin, G. (2016). Kinetics of formation and asymmetrical distribution of Hsp104-bound protein aggregates in yeast. *Biophys. J.* 110, 1605–1614.
- Pattengale, N.D., Alipour, M., Bininda-Emonds, O.R., Moret, B.M., and Stamatakis, A. (2010). How many bootstrap replicates are necessary? *J. Comput. Biol.* 17, 337–354.
- Schmidt, G.W., Cuny, A.P., and Rudolf, F. (2018). Preventing photomorbidity in long-term multi-color fluorescence imaging of *S. cerevisiae* and *S. pombe*. *bioRxiv*. <https://doi.org/10.1101/180018>.
- Sullivan, G.M., and Feinn, R. (2012). Using effect size-or why the P value is not enough. *J. Grad. Med. Educ.* 4, 279–282.
- Tong, A.H., and Boone, C. (2007). High-throughput strain construction and systematic synthetic lethal screening in *Saccharomyces cerevisiae*. *Methods Microbiol.* 36, 369–386, 706–707.
- Versari, C., Stoma, S., Batmanov, K., Llamasi, A., Mroz, F., Kaczmarek, A., Deyell, M., Lhoussaine, C., Hersen, P., and Batt, G. (2017). Long-term tracking of budding yeast cells in brightfield microscopy: CellStar and the evaluation platform. *J. R. Soc. Interface* 14.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* 17, 261–272.
- Wickham, H. (2011). ggplot2. *Wiley Interdiscip. Rev. Comput. Stat.* 3, 180–185.
- Wilcox, R.R. (2010). *Fundamentals of modern statistical methods: substantially improving power and accuracy*, Second Edition (Springer).
- Xu, Z., Fallet, E., Paoletti, C., Fehrmann, S., Charvin, G., and Teixeira, M.T. (2015). Two routes to senescence revealed by real-time analysis of telomerase-negative single lineages. *Nat. Commun.* 6, 7680.
- Waskom, M., Botvinnik, O., Hobson, P., Cole, J.B., Halchenko, Y., Hoyer, S., Miles, A., Augspurger, T., Yarkoni, T., Megies, T., et al. (2014). seaborn: v0.5.0. Zenodo, <https://doi.org/10.5281/zenodo.592845>.