

## 1 What the plotting script does

1. Downloads *Passaging* and *Media* records from the **CLONEID** database
2. Reconstructs each seed → harvest lineage you care about
3. Fits an exponential (log-linear) growth model to every lineage
4. Optionally produces three kinds of plots
  - Growth-curve panels
  - Generation-rate bar plots
  - Passage-vs-growth overview

Everything is configured in **one JSON file** called `parameters.json`.

---

## 2 Before you start

Item	What to do
<b>R ≥ 4.2</b>	Install from <a href="https://cran.r-project.org">https://cran.r-project.org</a> .
<b>Packages (in addition to CLONEID)</b>	<code>install.packages(c("RMySQL", "jsonlite", "ggplot2", "stringr", "dplyr", "patchwork", "rlang", "viridis"))</code>
<b>Database</b>	Make sure you can log in to the <i>CLONEID</i> DB.
<b>db_creds.txt</b>	Holds database credentials, see the template in § 4
<b>parameters.json</b>	Holds all the parameters to control plotting functions, see the template in § 4

---

## 3 Running the script

What happens under the hood (you do not need to edit this):

1. `get_script_path()` sets the working directory to the script's folder.
  2. `load_db_vars()` checks that *db\_creds.txt* exists and is valid.
  3. The script opens *parameters.json*.
  4. It connects to MySQL using your credentials.
  5. Two queries pull the *Passaging* and *Media* tables.
  6. For each lineage you defined, the script walks from the **first** seed ID to the **last** and builds a tidy data frame.
  7. A simple `lm()` fit (`log(cell_count) ~ time`) gives intercept + growth-rate *g*.
  8. Plots are drawn **only** if you flipped the corresponding switches to `true`.
  9. The database connection closes automatically (unless you set `disconnect_on_completion : false`).
- 

## 4 Input files

**Database credentials: db\_creds.txt**

```
,
USER=
PASSWORD=
HOST=cloneid.cswgogbb5ufg.us-east-1.rds.amazonaws.com
DBNAME=CLONEID
,
```

## Plotting parameter file: parameters.json

Below is a fully commented template.

Leave all comments beginning with `_comment` unchanged.

```
{
  "_comment"          : "Edit the UPPER-CASE fields; leave the rest.",
  "database" : {
    "_comment"          : "Default queries - 'SELECT *' so no SQL knowledge needed.",
    "queries" : {
      "passaging_query" : "SELECT * FROM Passaging",
      "media_query"      : "SELECT * FROM Media"
    }
  },
  "filters" : [
    {
      "_comment" : "Each object = one lineage you want to analyse.",
      "ranges" : [
        { "first" : "FIRST_SEED_ID",      "last" : "LAST_SEED_ID",
          "label" : "GROUP_NAME",          "label2" : "SUBGROUP" }
      ]
    }
  ],
  "manual_filters" : [
    { "_comment" : "Optional - lines you want to hide.",
      "key"      : "id",
      "values"   : ["BAD_SAMPLE_1", "BAD_SAMPLE_2"] }
  ],
  "plotting" : {
    "_comment"          : "Toggle the three plot types and set labels.",
    "make_growth_curves" : true,
    "make_generation_barplot" : true,
    "make_passage_vs_growth" : false,

    "color_by"          : "media",    // Any column name (e.g. 'EnergySource_nM')
    "output_path"       : "./cloneid_plots.png",

    "axis_labels" : {
      "x_label"          : "Days after seeding",
      "y_label_cellcount" : "Cell count",
      "y_label_growth"    : "Growth rate (day-1)",
      "passage_label"     : "Passage ID"
    },

    "titles" : {
      "passage_vs_growth" : "Growth rate across passages"
    },

    "facet_grid" : {
      "_comment" : "'rows' and 'cols' decide the panel layout.",
      "rows"     : "label_value",    // typically group name
    }
  }
}
```

```

    "cols"      : "adjPass",
    "scales"    : "free"           // 'fixed' to keep identical axes
  },

  "database_settings" : {
    "disconnect_on_completion" : true
  }
}

```

## What each section means

Section	Why it matters
database → queries	These two <i>already</i> grab everything you need. Keep them as-is
filters → ranges	Think of <b>first</b> & <b>last</b> as bookends of a timeline. The script fills in every passage between them. Use <b>label</b> and <b>label2</b> for specifying experimental conditions and/or replicate structure in plot captions.
manual_filters	Sometimes you know a sample is bad. List its ID here and it will vanish from plots <i>and</i> stats.
plotting	Flip any of the three <code>make_*</code> switches to <code>false</code> if you are not interested in that plot. Everything else is cosmetic (colors, titles, axis text).
database_settings	Leave at <code>true</code> so R closes the database nicely when finished.

## 5 Plot types

Level	Turn on	Typical use case
<b>Growth curves</b>	<code>make_growth_curves = true</code> ( <i>the other two false</i> )	One lineage, see its scatter plot + fitted line.
<b>Multi-condition curves</b>	Same as above, but add several <code>ranges</code> and set <code>facet_grid rows = "sublabel_value"</code>	Compare control vs treatment on separate rows.
<b>Generation bar-plot</b>	<code>make_generation_barplot = true</code>	Show estimated <i>g</i> (height of bars) for each seed replicate.
<b>Passage vs growth</b>	<code>make_passage_vs_growth = true</code>	Check if there is a trend in growth rate over time.

## 6 Troubleshooting

Error message / symptom	What it usually means	How to fix
Could not trace ID...	Wrong <b>first</b> or <b>last</b> in <code>ranges</code> .	Double-check the IDs in Sequel Ace.
Empty plots	You filtered out all the rows (see <code>manual_filters</code> ).	Set <code>manual_filters</code> to an empty list <code>[]</code> and try again.
No color legend	<code>color_by</code> misspelled.	Use <code>names(df)</code> inside the script to see valid column names.
“Access denied”	Incorrect login in <code>db_creds.txt</code> .	Verify HOST / USER / PASSWORD with your DBA.