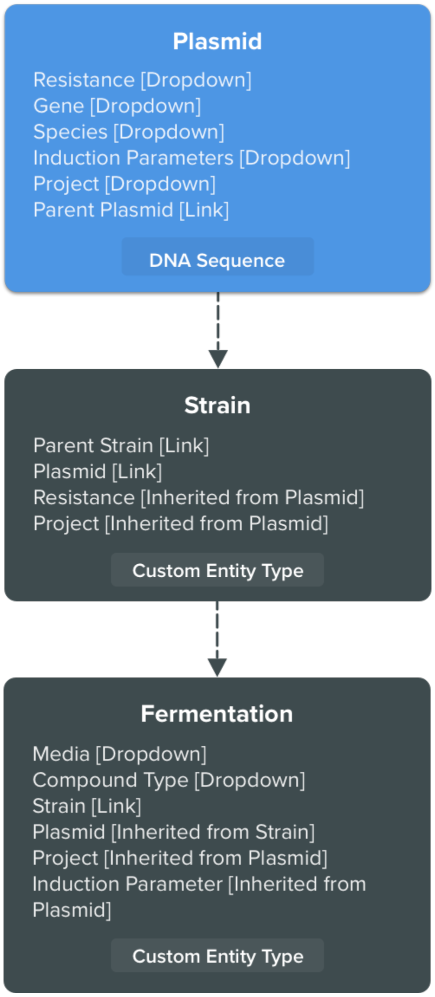
Benchling

# How to Configure a Bioregistry for Strain Engineering

From fermentation runs to their upstream strains and plasmids, strain engineering involves many entities and parameters that have to be digitally modeled and interlinked. Without a robust registration system that's configured to track and interlink all of these pieces, you end up with data that's inaccurate, redundant, or insufficient. But if you configure your Bioregistration system properly, you can answer pretty much any question about your large molecules. With these answers at your fingertips, you can drive research decisions with data and results that are reliable and comprehensive.

Coming up with a final data model for your strain engineering ultimately depends on your particular research needs, but here's how to use Benchling's point-and-click interface to configure a **Foundational Bioregistry for Strain Engineering**.

TABLE OF CONTENTS



**Plasmid**

Resistance [Dropdown]
Gene [Dropdown]
Species [Dropdown]
Induction Parameters [Dropdown]
Project [Dropdown]
Parent Plasmid [Link]

DNA Sequence

**Strain**

Parent Strain [Link]
Plasmid [Link]
Resistance [Inherited from Plasmid]
Project [Inherited from Plasmid]

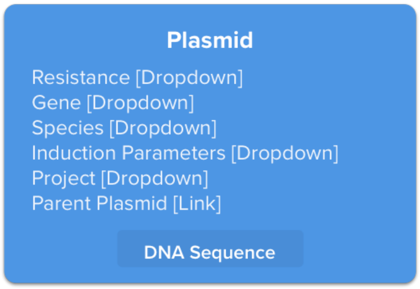
Custom Entity Type

**Fermentation**

Media [Dropdown]
Compound Type [Dropdown]
Strain [Link]
Plasmid [Inherited from Strain]
Project [Inherited from Plasmid]
Induction Parameter [Inherited from Plasmid]

Custom Entity Type

# The Plasmid



Plasmid

Resistance [Dropdown]
Gene [Dropdown]
Species [Dropdown]
Induction Parameters [Dropdown]
Project [Dropdown]
Parent Plasmid [Link]

DNA Sequence

If you're configuring a Bioregistry for strain engineering, plasmids will be a fundamental part of your registration needs. The first question is: What do you want to track on your **Plasmid** entity?
For starters, every plasmid should have a sequence associated with it, so we'll define the Plasmid entity overall as a "DNA sequence". This means that a **Plasmid** will have a DNA sequence, as well as additional metadata fields that we'll define next. (In Benchling, you can visualize the sequence of any registered plasmid directly in our Molecular Biology Suite.)

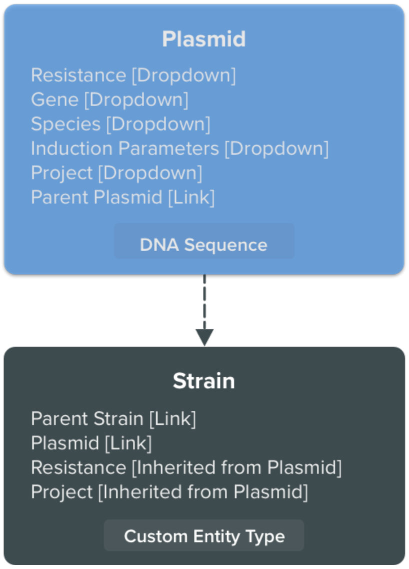
Beyond the sequence, the data to track on a **Plasmid** can vary significantly across companies, depending on the ultimate goal of their research. For our purposes, we'll keep things simple by adding fields for "Resistance", "Gene", and "Species". These are all fundamental fields that will help us get a basic sense of a **Plasmid's** characteristics. To maintain data integrity and searchability, we recommend giving these fields a dropdown data type. That way, scientists don't have to worry about naming conventions and which values they have at their disposal.

Next, we'll add a field for "Induction Parameters", which we also recommend structuring as a dropdown. This will make the precise conditions behind each **Plasmid's** induction explicit, and it will prove useful when analyzing results across experiments.

Similarly, we suggest adding another dropdown field called "Project", which will make it easy to instantly tell the experiment where any registered entity was used. As we'll see later, this field will automatically be carried over onto entities downstream of the **Plasmid**.

Lastly, we'll add a "Parent Plasmid" field. Unlike the other fields, which have all been dropdowns, "Parent Plasmid" will be a link. Specifically, we'll define "Parent Plasmid" as a link to ***another Plasmid***. Thus on each **Plasmid** record, there will be a link to the record of its parent **Plasmid**.

# The Strain



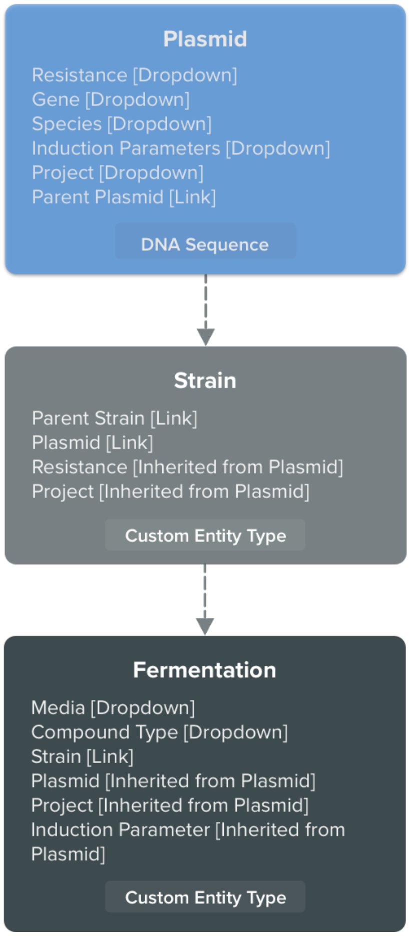
Unlike **Plasmids**, it's unnecessary for a **Strain** to have its own sequence (as you'll see, we'll be linking each **Strain** to a **Plasmid**). So instead of defining **Strains** as DNA sequences, we'll define them as a "custom entity". This merely means that **Strains** will be a collection of data fields that we configure and won't have a sequence in and of themselves.

The first field that we'll model for **Strains** is "Parent Strain". Analogous to "Parent Plasmid", this will be defined as a link to another **Strain** entity. Next, we'll define a "Plasmid" field. Similarly to "Parent Plasmid" and "Parent Strain", this will be a link field, but it will be across our two entity types. In other words, rather than be a recursive link to an entity of the same type (a la "Parent Plasmid" and "Parent Strain"), it will be a link from a **Strain** entity to a **Plasmid** entity.

For the purposes of this foundational structure, the remaining fields on **Strains** will be inherited directly from its linked **Plasmid**. In Benchling, fields can be shared in this way across linked entities, so you don't have to click through numerous related entities to find data that's relevant to an entity that's at-hand. This also has the benefit of ensuring that all your data is always up-to-date, since the values that the **Strain** inherits from the **Plasmid** will automatically update if the **Plasmid's** values change. (As we'll see later, these inherited fields can also be inherited across multiple intermediary links, even if two entities aren't themselves directly linked to one another.)

**Strains** will inherit the "Resistance" and "Project" values of their linked **Plasmid.**

# The Fermentation



**Plasmid**

Resistance [Dropdown]
Gene [Dropdown]
Species [Dropdown]
Induction Parameters [Dropdown]
Project [Dropdown]
Parent Plasmid [Link]

DNA Sequence

**Strain**

Parent Strain [Link]
Plasmid [Link]
Resistance [Inherited from Plasmid]
Project [Inherited from Plasmid]

Custom Entity Type

**Fermentation**

Media [Dropdown]
Compound Type [Dropdown]
Strain [Link]
Plasmid [Inherited from Plasmid]
Project [Inherited from Plasmid]
Induction Parameter [Inherited from Plasmid]

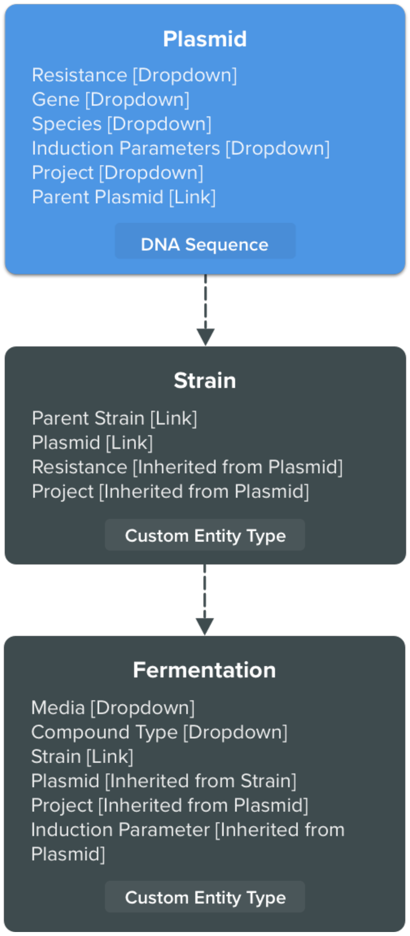
Custom Entity Type

Like **Strains**, **Fermentations** will be a “custom entity”. For its first two fields, we’ll define them as dropdowns: “Media” and “Compound Type”. (Depending on the needs of your company, there may be numerous other values that you’d like to track, and the information that we’re conveying here in a “Compound Type” dropdown might be structured as an entirely separate entity.)

We’ll then link **Fermentation** to a **Strain**. As alluded to earlier, this, in turn, gives us the ability to take data from the **Plasmid** that’s linked to the **Fermentation’s Strain** and surface it on the **Fermentation**. In other words, a **Fermentation** can inherit the data of any entity that’s upstream of it, be that data from a **Strain** (parent entity) or a **Plasmid** (grandparent).

Here, the Fermentation will inherit a link to the **Plasmid** that its **Strain**, in turn, is linked to. We’ll then pull the “Project” and “Induction Parameter” values from that **Plasmid** onto the **Strain**.

# The Big Picture:
# Answer Any Question About Your Large Molecules



**Plasmid**

Resistance [Dropdown]
Gene [Dropdown]
Species [Dropdown]
Induction Parameters [Dropdown]
Project [Dropdown]
Parent Plasmid [Link]

DNA Sequence

**Strain**

Parent Strain [Link]
Plasmid [Link]
Resistance [Inherited from Plasmid]
Project [Inherited from Plasmid]

Custom Entity Type

**Fermentation**

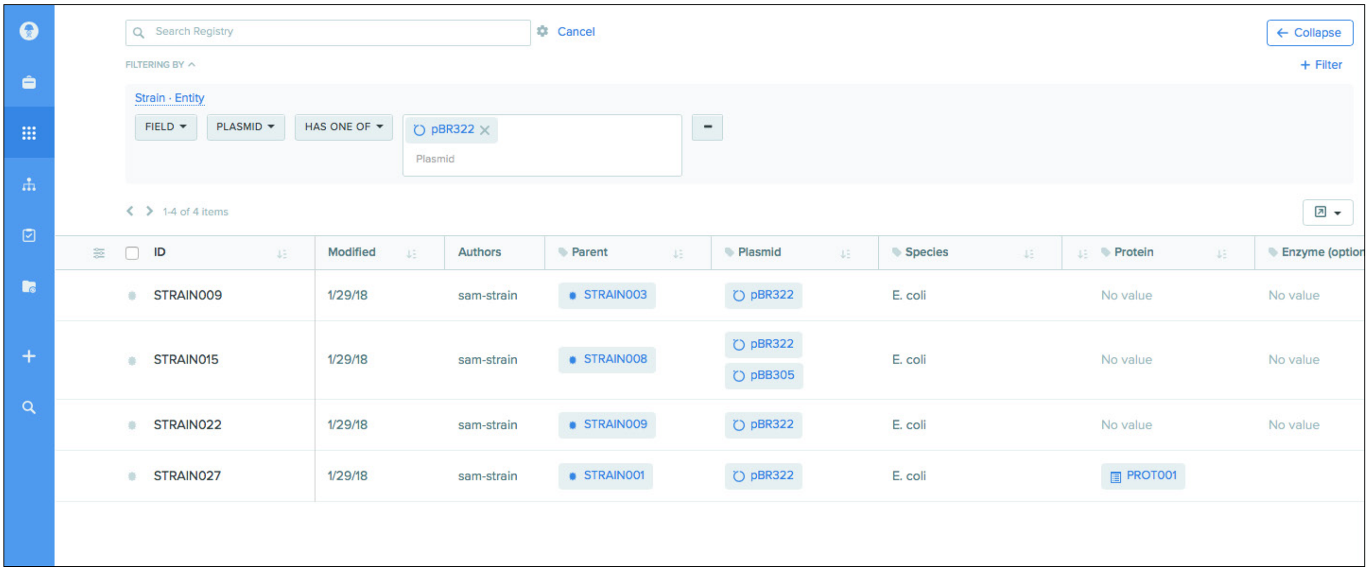
Media [Dropdown]
Compound Type [Dropdown]
Strain [Link]
Plasmid [Inherited from Strain]
Project [Inherited from Plasmid]
Induction Parameter [Inherited from Plasmid]

Custom Entity Type

In short order, we've developed an interlinked model of a basic registration system for strain engineering. Needless to say, there are numerous additional fields that you could be tracking on these entities; for example, you could break out "Gene" into its own DNA-type entity, or "Compound Type" into a "Compound" entity that could, perhaps, link into a chemistry registration system.

So now that we've created this data model, what can you do with it? Well, for one, you can run structured queries on any of the entities and fields that we just defined. Find every strain with a certain resistance. Or, find every fermentation run with a certain induction parameter. Or, of course, find all the entities produced for a certain project.

# By Tracking Biological Entities and Functional Data, You Can Make Better R&D Decisions



| ID | Modified | Authors | Parent | Plasmid | Species | Protein | Enzyme (option |
|---|---|---|---|---|---|---|---|
| STRAIN009 | 1/29/18 | sam-strain | STRAIN003 | pBR322 | E. coli | No value | No value |
| STRAIN015 | 1/29/18 | sam-strain | STRAIN008 | pBR322<br>pBB305 | E. coli | No value | No value |
| STRAIN022 | 1/29/18 | sam-strain | STRAIN009 | pBR322 | E. coli | No value | No value |
| STRAIN027 | 1/29/18 | sam-strain | STRAIN001 | pBR322 | E. coli | PROT001 | |

Beyond running these queries, though, Benchling gives you access to all of the functional data produced by these entities. You can trace from your **Strain** entities down to every physical container of them that has ever been produced. You can then answer questions such as, "For a certain strain, which induction parameters led to the most successful fermentation runs?"

The flexibility of Benchling's Bioregistry empowers you to do all of this and more, all in a point-and-click interface. The structure of your Bioregistry perfectly maps to your large molecules, while Benchling's biologically-aware features ensure that your data is always accurate. By properly structuring your Bioregistry, you can fully harness the multi-dimensional data of large molecules – from innate characteristics, to decision-quality results.

Ready to learn how Benchling maps to your R&D?
**Get a demo at benchling.com**