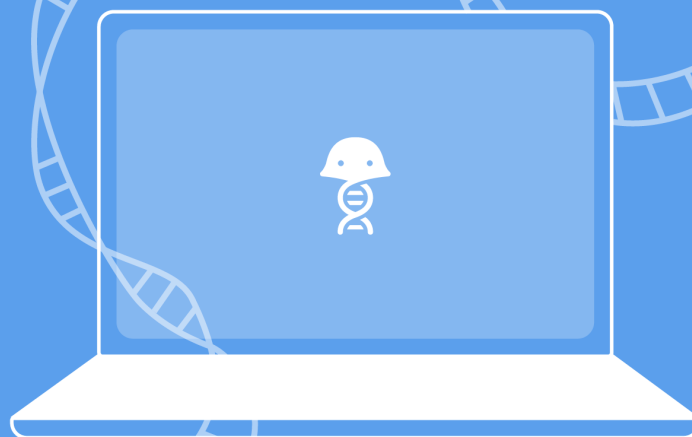# Benchling

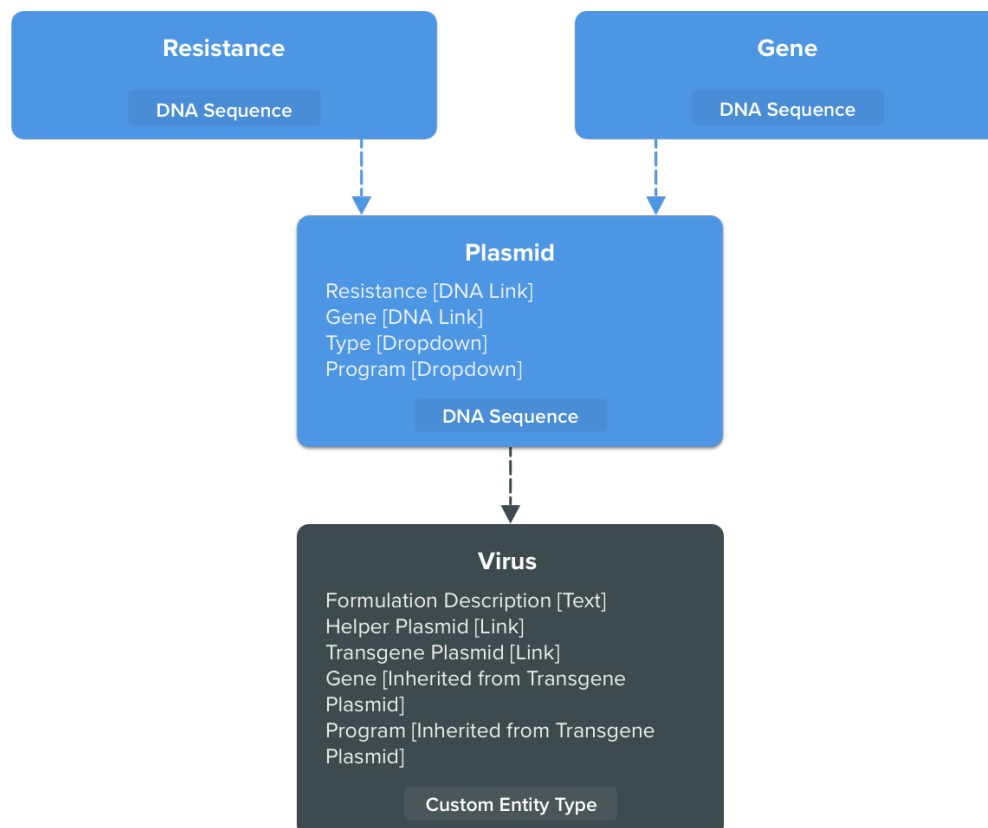## How to Configure a Bioregistry for Gene Therapy R&D

From vectors, to upstream plasmids and genes, gene therapy 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 R&D 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 Gene Therapy R&D**.

The Plasmid and Its Parts
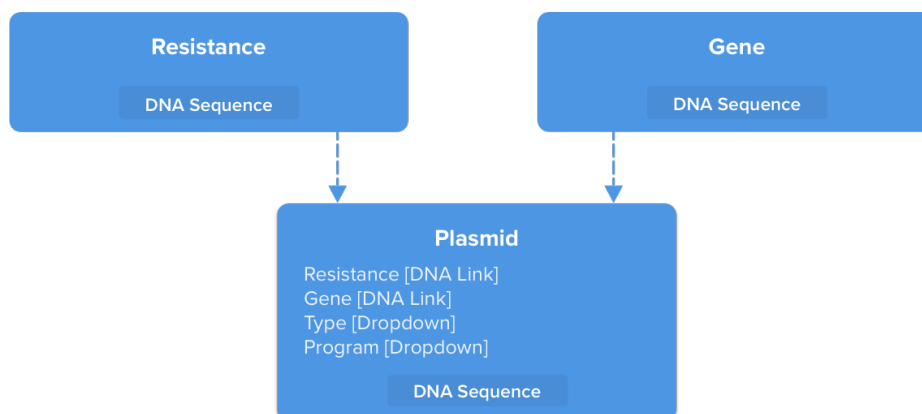

If you're configuring a Bioregistry for gene therapy, plasmids will be a fundamental part of your registration needs. But first, let's consider two upstream components of any plasmid: **Resistances** and **Genes**.

In our foundational model, we structure **Resistances** and **Genes** very simply: as DNA sequences. Rather than explicitly track other fields on these entities, we're going to register them only as named DNA sequences. (In Benchling, we can also view and manipulate these sequences in the built-in Molecular Biology Suite.)

Now that we've modeled **Resistances** and **Genes**, we can move on to assembling our **Plasmid** entity. Similar to **Resistances** and **Genes**, we'll model the **Plasmid** as a DNA sequence entity. But unlike them, we're going to track more than just the **Plasmid's** sequence. Firstly, we'll structure links between our **Resistance** and **Gene** entities, and our **Plasmid** entity. This means that any **Plasmid** will have to have at least one **Resistance** and one **Gene** linked to it. Here, Benchling's built-in bio-intelligence comes in handy; by structuring this type of link, whenever we register a new **Plasmid**, Benchling will automatically ask, "Of all the previously-registered **Resistance** and **Gene** sequences, do any of them appear in this **Plasmid** sequence?" If Benchling recognizes a **Resistance** or **Gene** sequence in a newly registered **Plasmid**, it will automatically link that **Resistance** or **Gene** to the **Plasmid**, taking the data entry burden off of scientists. (Benchling also has data validation rules to ensure that with any manual linking, the linked **Resistance** or **Gene** sequence exists in the **Plasmid** sequence.)

Next, we'll add two other fields to our **Plasmid** entity: "Type" and "Program". Unlike **Resistance** and **Gene**, these fields will be dropdowns, rather than links to other entities. Having "Type" and "Program" as dropdowns will make it easier for scientists both to enter their data and make sure it's consistent.

The Virus
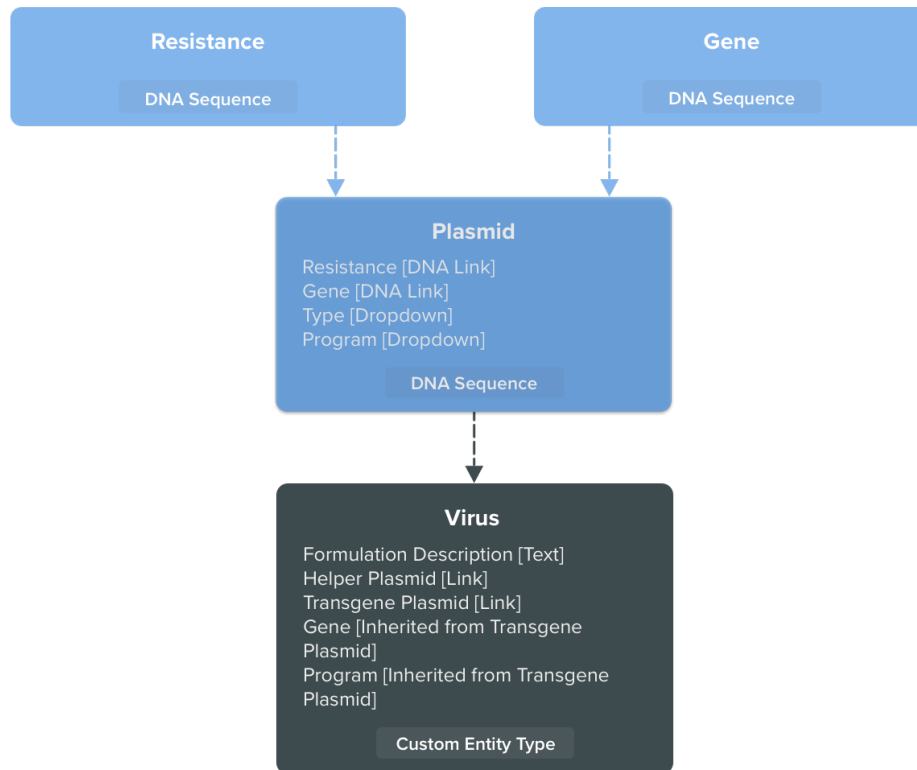
Unlike **Plasmids**, **Resistances**, and **Genes**, it's unnecessary for us to model a **Virus** as a DNA sequence-type entity. Instead, we'll model it as a "custom entity", which means that the **Virus** entity will be a collection of data fields that we configure, and it won't have a sequence in and of itself.

Our first **Virus** field will be "Formulation Description", which, unlike any of our previous fields, will contain freeform text rather than a dropdown. Next, we'll add two link-type fields: one for a **Plasmid** entity that the user will designate as the **Virus's** "Helper Plasmid", and one for a **Plasmid** that the user will designate as the **Virus's** "Transgene Plasmid". This linkage across entity types is similar to the **Plasmid** entity's links to a **Resistance** and a **Gene** (although again, unlike those other entities, the **Virus** entity has no sequence in and of itself).

Finally, we're going to structure our **Virus** entity such that it automatically surfaces the **Gene** and "Program" values that are on its linked **Transgene Plasmid**. Rather than have scientists clickthrough to the **Virus's Transgene Plasmid** to see which **Gene** is being delivered and which "Program" it's associated with, this allows scientists to see this data directly on the **Virus** itself. This reduces manual data entry, increases data integrity, and makes it all around easier for scientists to find the information that they need.
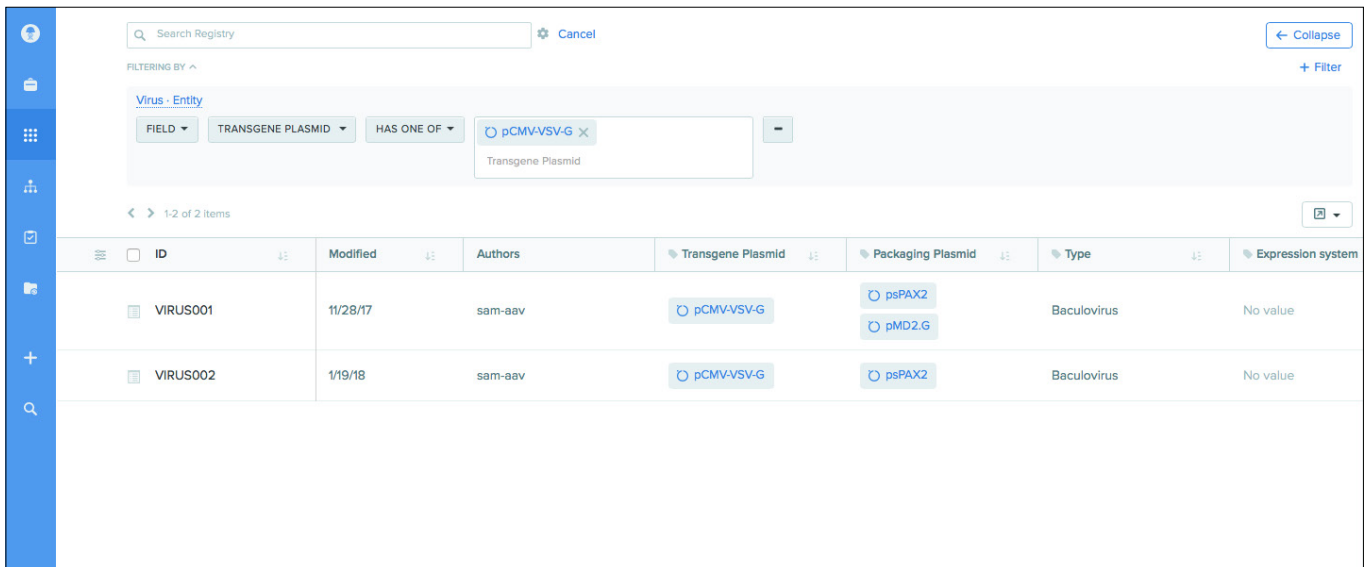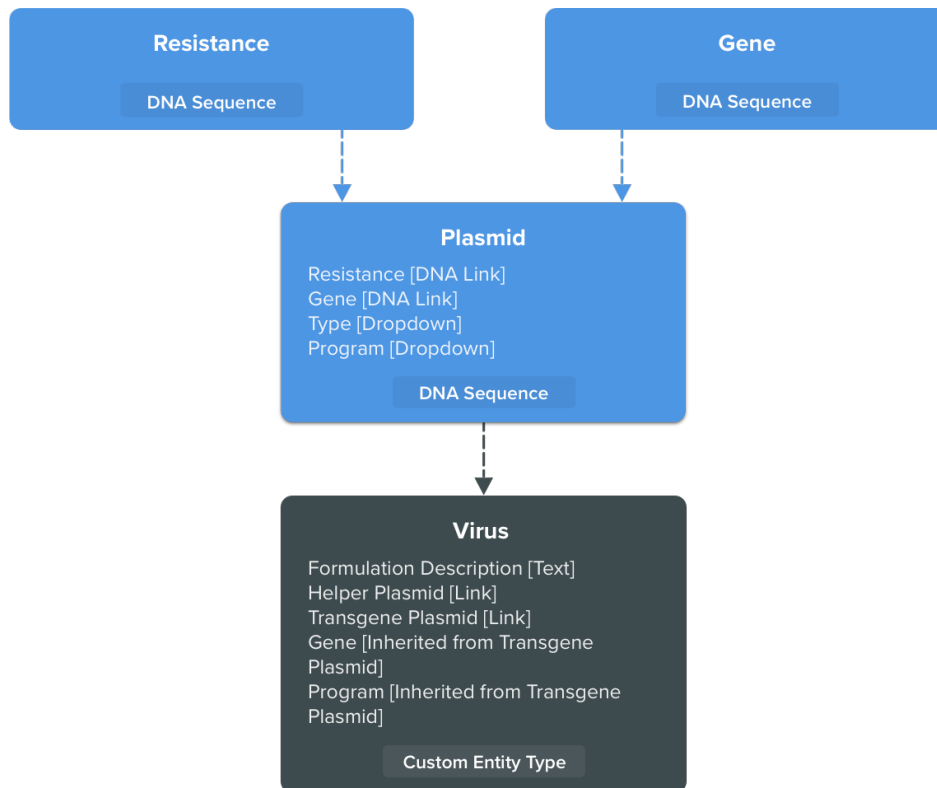
In short order, we've developed an interlinked model of a basic registration system for gene therapy. Needless to say, there are numerous additional fields that you could be tracking on these entities; for example, you could add fields for "Type" to Virus, or "Promoter" to Plasmid. You could also model downstream entities, such as Cell Lines or Tissue Samples, or even model key consumables.

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 plasmid with a certain resistance. Or, find every virus with a certain transgene. Or, of course, find all the viruses produced for a certain program.

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

Beyond running these queries, though, Benchling gives you access to all of the functional data produced by these entities. You can trace from your **Viruses** down to every physical container of them that has ever been produced. You can then answer questions such as, "For a certain transgene, which virus led to the highest relative gene expression?"

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