Benchling

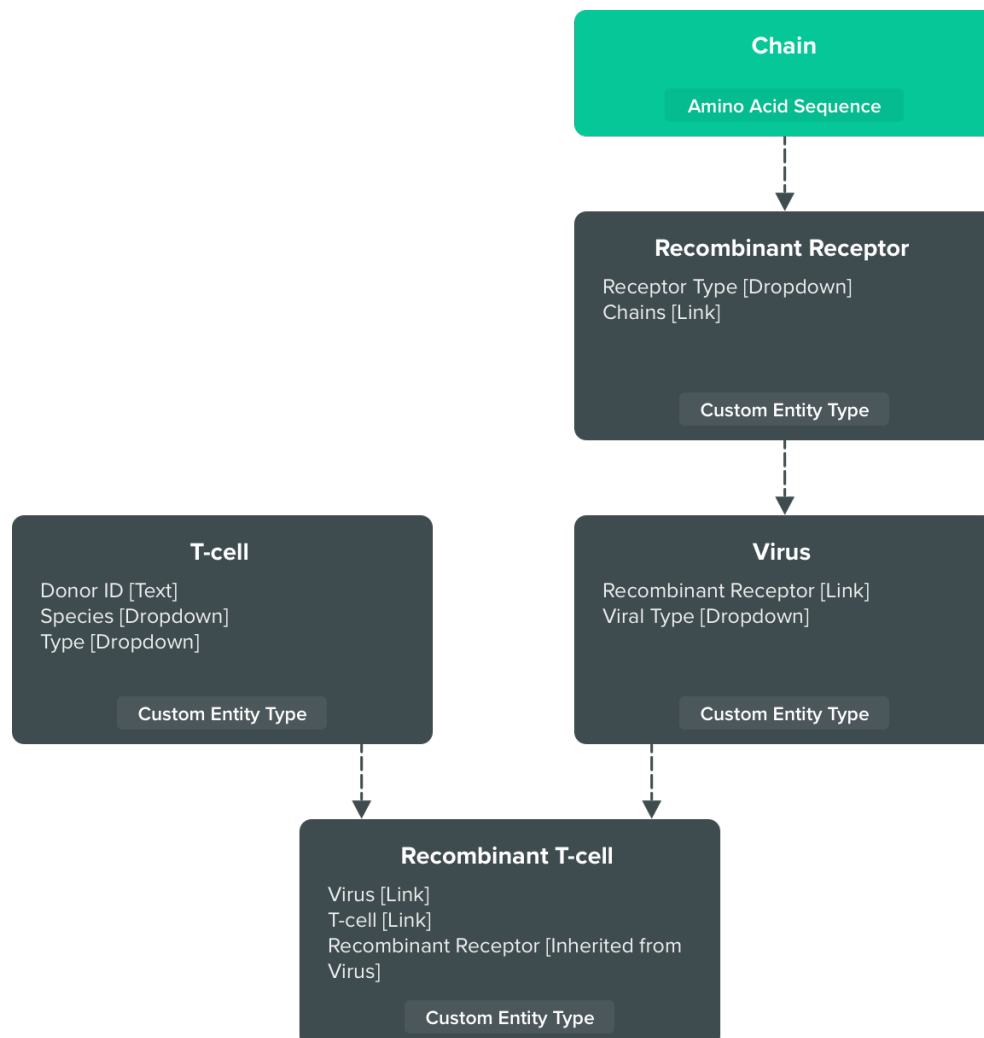## How to Configure a Bioregistry for TCR and CAR T R&D

From T-cells, to upstream receptors and chains, TCR and CAR T R&D involve 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.

The basic data model that we discuss here can serve as a starting-point for either TCR or CAR T R&D. 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 TCR and CAR T R&D**.

Chain

Amino Acid Sequence

Recombinant Receptor

Receptor Type [Dropdown]
Chains [Link]

Custom Entity Type

T-cell

Donor ID [Text]
Species [Dropdown]
Type [Dropdown]

Custom Entity Type

Virus

Recombinant Receptor [Link]
Viral Type [Dropdown]

Custom Entity Type

Recombinant T-cell

Virus [Link]
T-cell [Link]
Recombinant Receptor [Inherited from Virus]

Custom Entity Type

The Chain, the Receptor, and the Virus

```
┌─────────────────────────────────────┐
│              Chain                   │
│                                      │
│        ┌──────────────────┐          │
│        │ Amino Acid Sequence │       │
│        └──────────────────┘          │
└─────────────────────────────────────┘
                   ┊
                   ▼
┌─────────────────────────────────────┐
│        Recombinant Receptor          │
│                                      │
│  Receptor Type [Dropdown]            │
│  Chains [Link]                       │
│                                      │
│           ┌──────────────────┐       │
│           │ Custom Entity Type │     │
│           └──────────────────┘       │
└─────────────────────────────────────┘
                   ┊
                   ▼
┌─────────────────────────────────────┐
│               Virus                  │
│                                      │
│  Recombinant Receptor [Link]         │
│  Viral Type [Dropdown]               │
│                                      │
│           ┌──────────────────┐       │
│           │ Custom Entity Type │     │
│           └──────────────────┘       │
└─────────────────────────────────────┘
```
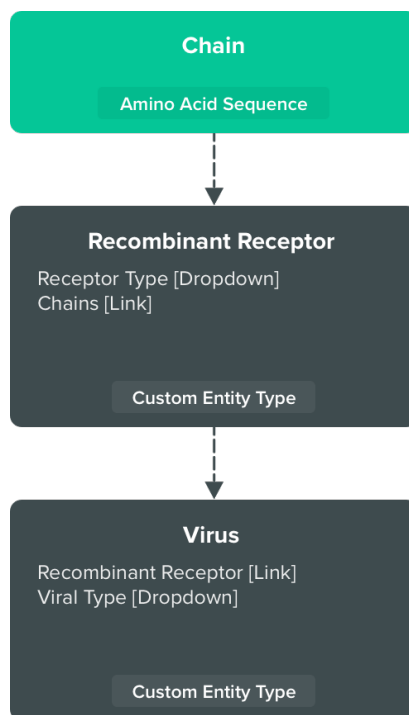
If you're configuring a Bioregistry for TCR/CAR T therapies, viruses will be a fundamental part of your registration needs. But first, let's consider two entities that are upstream of any virus in this workflow: **Chains** and **Recombinant Receptors**.

In our foundational model, we model **Chains** very simply: as amino acid sequences. Rather than explicitly track other fields on the **Chain** entity, we're going to structure it only as a named amino acid sequence. (In Benchling, we can also view and manipulate sequences in the built-in Molecular Biology Suite.)

The next entity to model is the **Recombinant Receptor**. Unlike **Chains**, it's not necessary for us to give **Recombinant Receptors** a sequence. Instead, we'll model **Receptors** as a custom entity type, which means that they'll be a collection of data fields that we configure, without having a sequence in and of themselves. The first field that we'll add to our **Recombinant Receptor** is "Receptor Type", which we'll model as a dropdown list. For organizations doing both TCR and CAR T R&D, this is a chance for them to distinguish the receptors that they're registering, by either labeling them "TCR" or "CAR". The next field that we'll add to our **Recombinant Receptor** is a link to one of the **Chain** entities that we referenced earlier. From a scientific standpoint, this makes sense: scientists will want to see the **Chain** upstream of each **Receptor**.
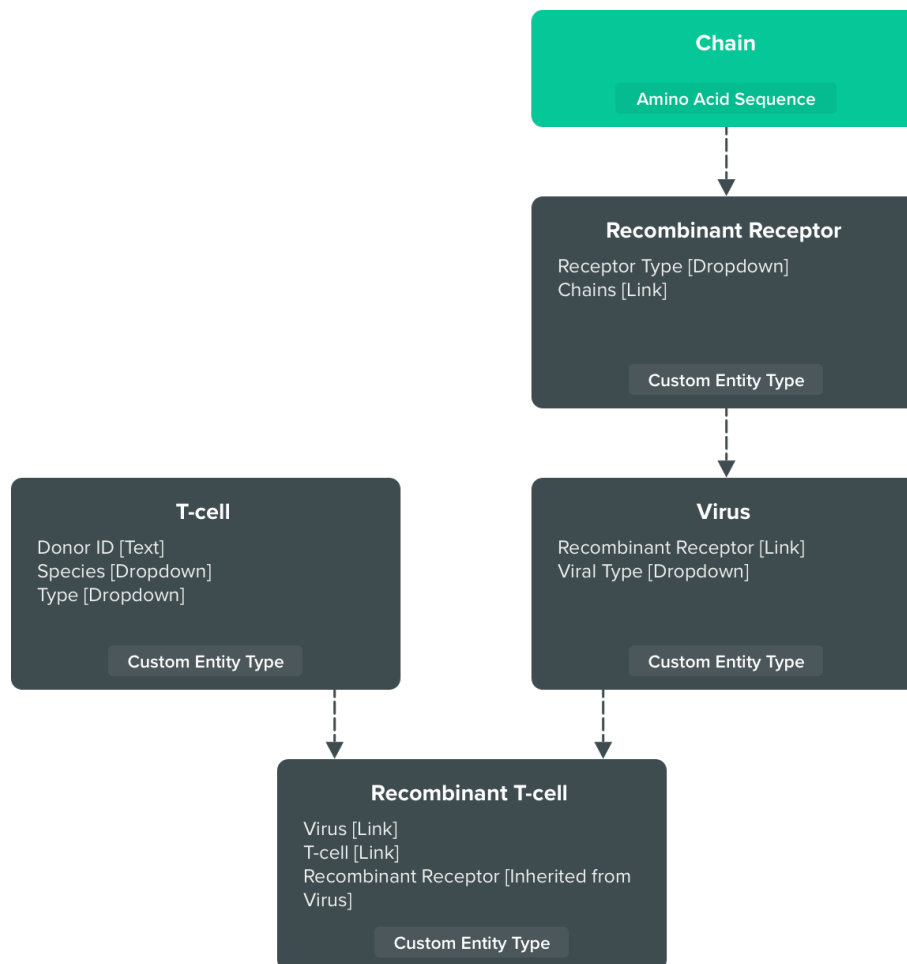
Now that we've modeled **Chains** and **Recombinant Receptors**, we can model our **Virus** entity. Like **Receptors**, the **Virus** will be a custom entity type. Its first field will be a link to a **Recombinant Receptor** (in the same way that we linked **Recombinant Receptors** to **Chains**). The second field we'll structure on **Viruses** is "Viral Type", which we'll format as a dropdown to make it easier for scientists both to enter their data and make sure it's consistent.

For a more advanced data model, you might consider including a **Plasmid** entity. A **Plasmid** would have a DNA sequence, and you could structure a validation rule to ensure that part of its DNA sequence should be translatable into the **Chain**'s AA sequence. This would allow you to track which plasmids were used to produce which recombinant T-cells, and track your production over time.
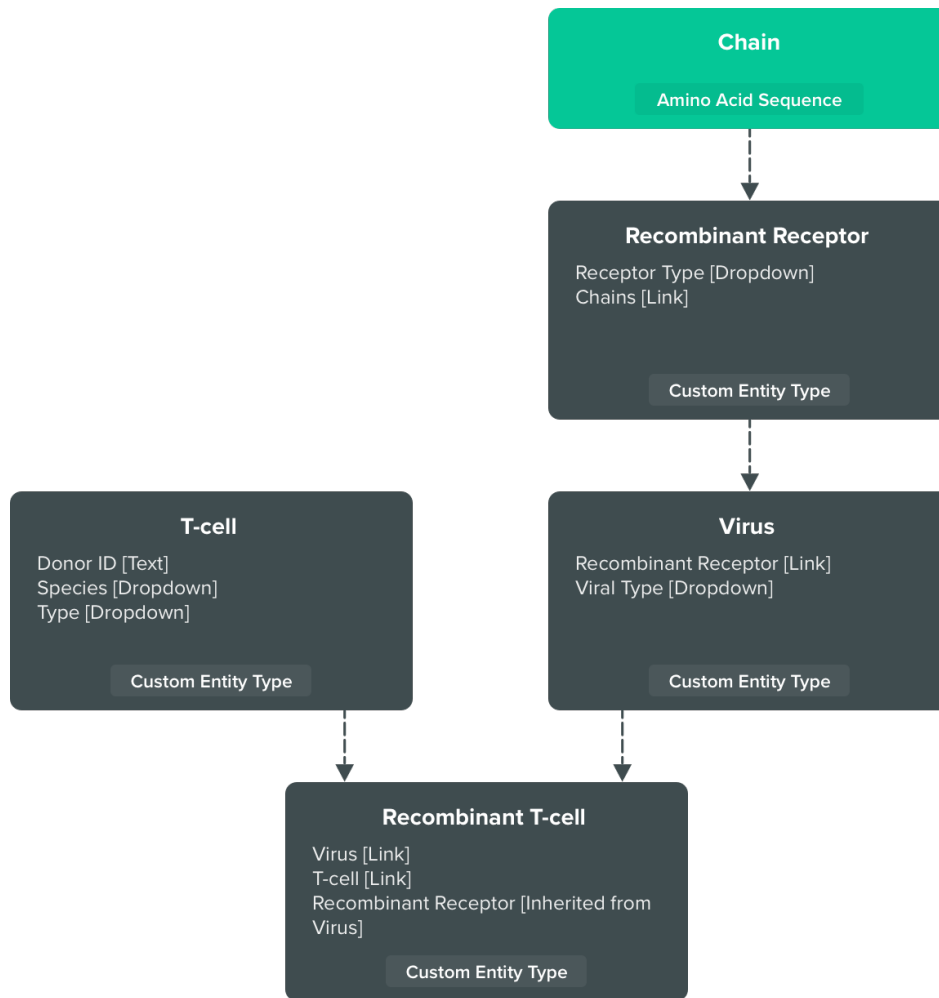
The T-Cell and the Recombinant T-Cell

```
            ┌─────────────────────────────────┐
            │            Chain                 │
            │   ┌─────────────────────────┐    │
            │   │  Amino Acid Sequence    │    │
            │   └─────────────────────────┘    │
            └─────────────────────────────────┘
                           │
                           ▼
            ┌─────────────────────────────────┐
            │      Recombinant Receptor        │
            │  Receptor Type [Dropdown]        │
            │  Chains [Link]                   │
            │                                  │
            │        ┌──────────────────┐      │
            │        │ Custom Entity Type│     │
            │        └──────────────────┘      │
            └─────────────────────────────────┘
                           │
                           ▼
┌──────────────────────┐   ┌─────────────────────────────────┐
│       T-cell          │   │            Virus                 │
│  Donor ID [Text]      │   │  Recombinant Receptor [Link]     │
│  Species [Dropdown]   │   │  Viral Type [Dropdown]           │
│  Type [Dropdown]      │   │                                  │
│  ┌──────────────────┐ │   │      ┌──────────────────┐        │
│  │Custom Entity Type│ │   │      │Custom Entity Type│        │
│  └──────────────────┘ │   │      └──────────────────┘        │
└──────────────────────┘   └─────────────────────────────────┘
            │                            │
            ▼                            ▼
        ┌───────────────────────────────────┐
        │        Recombinant T-cell          │
        │  Virus [Link]                      │
        │  T-cell [Link]                     │
        │  Recombinant Receptor [Inherited from│
        │  Virus]                            │
        │       ┌──────────────────┐         │
        │       │Custom Entity Type│         │
        │       └──────────────────┘         │
        └───────────────────────────────────┘
```

Separate from our hierarchy of **Chain ➜ Recombinant Receptor ➜ Virus**, we'll model a **T-cell** entity, which will be, again, a custom entity type. First, we'll put a "Donor ID" on this entity, which will be a text field to accommodate free-form data entry (one opportunity here is to integrate the Benchling Bioregistry with a manufacturing database, so that Donor IDs can be automatically sync'd). We'll also structure fields for "Species" and "Type", both of which will be dropdowns.

Our final entity is the **Recombinant T-cell**, which brings together all the entities we've modeled previously. Its first two fields will be links to a **T-cell** entity and a **Virus** entity, while its final field will be a link to the **Recombinant Receptor** *inherited through the linked **Virus***. In other words, rather than force scientists to manually link **Recombinant T-cells** to **Recombinant Receptors**, Benchling will automatically detect and surface the **Recombinant Receptor** that the **Recombinant T-cell's Virus** is linked to. Allowing entities to inherit data in this way reduces manual data entry, increases data integrity, and makes it all around easier for scientists to find the information that they need.
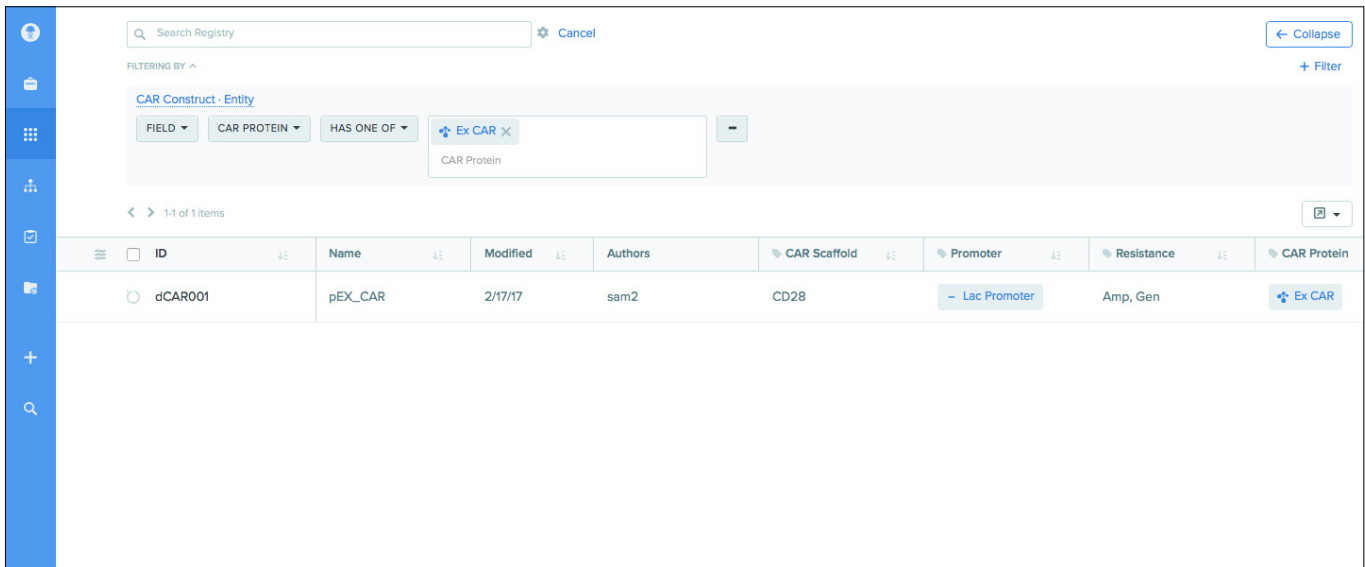
Chain
Amino Acid Sequence

Recombinant Receptor
Receptor Type [Dropdown]
Chains [Link]

Custom Entity Type

T-cell
Donor ID [Text]
Species [Dropdown]
Type [Dropdown]

Custom Entity Type

Virus
Recombinant Receptor [Link]
Viral Type [Dropdown]

Custom Entity Type

Recombinant T-cell
Virus [Link]
T-cell [Link]
Recombinant Receptor [Inherited from Virus]

Custom Entity Type

In short order, we've developed an interlinked model of a basic registration system for TCR and CAR T R&D. Needless to say, there are numerous additional fields that you could be tracking on these entities; for example, you could create entities that model Donors, or Plasmids. You could then link Donor entities to **T-cells**, and various types of Plasmids to **Viruses**.

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 **Recombinant T-cell** with a certain **Receptor**. Or, find the locations of every R**ecombinant T-cell** batch that came from a certain Donor.

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 down from your entities down to every physical container of them that has ever been produced. You can then answer questions such as, "For a certain donor, which virus led to the highest receptor expression levels?"

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