Chapter Y Structuring Your Data

# STRUCTURING YOUR DATA

## HIGH LEVEL ORGANIZATION

In order to enter your data into CRIPT, it helps to have a plan for how you would like to organize your data before you start using the various software tools to enter it. While there are many ways to do this correctly, this chapter presents a systematic approach to thinking about your data that will help you to structure it in a way that makes ingestion into CRIPT relatively straightforward. The goal of this chapter is to end with a graphical layout of your data that follows CRIPT’s data model,[REF] and we will follow an example from the literature[REFS] to illustrate what this might look like.

**High-Level Organization Quick Guide**

1. Identify the *Project* to which you will add your data or make a new *Project*. Make sure you have identified all collaborators who should have access and include them!
2. Add a new *Collection* to your *Project* if necessary.
3. Create an *Inventory* within the *Collection* to hold *Materials* if you do not already have a suitable *Inventory*.

A detailed description of CRIPT’s data model has been published,[REFS] and additional documentation is available on the CRIPT website at the help page.[REFS] You may find it useful to consult this documentation if you have additional questions on technical details as you work through this chapter of the CRIPT manual.

The first step to structuring your data is to establish the high-level organizational tools that are needed to hold your data. These include three types of nodes (node names are italicized and capitalized to clearly identify them as official terms in the CRIPT data model): *Project*, *Collection*, and *Inventory*. Of these, the highest level organizational structure and the place to start is the *Project*. A project represents a sustained scientific effort by a group of individuals that spans one or more scientific publications. In some instances, a researcher (for example, a university PI) may wish to use a *Project* as an organizational tool in which the entire team is a member. However, the prevalence of interdisciplinary research often means that members of a given research team may contribute to research both within the team as well as, for example, a collaborative research center with other universities. In these cases, *Projects* can be used to represent the different research thrusts. Your data may be added to an existing *Project* in CRIPT, or you may decide to create a new *Project.* Since *Projects* are the primary method of controlling who has access to the data, think about both how the intellectual content of the work fits with other *Projects* and if the group of people involved in the *Project* is identical to existing *Projects*.

Within each *Project*, you must now define *Collections*. *Collections* are the organizational unit used for publication of your data, so it is strongly recommended to create one *Collection* for each publication you will submit on your research. Of course, you may create a *Collection* before publication and publish it at a later time following the instructions in Chapter ABC of this manual. For the example considered here, representing a single published article, a single *Collection* will be defined.

Finally, *Inventories* must be defined. An *Inventory* is an organizational tool that is used to hold *Materials*, and it will be stored within your *Collection* so that it is published as a part of publishing the collection. *Inventories* may also be stored directly in a *Project*, but in this case they do not publish as part of publishing a collection; therefore, you will need one or more *Inventories* within your Collection in order to publish. Figure Y-1 illustrates this high-level organizational structure for our network synthesis and characterization example.

## IDENTIFY YOUR MATERIALS

The next step in laying out your experiment is to identify all of the *Materials*. Each material in your research is represented by a separate *Material* node, and these nodes can be reused within a *Project*. Mixtures are themselves a *Material* composed of other *Materials*, so your *Materials* will have a hierarchical arrangement. All *Materials* are added directly to your *Project*, and those you wish to publish as a part of your *Collection* should be additionally linked to your *Inventory*. Do not forget to include all of the reagents for your syntheses! In our Example, illustrated in Figure Y-2, note that all of the common reagents are not linked to the *Inventory* because we do not wish to publish them as novel compounds, but all of the custom compounds are linked to the *Inventory*.

**Identifying Materials Quick Guide**

1. List out all of your *Materials*, being sure to include reagents and products and to list each mixture and component of a mixture as a separate *Material*.
2. List all the information you wish to include about each Material, and categorize it as *Identifier*, *Component*, *Property*, or *Computation\_Forcefield*.
3. For *Properties* and *Identifiers*, find the relevant term in the CRIPT Controlled Vocabulary to label them.
4. Add *References* to your *Materials* or *Properties* as necessary.

Once you have delineated a list of all your materials, you should list the information that you wish to report on each one. CRIPT requires a certain minimum set of information,[REFS] but you will likely want to list much more than that. Items you should think about include chemical structure (in line notation such as BigSMILES[REF] or another representation), molar mass, and properties such as radius of gyration, glass transition, or phase. These different items will be classified as *Identifiers* (material identifiers such as chemical structure, IUPAC name, or trade name), *Components* (the materials that make up a material that is a mixture; must be selected from other materials in your *Project*), *Properties* (the properties such as molar mass and glass transition temperature), and *Computation\_Forcefield* (the force field used to generate computational materials; does not apply to experimental systems). While you don’t have to classify the information you wish to list into these groups initially, it will certainly help you when it is time to enter the data into CRIPT because each class has somewhat different input specifications. Figure Y-3 illustrates a few examples of *Materials*, the information that is listed for each one, and the classification of each piece of information.

*Identifiers* and *Properties* have an additional level of complexity because they are subject to a controlled vocabulary. The point of a controlled vocabulary is to prevent the same property or identifier from being entered into the database under multiple different names, making it difficult to find all the synonyms. However, this is also a challenge for new users, as it takes some time to gain familiarity with the controlled vocabulary. The controlled vocabulary for *Identifiers* is called “Material Identifier Key” while that for *Properties* is called “Material Property Key.”[REFS] CRIPT’s controlled vocabulary is dynamic and growing with the scope of researchers participating in CRIPT. If you don’t see a term you need in the controlled vocabulary, you can help to add it by following the instructions in Chapter X. Where a term does exist in the vocabulary already, it is helpful to include it in your planning since these will be the descriptors used in CRIPT’s software system to indicate the correct *Identifier* or *Property*. Figure Y-3 also illustrates this concept.

In many cases, you will be entering materials from your own experiments. However, in some cases, you have extracted data about materials from previously published literature. In this case, you should also create a *Reference* node that will contain bibliographic information about the literature source of your data, and you should link it to the relevant *Material* and/or *Property*. An example is included in **WHICH FIGURE?**

## BUILDING YOUR EXPERIMENTS

**Building Experiments Quick Guide**

1. Create one or more *Experiments* to hold your *Processes* and *Data*.
2. Add *Processes* representing the physical or chemical transformations that link your *Materials* together.
3. List all the information you wish to include about each *Process* and categorize it as *Description*, *Equipment*, *Condition*, or *Property*.
4. For *Equipment, Conditions,* and *Properties*, find the relevant term in the CRIPT Controlled Vocabulary to label them.
5. Add *References* to your *Processes* as necessary.
6. Add *Data* describing your *Processes* and *Materials* for each type of data that you acquired.

The next step in structuring your data is to enter *Processes* that were used to make your materials and *Data* that describes them. These two nodes are contained within an organizational structure *Experiments*, so you must first create at least on *Experiment* before you enter this information. What is included in each *Experiment* is up to the user and their personal organizational style. If you prefer not to use this organizational tool, simply create a single *Experiment* and use it to hold all your *Processes* and *Data*. However, you may also use *Experiments* to divide *Processes* and *Data* as granularly as you choose.

After deciding on how many *Experiments* you will have, you should enter your *Processes*. *Processes* are used to transform one experimental material into another (for computation, see “Theory, Simulation, and Computational Processes” later in this chapter). They may be chemical syntheses like polymerizations, they may be physical processes such as extrusion or film casting, or they may be combinations of the two such as thermal curing. A *Process* will link to *Materials* that it takes as inputs and produce other *Materials* as products or waste. *Processes* may also exist in a series with no Materials in between, providing a method of representing complex, multi-step *Processes*. How you divide up a *Process* is up to you; you may build each step as a separate *Process* node or include all the different steps within a single node. The advantage of splitting a *Process* up is that the individual steps become more easily searched; the disadvantage is that it will typically increase the data entry burden on the user. Making this decision when you are initially structuring your data will dramatically simplify your work when you try to upload data into CRIPT. Linking of *Processes* to *Materials* and *Experiments* is illustrated in Figure Y-4.

In addition to the *Material* inputs and outputs, each *Process* may contain information about how the procedure it describes was performed. This information is typically stored as *Description*, *Equipment*, *Conditions* and *Properties*. *Description* provides a text description of the *Process*; you may enter a paragraph with whatever information you like into this field. It’s not necessary to include this paragraph in your data graph layout, but it is useful to have it handy before you start the data ingestion process. *Equipment* is a list of the required equipment for the *Process*, and it has a Controlled Vocabulary called “Equipment Key.”[REF] *Conditions* and *Properties* both describe information about the process such as temperatures, pressures, flow rates, or reaction times. *Conditions* represent information that is specified or controlled (such as when a temperature is held constant using a temperature controller), while *Properties* represent information that is measured about a process, such as the torque curve produced by an extruder operating at constant rpm. Both *Conditions* and *Properties* have corresponding Controlled Vocabularies called “Condition Key” and “Process Property Key,”[REFS] respectively. If your Process is from the literature, you may also add a *Reference*. Figure Y-5 shows examples of *Processes* with detailed information specified.

The final step in building your *Experiments* is to add *Data*. Data nodes include experimental data that describes a *Material*, *Process*, or *Computational\_Process* (see “Theory, Simulation, and Computational Processes” below). *Data* includes things such as GPC chromatograms, SANS scattering patterns, TEM images, and UV/Vis spectra. This information can be very complex, as it includes a variety of different steps for data acquisition and analysis, all of which can be captured in CRIPT and are described in subsequent sections. For now, simply enumerate the types of Data that you have collected and wish to include, and link each one to the Material or Process that it describes. Do not worry about defining all the descriptive information you will need just yet. Figure Y-6 shows an example.

## SAMPLE PREP IS JUST ANOTHER PROCESS

Entering sample preparation information into CRIPT is supported in two ways depending upon the preferences of the user. Fundamentally, the decision comes down to scientific judgement about whether the sample being analyzed is identical to or different from the parent material. When a sample is identical to the parent material, the sample preparation information is entered into a *Process* node, as described above, but the *Process* node does not have any product *Materials*. Examples that fall into this paradigm would typically include GPC analysis (where the sample is prepared by mixing a polymer with solvent but the final GPC result is indicative of the original polymer) or oscillatory shear rheology of polymer melts (the sample is often shaped to fit into the rheometer, but pains are taken to avoid shear history effects due to this preparation so it is believed that the shear rheology represents the original polymer). The second paradigm is when sample preparation fundamentally transforms a material. In this case, the sample to be analyzed should be explicitly included as a *Material* node, and the sample preparation *Process* appears as a process linking input and output *Materials*, exactly as described above. Examples that fall into this paradigm might include spin-casting a block copolymer film for AFM analysis (since the film can have a different nanostructure than the bulk material) or crystallization of polyolefins during flow processes where the processing has a profound impact on the percent crystallinity. As you consider which paradigm applies to all of your data, you may find that you need to add new *Material* nodes to describe some samples. Examples of both of these paradigms are illustrated in Figure Y-7.

**Sample Prep Quick Guide**

1. Add *Processes* to describe sample preparation.

## ADDING COMPUTATIONAL WORKFLOWS FOR DATA ANALYSIS

Converting raw data from an experiment into usable information about a material or process typically involves a chain of different steps: a sample is prepared, it is analyzed to generate raw data, that raw data may be reduced or analyzed into intermediate data sets, and eventually a final data set is analyzed to extract a *Property* of the *Process* or *Material* such as its torque, heat flux, flow rate, phase, percent crystallinity, glass transition, or molar mass. How much detail about this process you include is up to you. At the least detailed level, all of the different levels of analysis can be attached to a single *Data* node as different files. However, you may also enter details on the computational steps you use to reduce, analyze, fit, or otherwise process your data within *Computation* nodes. To do this, you build a graph of alternating *Data* and *Computation* nodes where the *Computation* node holds details of the analysis that was used to convert the data in one *Data* node into the data in another. Figure Y-8 illustrates how such a chain is assembled. Note that it is not necessary for the final *Data* node in an analysis pipeline to link back to a *Material* or *Process* node. For example, you may take a TEM image of a polymer without actually extracting a property from it. If you are entering data before all of your analysis and experimentation is complete, there will be many more such *Data* nodes without links.

For each *Data* and *Computation* node in the chain, you will need to specify key information. For the *Data* node, this is simply the *Types* of data/files that you wish to attach to the node. Type is governed by the Controlled Vocabulary “Data Type.” Although it’s not required to produce your data graph, it is also helpful as you are thinking through which data you will include to gather all of the files that contain this data. Each *Computation* may represent either a single analysis step or a multi-step analysis, or you may granularize your anaysis by linking *Computations* in series as you see fit. For each *Computation*, you will want to think about which information on *Software\_Configuration* and *Conditions* you wish to include. *Conditions* is governed by the “Condition Key”[REF] Controlled Vocabulary and describes controlled variables during the analysis as it does for *Processes*. This is most commonly used in molecular simulation and would be less commonly required in data analysis. *Software\_Configuration* contains information on the software used and the variable settings within that software that were used to perform the analysis. Figure Y-9 shows examples of the relevant detail for *Computation* nodes used in data analysis. You may also add *References* to link *Computation* to the source of the code or analysis method in the published literature.

**Data Analysis Quick Guide**

1. Expand each *Data* node into a series of *Data* and *Computation* nodes that represent your data analysis pipeline.
2. Add relevant information describing each *Data* and *Computation* node, using relevant Controlled Vocabularies where appropriate.

## THEORY, SIMULATION, AND COMPUTATIONAL PROCESSES

Details on computational process and further elaboration of computational workflow needed to do simulation work.

## FURTHER DETAIL AND FURTHER EXAMPLES

The CRIPT data model contains significant additional complexity to deal with complex experimental systems and a researcher’s desire to annotate data to make it more detailed or more findable. This chapter has focused on the most common and important information and on the high-level organization of data to match the data model; it is not a comprehensive introduction to all data model features. The article on the CRIPT data model [REF] and its supporting information[REF] describe all of the additional features, and you will encounter them throughout all of the different ingestion tools described in this manual in **Chapters X-Z**. The article describing the CRIPT data model also contains a myriad of other graphical examples illustrating a variety of different possible experiments and their high-level data graph organization. The full detail of the nodes discussed here is identical as would be required in those graphs, allowing you to use them for additional inspiration in how your data might be laid out.