Genome Analysis Toolkit (GATK) User Manual

Matt Hanna

30 Mar 2009

1 Walkers

1.1 Command-line Arguments

Users can create command-line arguments for walkers by creating public member variables annotated with @Argument in the walker. The @Argument annotation takes a number of different parameters:

fullName	The full name of this argument. Defaults to the toLowerCase()'d member name. When specifying fullName on the command line, prefix a double dash (–).
shortName	The alternate, short name for this argument. Defaults to the first letter of the member name. When specifying shortName on the command line, prefix a single dash (-).
doc	Documentation for this argument. Will appear in help output when a user either requests help with the –help (-h) argument or when a user specifies an invalid set of arguments.
required	Whether the argument is required when used with this walker. Default is required = true.
exclusive	Specifies that this argument is mutually exclusive of another argument in the same walker. Defaults to not mutually exclusive of any other arguments.

The GATK will parse the argument differently depending on the type of the public member variable's type. Many different argument types are supported, including primitives and their wrappers, arrays, typed and untyped collections, and any type with a String constructor.

When the GATK cannot completely infer the type (such as in the case of untyped collections), it will assume that the argument is a String. GATK is aware of concrete implementations of some interfaces and abstract classes. If the argument's member variable is of type List or Set, the GATK will fill the member variable with a concrete ArrayList or TreeSet, respectively. Maps are not currently supported.

1.2 Output

By default, the walkers provide protected out and err PrintStreams. Users can write to these streams just as they write to System.out and System.err. This output can be redirected to a file using the out, err, and outerr command-line arguments.