Pipeline Management

A set of computational tools, which run either sequentially or parallelly in order to achieve a specific data analysis objective. Tools/commands are designated as steps in a pipeline.

- Favorites Pipelines
- My Pipelines
- Pipeline Creation
- Pipeline Filters
- Copy Pipeline
- De Novo Pipeline
- Pipeline Execution

Favorites Pipelines

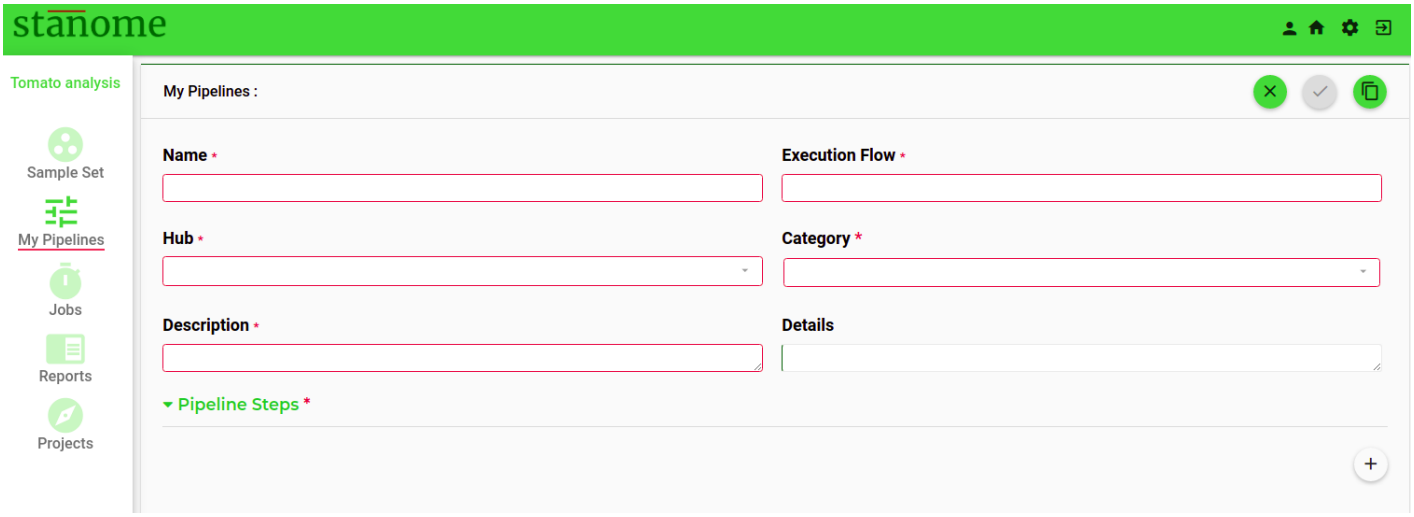
Pipelines, from any project, can be added or deleted from the favorites by clicking the  icon next to the pipeline name in the list view i.e. **My Pipelines**. Stanome owned pipelines can’t be added to the favorites. Favorite pipelines are visible in the **Pipeline Library** and **My Pipelines** (filtered based on the Owner, Hub, and Category).

My Pipelines

My Pipelines show the list of pipelines within a project. Pipelines can be created, viewed, edited, and deleted within the project scope. **My Pipelines** is empty, by default, and users can create new pipelines *de novo* or copy the pre-configured pipelines from the **Pipeline Library**. Follow the instructions in the next section to create a pipeline.

Pipeline Creation

The new pipeline creation window is accessed from two locations. Either click on the project window or click on the **My Pipelines** window to create a new pipeline (Fig. 1).



The new pipeline creation window displays three icons in the upper right corner.

Exit out of the pipeline creation without saving.

Save the pipeline.

Copy pipeline.

New pipelines can be added to a project in three ways.

- Copying pre-configured pipelines
- Creating *de novo*
- Favorite pipeline

Pipeline Filters

Pipelines on the platform are tightly connected with the projects. The **Data type** field in the projects is tied with the **Hub** field in the pipelines. Based on the **Data Type** definition in the project setup, only relevant pipelines are shown. The following table shows the corresponding terms between projects and pipelines.

Data Type in **Project**	**Pipeline Hub**
Whole Genome	Genome
Microbiome	Micro
Transcriptome	Transcript
Targeted Genome	Variant

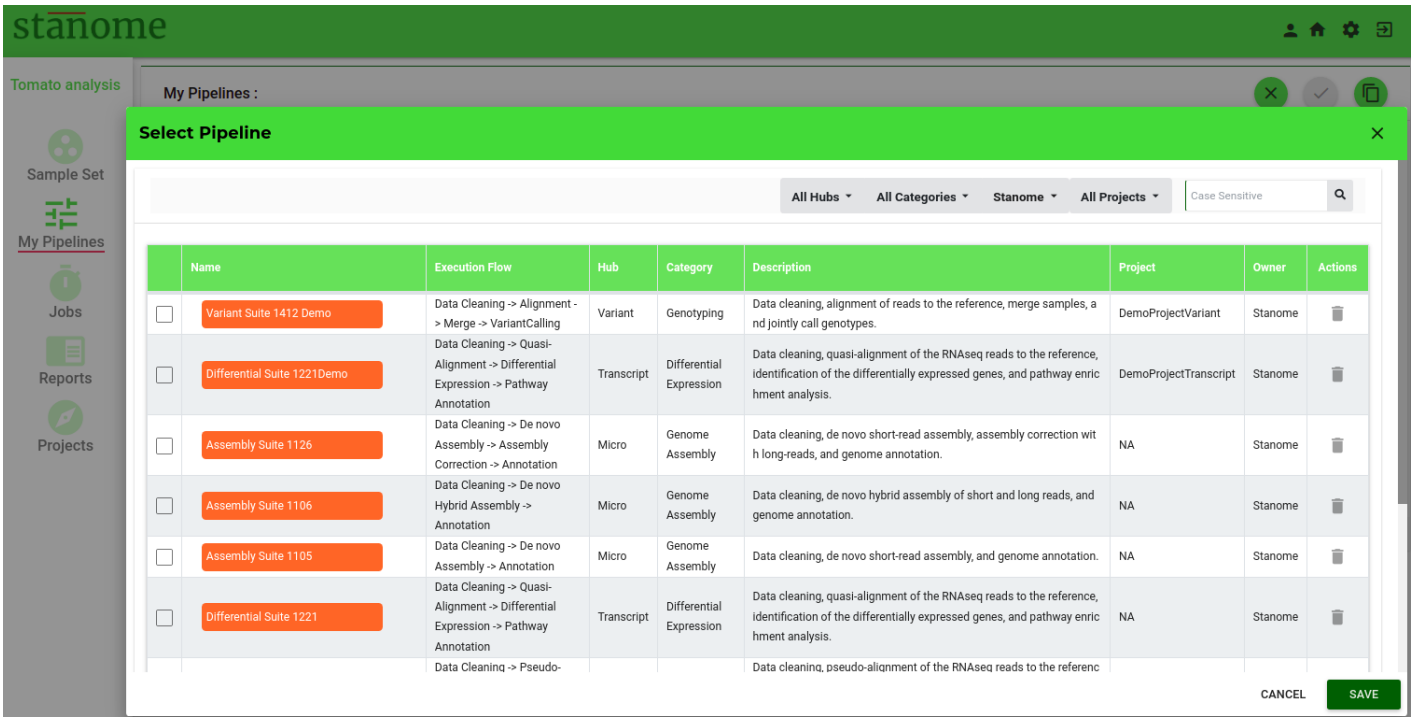
Table 4: Terms associated between projects and pipelines

Copy Pipeline

A new pipeline can be created by copying an existing pipeline from other projects or a pre-configured pipeline from **Pipeline Library**.

Click in the upper right corner of the pipeline creation window to see the existing pipelines via the **Select pipeline** dialog box (Fig.). Using the project setup information, a list of prefiltered pipelines is listed. Users can use multiple field combinations to filter the pipelines. Pipelines from the can be viewed by selecting the owner as “Stanome”. Select a pipeline and click on to copy a pipeline into the current project. The pipeline steps, tools, parameters, and other details are auto-populated (except the pipeline name). Name the new pipeline uniquely (duplicate names are not allowed) and verify the tools and commands before saving the pipeline.

HINT: Pipeline name should be less than 50 characters long and only alphanumeric characters and spaces are allowed.



De Novo Pipeline

De novo pipeline building requires bioinformatics expertise. Please contact the technical support team for assistance.

The creation of a brand new pipeline is more challenging than copying an existing pipeline. Fill in the following details on the pipeline creation window (Fig. 1) to create a new pipeline. Mandatory fields are indicated with asterisks (*).

- **Name*** - Provide a unique name
- **Execution Flow*** - Enter the tool names in the order of execution (e.g. Trimmomatic -> Salmon -> Sleuth)
- **Hub*** - Indicate the pipeline group
- **Category*** - Select the functional category
- **Description*** - Provide a brief description of the pipeline's general purpose
- **Details** - Provide detailed information about tools, inputs, outputs, arguments, and other pertinent information
- **Steps*** - Step field helps to add tools and commands to a pipeline.

At least one step is required for a functional pipeline.

Click the  icon to add a new step (or tool) to the pipeline(Fig. 1). There are eight fields in each step:

- **Number** - The number determines the step number in the pipeline. It is automatically filled when a new step is added.

 **HINT**: Only positive integers are allowed

- **Name** - Provide a name to the step (e.g. Bowtie2 alignment). The outputs will override if the name field is not unique because the name is used as a directory to store the output files of the step.

 **HINT**: The name should be unique.

- **Tool** - Select a tool from the drop-down menu

- **Command** - Select a command from the drop-down menu for the selected tool
- **Predecessor** - The number determines the dependency step of a step. A step executes only after the dependency step.

HINT: Only positive integers are allowed

- **Merge** - This is a critical variable that indicates if all the inputs need to be combined into a single output file. This is useful in some analyses where all files need to be analyzed together (Examples: Differential gene expression and Joint genotyping). Default: No.

HINT: The first step can't be a merge step

- **Input Source** - Determines the source of the input files for a step. Typically, input file sources are either **Data Store** (Sequence Data, References, Annotations, and Metadata) or output files from predecessor steps.

HINT: Input sources from multiple steps are allowed. (Example: BAM and BAI files created in different steps required for Variant calling).

HINT: Currently, **Data Store** is allowed for the first step only

- **Actions -** Each step allows the following actions
  - Access the **Command builder** dialog box
  - Delete a step
  - Copy a step

1. **Command Builder**

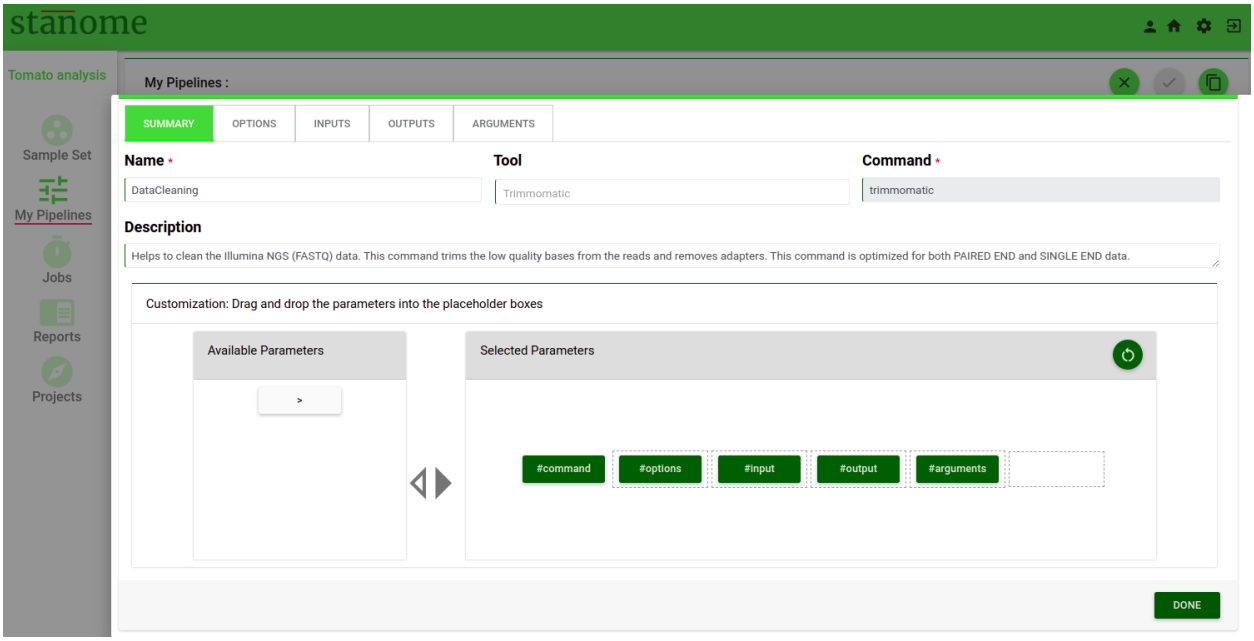
   - Command building is one of the complex processes on the platform and the **Command Builder** dialog box helps to navigate the process easily.

   Commands are preconfigured by the platform admin. Users can only edit the commands.

   - During the pipeline development stage users define the commands and the final executable command will be dynamically created during the pipeline execution stage.

   This is a generic command building process. You are NOT making the actual file selections required for the analysis. The platform does it automatically based on your definitions.

- Command Builder has two modes: View mode and Edit mode. The former allows viewing and the latter allows command modification, as described below. Access the **Command Builder** dialog box (Edit mode) (Fig. 1) by clicking  under the **Actions** column.



The first tab of the **Command Builder** describes the generic details(summary) about a command.

- **Name** - The step name as given by the user while creating the step.
- **Tool** - The selected tool (cannot be modified)
- **Command** - The actual command to be executed (cannot be modified)
- **Description** - Brief description of the command (auto-filled but can be modified by the user)
- **Build Your Command** - This box helps to build the actual command. The left box shows the available parameters and the right box (Fig. 2) shows the selected parameters. The order of the parameters is extremely important and should be maintained for the command to execute. All commands come with a default parameter sequence (Stanome defined). The parameters are prefixed with a #(hash). The default pattern can be modified by dragging and dropping the parameter buttons (green color) between the left and right boxes. Based on the selection the parameter

tabs are enabled on the top.

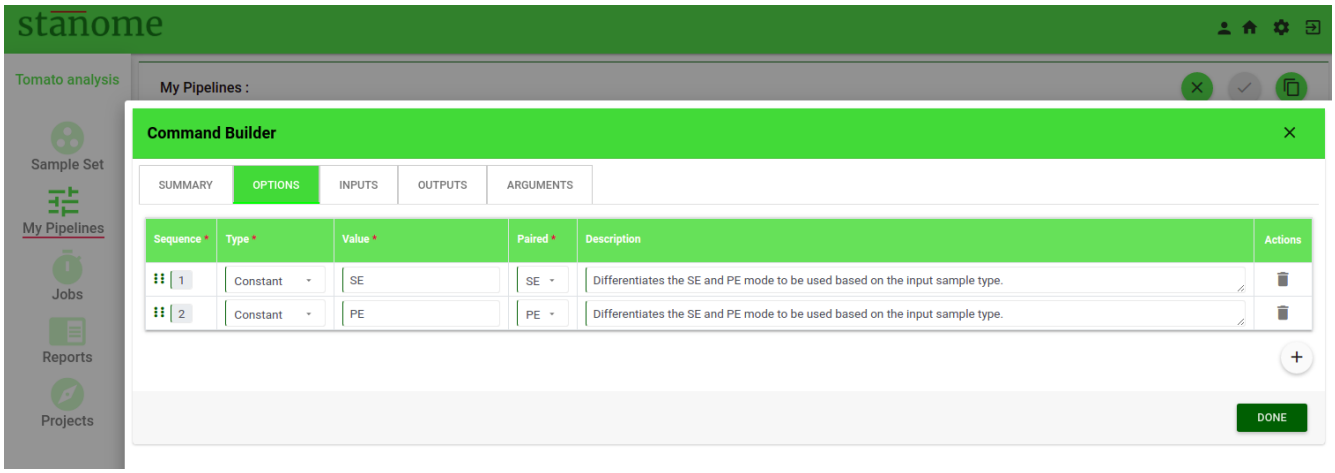
Default pattern: *#command #options #arguments #input #output*

The pattern should ALWAYS start with *#command* and can't be edited.

Allowed character: Parameter words, #, space, and >

">" is allowed preceding the *#output ONLY*

The second tab of the **Command Builder** (Fig. 2) describes the **Options** parameter. Details of the Options tab are described below:



- ***Options***

  Single-word parameters should be defined as options (Examples: --ignore, --1, PE, SE). All the options are listed in a table format. New row(s) can be added using the '+' sign at the bottom of the table. Six fields are available under each option.

  - **Sequence:** This number determines the order of the option in the command
  - **Type:** Six choices are available in the drop-down. Select the type of option. (Example: Annotation, Constant, Metadata, Reference, Threshold, and Variable)
  - **Value:** Based on the field **Type**, the corresponding values in the drop-down change. Select an appropriate value. See the table below for available field types and their values.

    CAUTION - Verify usage of each option before using

Field Type	Value

Annotation	• Variant annotation files (Mills1000G_INDELS, DBSNP, 1000G_HC, 1000G_OMNI, and HAPMAP), GATK • Pathway or GO • VEP Cache and VEP Cache Version • GTF • ABR
Constant	• Any constant value (alphanumerics) (Examples: -o, --i, and --single)
Metadata	• Experimental Design • Targets • Genelist • Amplicon ranges
Reference	Define references to select • References: Genome/Transcriptome • Indexed references: BWA, Bowtie2, etc
Threshold	Define threshold values to use • qvalue • pvalue
Variable	Native variables of the platform • JobID • Organism • Ploidy • Sample Name • Reference Version • Sequencing Platform

Table. Available **Field Types** and their corresponding **Values.**

- **Paired**: Indicates if an option can be used for paired-end, single-end files, or both (Default: All)
- **Description:** A brief description of the option functionality or usage guidelines.
- **Actions**: Allows the deletion of an option.

CAUTION - Please refer to the **Arguments** section for defining the parameters with key-value pairing

- ***Inputs and Outputs***

  Input and output files are defined under **INPUTS** and **OUTPUTS** tabs, respectively. Eight fields are available under each of these parameters (Fig. 3).

- **Sequence:** This number determines the order of the inputs or outputs in the command
- **Name:** The name of the value (Examples: --input, -output)
- **Type:** Input/output data can be provided to a command in three formats - File, FileList, and Directory. Select the format from the drop-down list.
    - File - Input is a single file (Example: Prefix.fastq)
    - FileList - Input is Paired-end files (Example: Prefix_R1.fastq, Prefix_R2.fastq)
    - Directory - Input is a directory path
- **Delimiter:** Character separating the input/output file(s) from its name in the command (Example: --input **:** Prefix.fastq).

CAUTION - Allowed delimiters are **=, -, :,** and **;**

- **Format**: Depending on the data select file extension from the drop-down. This value is used to make the right file selections during the pipeline execution. Required for File and FileList types only and not required for Directory type.

CAUTION - The file extensions should be precise; even the FASTQ and FQ are treated distinctly.

- **File Name Pattern:** This field is applicable for the Inputs parameter only. Regular expressions can be used to select specific input files. This value is used in combination with the **Format** field. Few examples are provided below for an easy understanding of regular expression usage.

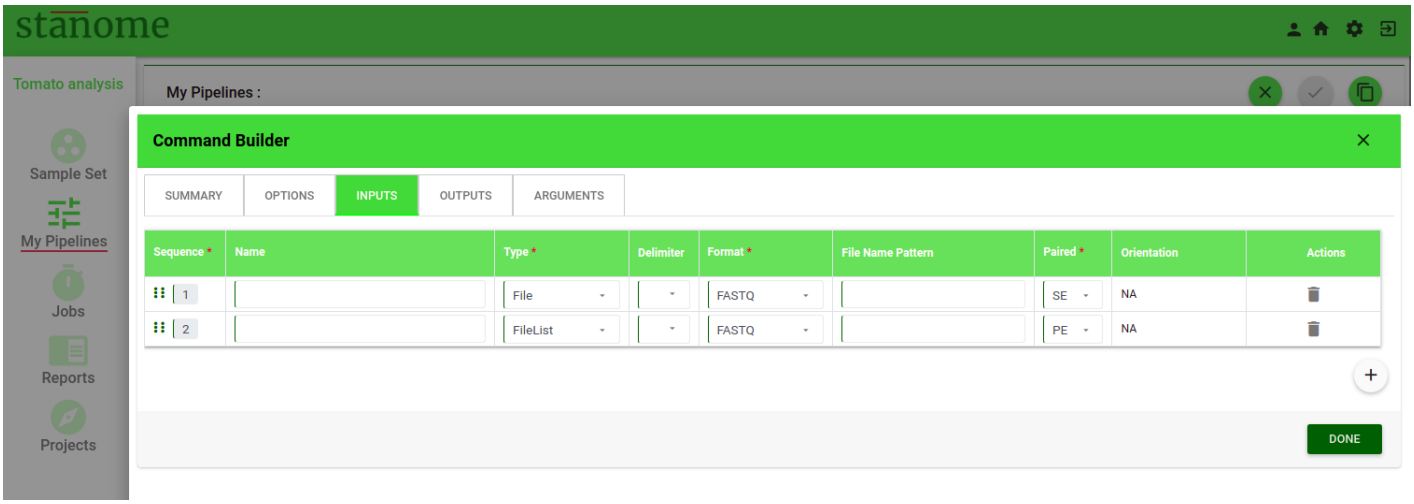
	Input file names	**Regular expression**
Example 1	castor1_R1.fastq	R1
Example 2	castor1_R1_trimmed.fastq	R1_trimmed
Example 3	abcd_1.fastq	_1
Example 4	abcd_1_R.fastq	_1_R

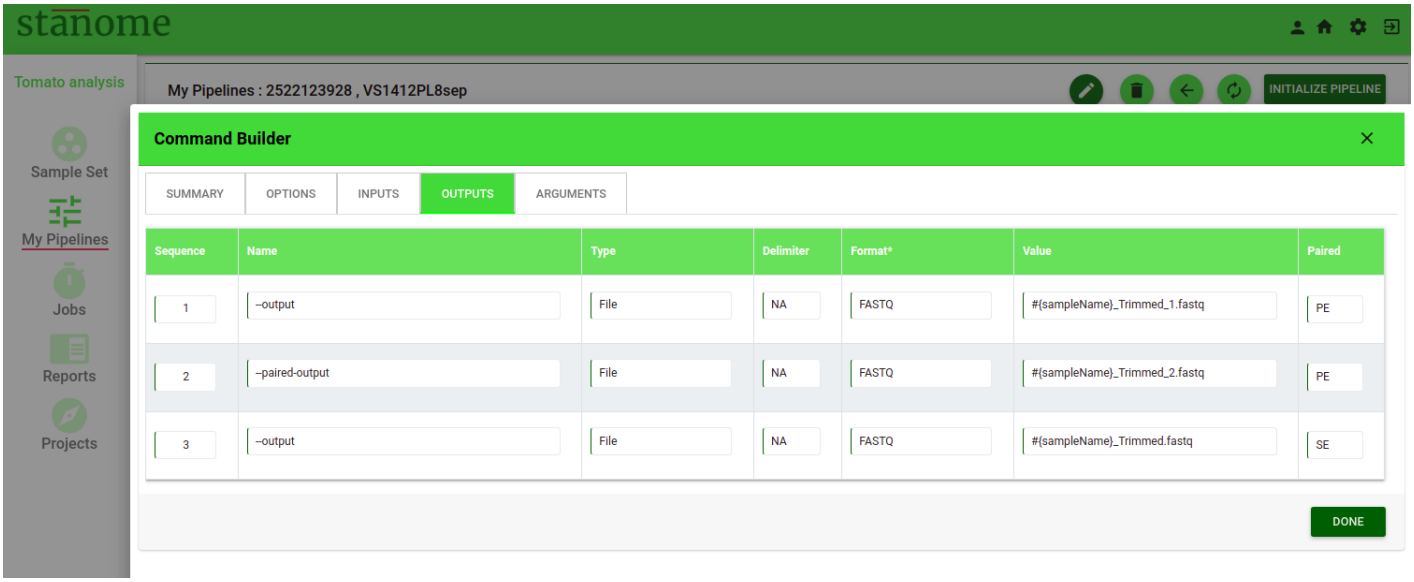
- **Value:** This field applies to the **Outputs** parameter only. Output value contains three parts

- Prefix: Stanome sample variable (**${sampleName}**)
- Suffix: Step name
- File extension

(Example: ${sampleName}_trim.fastq for trimmomatic step). This helps track the files across the entire pipeline execution.

- **Paired**: Indicates if the files (Inputs/Outputs) parameter is applicable to paired-end, single-end files, or both (Default: All).
- **Actions**: Allows the deletion of an input or output.





- ***Arguments***

Parameters defined as a key-value pair should be defined as arguments (Fig. 4). Arguments can be used for any parameters supported by the tools and other required files (reference files, gtf or annotation files, target or hotspot files). They are defined by the following eight features:

CAUTION - Please refer to the **Options** section for defining the singleton parameters

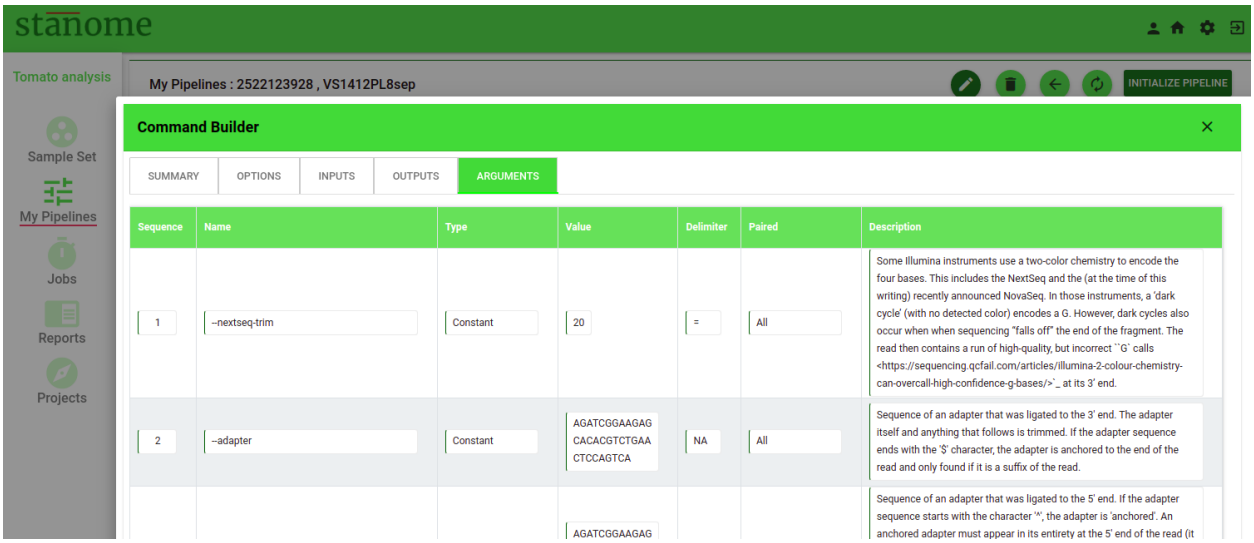
- **Sequence:** This number determines the order of the argument in the command.
- **Name:** Name of the argument used by the command to identify it

Arguments are grouped into categories to support diverse tools and commands. In arguments, two fields (Type and Value) work together to define an argument.

- **Type:** Seven choices are available in the drop-down. Select the type of argument. (Example: variable, constant, and annotation_DNAseq)
- **Value:** Based on the **Type** selected, the values in the drop-down change. Select the appropriate value. Refer to the table given in options for the available types and their values.
- **Delimiter:** Character separating the keys and values in the command (Example: --count**:** 10). Not all arguments require delimiters between the **Name** and the **Value** fields.

CAUTION - Allowed delimiters are **=, -, :, %,** and **,**

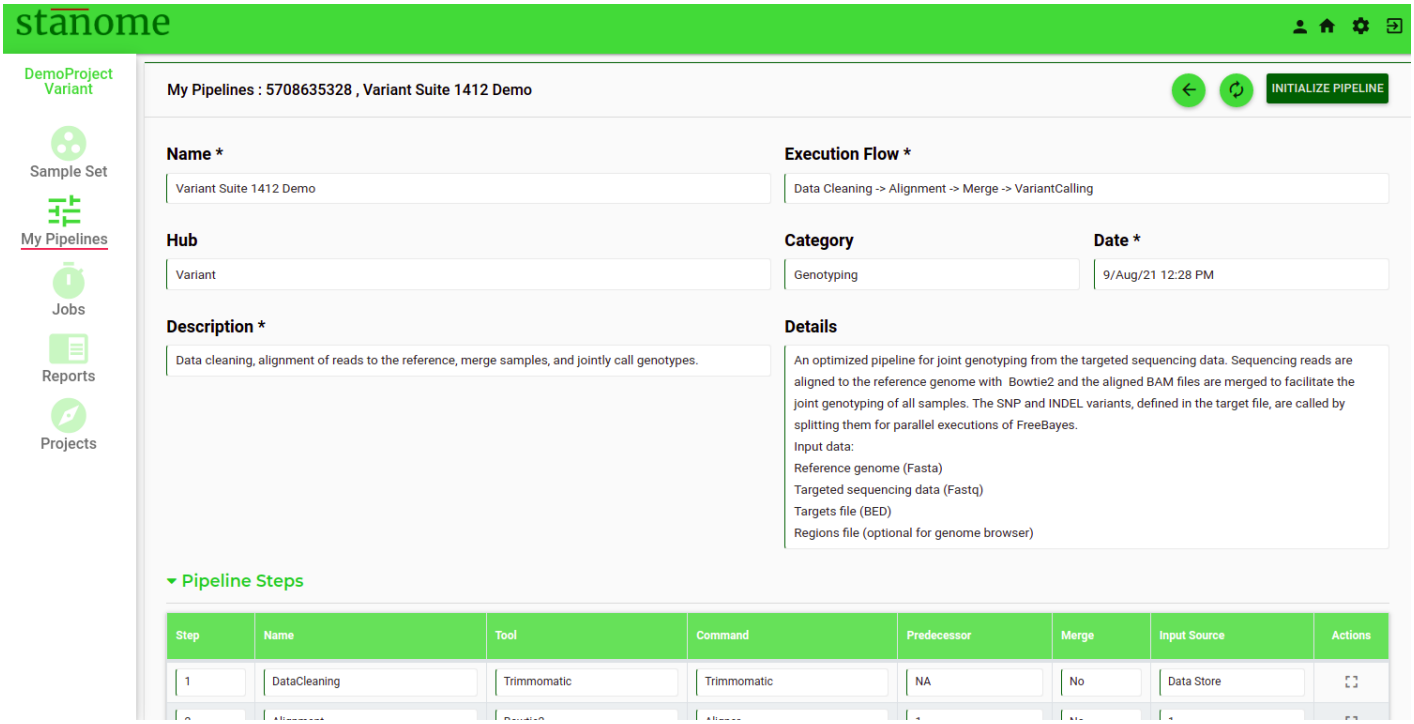
- **Paired:** Indicates if the arguments parameter applies to paired-end, single-end files, or both (Default: All).
- **Description:** A brief description of the argument describing the function and utility of the argument.
- **Actions**: Allows the deletion of an argument.



Click Image not fo on the bottom right corner to save the changes to the command. This is the completion of the first step in the pipeline. Continue adding all the steps until the pipeline is complete. Steps can be dragged and dropped at any position with the Im icon. Step number, predecessor, and input source get automatically readjusted for all the steps. Click Imag to save the pipeline.

Pipeline Execution

Once a pipeline is successfully created and validated within a project, it's ready for execution. A newly created pipeline is shown in Fig. 1.



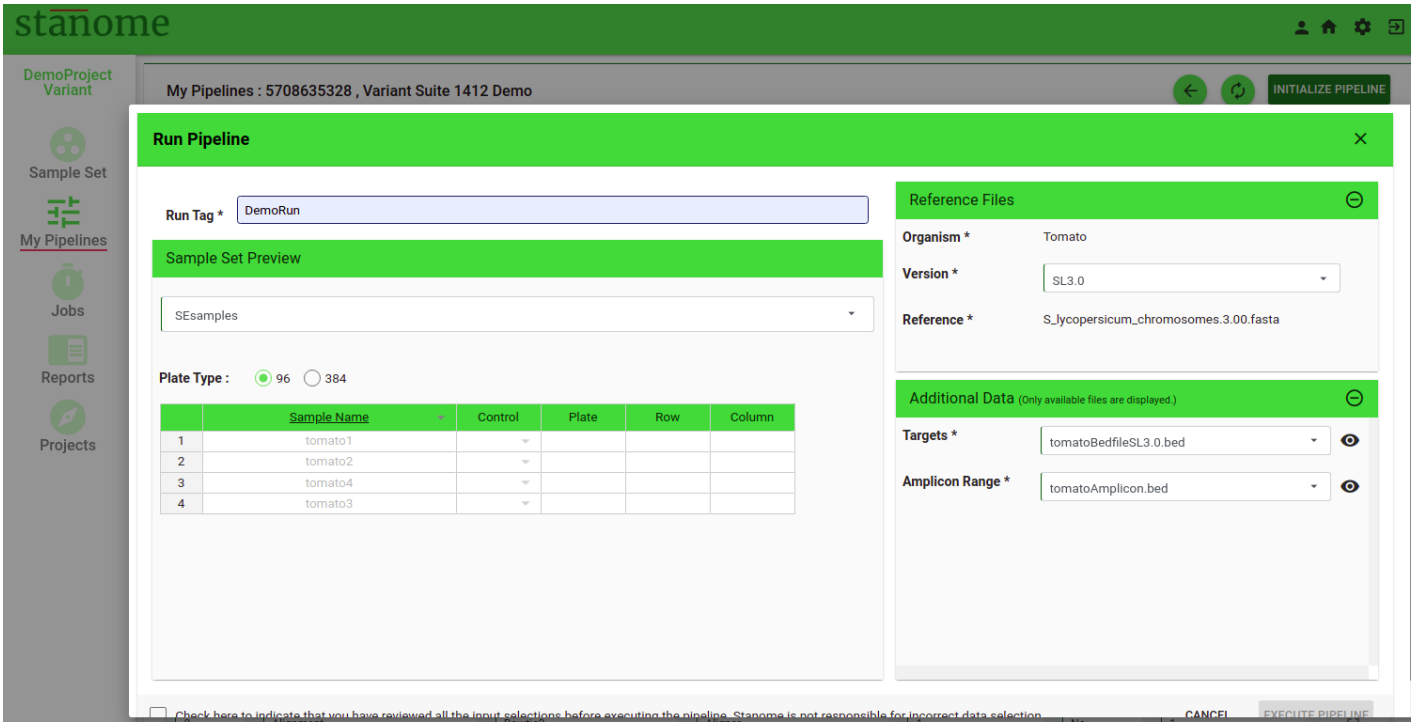
- The following actions are allowed on the **Pipeline** window: view/delete/edit/initialize.
  - **Pipeline** deletion
  - **Pipeline** edit
  - Page refresh
  - Back navigation
  - **Pipeline** initialization
- Click to access the **Run Pipeline** dialog box (Fig. 2). Final data selections happen during this stage and all fields are required to be filled. The contents of the dialog box change dynamically based on the analysis type and the tools in the pipeline.
- Provide a unique Run Tag
- Sample Set selection from the drop-down shows the sample names in a tabular format

- Select the plate format: 96-well or 384-well (This information is extensively used in the **Reports** to show the results in the plate format)
- Fill in the sample set table with the plate format details and control sample information.
- Latest reference genomes and corresponding additional files are preselected based on the organism of the project. Please confirm or change the selections using the drop-downs.

HINT: The contents of the metadata files can be viewed by clicking the icon

CAUTION - Differential Suite pipelines need at least two conditions with two replicates for each. Variant Suite pipelines need properly formatted target files.

- Agree to the terms and conditions to enable .
- Click to run the **Pipeline**.



Computing resources are initialized upon pipeline execution. The pipeline window automatically refreshes and redirects to the jobs window. Executed jobs appear in the jobs table. Refresh the window if the job is not visible. Jobs wait in the queue until computing resources are available and the status appears as pending and changes to Running. An email is sent when the job starts and also upon completion.

Pipeline Cancellation

Click the **STOP** button on the **JOBs** window to cancel an active pipeline execution and this will abort the run.

Pipeline Deletion

Click  to delete a **Pipeline** from the **Pipeline** window (Fig. 1). This action deletes the pipeline records entirely from a project and can’t be retrieved.