

Welcome Screen

Alaska: Complete, Automated RNA-seq Analysis

- 1 New Project
- 2 Load Project

1 Clicking this button creates a new project.

User's Perspective: proceed to next step

Command: `<PATH TO: request.sh>/Request.sh new_proj`

```
Correct output: lioscro@lioscro16:/media/sf_1/github/alaska/src5 sudo ./Request.sh new_proj
[sudo] password for lioscro:
Creating new proj request
Connecting to server on port 8888
Connected successfully
Waiting for response
APo7y6vr: creating
APo7y6vr: -/projects/APo7y6vr/ temp created
APo7y6vr: -/projects/APo7y6vr/ raw reads created
APo7y6vr: -/projects/APo7y6vr/1 alignment created
APo7y6vr: -/projects/APo7y6vr/2_diff_exp created
APo7y6vr: created successfully
END
```

Portal has to remember the ID of new project because it will be used in all downstream processes.

Read Upload Screen

Some text & info about uploading read files via FTP
Also need to make it clear that each sample must be in separate folders.
In the meantime, Raymond please allow us to access the "projects" folder under the Alaska root directory via FTP.

- 1 Upload finished

1 Clicking this button indicates the user has finished uploading their read files to our server.

User's Perspective: proceed to next step

Command: `<PATH TO: request.sh>/Request.sh infer_samples --id <PROJECT_ID>`

```
Correct output: lioscro@lioscro16:/media/sf_1/github/alaska/src5 sudo ./Request.sh infer_samples --id APo7y6vr
ID: APo7y6vr
[sudo] password for lioscro:
Creating infer_samples request
Connecting to server on port 8888
Connected successfully
Waiting for response
APo7y6vr: getting raw reads
APo7y6vr: calculating MD5 checksums
APo7y6vr: successfully retrieved raw reads
```

Note: The portal has to show a loading circle while the server is unpacking the reads and calculating the MD5 checksums. (The loading screen is shown below).

Loading Screen

Some text & info about uploading read files via FTP
Also need to make it clear that each sample must be in separate folders.
In the meantime, Raymond please allow us to access the "projects" folder under the Alaska root directory via FTP.

- 1 Useful text here to indicate unpacking & MD5 calculation progress.

- 1 Upload finished

1 This field shows some useful text about read unpacking and MD5 calculation

User's Perspective: changes from *getting raw reads* to *calculating MD5 checksums* to *successfully retrieved raw reads*

Note: The text changes according to the most recent output of the command run in **Read Upload Screen**. The window automatically proceeds to the next step when the command finishes successfully. (Assume all command terminations are successful terminations for now.)

Metadata Input Screen

Project ID: 1 <Project_ID>

Project Title: 2

Summary: 3

Contributor(s): 4 Add contributor

SRA Center Code: 5

Email: 6

... These indicate that there may be more fields in the future.

Experiment Design 7 1-factor 2-factor

Samples

Sample ID: 8 <Sample_ID>

Sample Title: 9

Description: 10

Contributor(s): 11 Add contributor

Organism: 12

Source: 13

Characteristics: 14

... This is my control: 15

Control Characteristic(s): 16

Read Type: 17 Single-end Paired-end

Index: 18

Length: 19

Standard Dev: 20

Bootstraps: 21

Reads & MD5 Checksums: 22

Read File	MD5 Checksum

23

Sample ID: <Sample_ID>

Sample ID: <Sample_ID>

Sample ID: <Sample_ID>

24 Submit

1 This text label shows the project ID.

User's Perspective: uneditable

Which field in JSON: *id*

2 This textfield is for the user to input a project title.

User's Perspective: editable

Which field in JSON: *meta - title*

3 This textfield is for the user to input a project summary.

User's Perspective: editable

Which field in JSON: *meta - summary*

4 This textfield is for the user to input contributor(s).

User's perspective: editable

Which field in JSON: *meta - contributors*

Note: Clicking the button *Add contributor* adds another textfield directly below the original textfield.

5 6 Same process as 2 and 3.

7 These radio buttons are for the user to choose their experimental design.

User's Perspective: choose between two radio buttons

Which field in JSON: *design*

Note: *1-factor* saves to the integer 1, and *2-factor* saves to the integer 2 in JSON

8 This text label shows the sample ID.

User's Perspective: uneditable

Which field in JSON: *samples - <Sample_ID> - id*

Note: Each sample is organized in a ribbon-like fashion. Each sample gets their own ribbon. Here is an example of what I think this should look like.

Dropdown Properties			
Name	Width	Input Size	Lang
Type	Dropdown (0)	Input Size	Lang

9 10 11 12 13 Similar to 2-6. Each field maps to their corresponding field in meta field.

14 This table is for the user to input the sample characteristics.

User's Perspective: editable (including adding/removing rows)

Which field in JSON: *samples - <Sample_ID> - meta - characteristics*

Note: Data in the table is saved as a hash with *Category* as the keys and *Details* as the values. Here is an example of what I think this should look like.

Custom Table Edit

SNO	Empid	Email	First Name	Last Name	Designation	Project Name	Actions
1	10001	Jayaram@gmail.com	Jayaram	JP	Software Engineer	Jayavsky	<input type="checkbox"/> Edit <input type="checkbox"/> Delete
2	10002	Ajuna@gmail.com	Ajuna	JD	Test	Sample Project	<input type="checkbox"/> Edit <input type="checkbox"/> Delete
3	1004	kumar@outlook.com	kumar	jp	Consultant	Test Project	<input type="checkbox"/> Edit <input type="checkbox"/> Delete
4	1005	hss					<input type="checkbox"/> Edit <input type="checkbox"/> Delete
5							<input type="checkbox"/> Edit <input type="checkbox"/> Delete

15 This checkbox is for the user to select their control.

If the sample is their control sample, they would check this box.

User's Perspective: editable

Which field in JSON: *ctrl_ids* (explained further below)

16 These dropdowns are for the user to select what their controlling characteristic(s) is(are).

User's Perspective: editable

Which field in JSON: *ctrl_ftrs* (explained further below)

Note: IF 7 is *1-factor*: only the first dropdown is active.

IF 7 is *2-factor*: both dropdowns are active & two different characteristics must be selected

17 These radio buttons are for the user to choose their read type.

User's Perspective: editable

Which field in JSON: *samples - <Sample_ID> - type*

Note: *Single-end* saves to the integer 1, and *paired-end* saves to the integer 2 in JSON

18 This dropdown is for the user to specify which transcriptome to use.

User's Perspective: editable

Which field in JSON: *samples - <Sample_ID> - idx*

Note: The choices of this dropdown are the list of *.idx* files in `<PATH TO: root folder>/idx`

19 This textfield is for the user to input the read fragment length.

User's Perspective: editable

Which field in JSON: *samples - <Sample_ID> - length*

Note: Data validation is needed when the user provides input to this textfield. (I'll specify this in a separate time.)

20 This textfield is for the user to input the standard deviation of the fragment lengths.

User's Perspective: editable

Which field in JSON: *samples - <Sample_ID> - stdev*

Note: Data validation is needed when the user provides input to this textfield. (I'll specify this in a separate time.)

21 This textfield is for the user to specify the number of bootstraps to perform.

User's Perspective: editable

Which field in JSON: *samples - <Sample_ID> - bootstrap_n*

Note: Data validation is needed when the user provides input to this textfield. (I'll specify this in a separate time.)

22 This table lets the user verify the MD5 checksums of their reads.

User's Perspective: uneditable

Which field in JSON: *samples - <Sample_ID> - reads AND samples - <Sample_ID> - chk_md5*

Note: The *reads* and *chk_md5s* are paired by order (first in *reads* with first in *chk_md5*, etc.)

23 Other sample ribbons containing the same stuff as 8-22.

24 This button submits all input data.

User's Perspective: If verification succeeds: proceed to next step

If verification fails: return the user to data input & highlight which fields failed verification

Clicking Submit saves all input as JSON to `<PATH_TO: root folder>/projects/<Project_ID>/_temp as <Project_ID>.json`

Then, issues the following **command:** `<PATH TO: request.sh>/Request.sh set_proj --id <PROJECT_ID>`

```
Correct output: lioscro@lioscro16:/media/sf_1/github/alaska/src5 sudo ./Request.sh set_proj --id APw5cz81
ID: APw5cz81
Creating finalize_proj request
Connecting to server on port 8888
Connected successfully
Waiting for response
APw5cz81: finalizing
APw5cz81: making directories for read alignment
APw5cz81: -/projects/APw5cz81/1 alignment/ASuomz6 created
APw5cz81: -/projects/APw5cz81/1 alignment/AS68u13e created
APw5cz81: -/projects/APw5cz81/1 alignment/AS16u135 created
APw5cz81: -/projects/APw5cz81/1 alignment/ASe3viq created
APw5cz81: -/projects/APw5cz81/ temp/APw5cz81.json removed
APw5cz81: successfully finalized
END
```

15 16 **Notes for ctrl_ids and ctrl_ftrs:**

- *ctrl_ids* is saved as a list of all Sample IDs that have the *This is my control* checkbox checked.

- *ctrl_ftrs* is saved as a hash of the *Category-Details* pair(s) chosen in 16

Data Review Screen

Simply the user to review their input data and check if everything's right. Displays the same information as the data input form, but without any editable fields.
Nothing occurs server-side.

- 1 Finalize

1 Clicking this button finalizes the project.

User's Perspective: Move on to next step.

Command: `<PATH TO: Request.sh>/Request.sh finalize_proj --id <Project_ID>`

```
Correct output: lioscro@lioscro16:/media/sf_1/github/alaska/src5 sudo ./Request.sh set_proj --id APw5cz81
ID: APw5cz81
Creating finalize_proj request
Connecting to server on port 8888
Connected successfully
Waiting for response
APw5cz81: finalizing
APw5cz81: making directories for read alignment
APw5cz81: -/projects/APw5cz81/1 alignment/ASuomz6 created
APw5cz81: -/projects/APw5cz81/1 alignment/AS68u13e created
APw5cz81: -/projects/APw5cz81/1 alignment/AS16u135 created
APw5cz81: -/projects/APw5cz81/1 alignment/ASe3viq created
APw5cz81: -/projects/APw5cz81/ temp/APw5cz81.json removed
APw5cz81: successfully finalized
END
```