Rep element pipeline

10-18-2017

Repetitive Element Mapping Pipeline

- **What:** maps HUMAN CLIP reads to repetitive elements
- **Why:** Eric does some neat analysis, let's share (and let's share in a way that is easier to integrate into the current eCLIP pipeline)!
- **How:**
  - `module load ecliprepmap`
- **Code:** https://github.com/YeoLab/repetitive-element-mapping
- **Input:**
  - R1+R2 fastq.gz for barcode 1 and barcode 2
  - rmRep.bam* files for barcode 1 and barcode 2
- **Output:**
  - Barcode-merged, de-duplicated reads mapped to both repetitive and non-repetitive elements in a SAM-like format

*TrTr--SoMaSoCoSo.bam

Repetitive Element Mapping Pipeline: Usage

```
ssh tscc-X-YZ                    # get interactive node

pathreset                        # reset path

module load ecliprepmap          # load the module

cp $ECLIPREPMAP_HOME/example/TEMPLATE.ecliprepmap ./  # *
```

*** Copy and fill out the TEMPLATE.ecliprepmap YAML file:**

```
    barcode1r1FastqGz:
      class: File
      path: data/RBFOX2_A01_R1.fastq.gz
    barcode1r2FastqGz:
      class: File
      path: data/RBFOX2_A01_R2.fastq.gz
    barcode1rmRepBam:
      class: File
      path: data/RBFOX2_A01_rmrep.bam
    ...
./TEMPLATE.ecliprepmap           # run the pipeline
```

Repetitive Element Mapping Pipeline: Output

```
├── data
│   ├── RBFOX2_A01_R1.fastq.gz
│   ├── ...
├── RBFOX2
│   ├── ECLIPREPMAP_RBFOX2_INPUT.yaml
│   ├── ECLIPREPMAP_RBFOX2_LOG.txt
│   ├── ECLIPREPMAP_RBFOX2_OUTPUT.json
│   ├── ECLIPREPMAP_RBFOX2_PIPELINE-ecliprepmap
│   ├── ECLIPREPMAP_RBFOX2_VERSION-0.0.1
│   ├── ECLIPREPMAP_RBFOX2_WORKFLOW-wf_ecliprepmap
│   └── results
│       ├── ecliprepmap_barcode1.concatenated.sam
│       ├── ecliprepmap_barcode2.concatenated.sam
│       └── ecliprepmap_concatenated.sam
├── RBFOX2.ecliprepmap
```

RBFOX2 (204_01): **`$ECLIPREPMAP_HOME/example/RBFOX2_expected/`**