ChIPathlon: A competitive assessment for gene regulation tools.

Avi Knecht  
*University of Nebraska-Lincoln, avikurtknecht.com*

Adam Caprez  
*University of Nebraska-Lincoln, acaprez2@unl.edu*

Istvan Ladunga  
*University of Nebraska-Lincoln, sladunga@unl.edu*

Follow this and additional works at: [http://digitalcommons.unl.edu/ucarereresearch](http://digitalcommons.unl.edu/ucarereresearch)

Part of the [Bioinformatics Commons](http://digitalcommons.unl.edu/bioinformatics_commons), [Cancer Biology Commons](http://digitalcommons.unl.edu/cancerbiology_commons), [Other Computer Sciences Commons](http://digitalcommons.unl.edu/othercomputercommons), and the [Software Engineering Commons](http://digitalcommons.unl.edu/softwareengineering_commons)
ChIPathlon: a competitive assessment for gene regulation tools

Avi Knecht, Adam Caprez, Istvan Ladunga
Gene regulation: why do we care?

- When gene regulation of the cell cycle malfunctions, it frequently causes cancer.
- Adult, differentiated cells can be reprogrammed to induced pluripotent stem cells
  - Which can then be reprogrammed to heart muscle, skin, etc., to repair damaged tissue (to limited extent in clinical practice)
Mapping transcription factors & histone modifications to genome

- Genes are regulated by transcription factors and proteins, that bind to specific sequences on the DNA.

- Transcription factors are mapped to the DNA by chromatin immunoprecipitation followed by next-generation sequencing.
Critical steps in data analysis

Peak calling

Using strand dependent bimodality in peak calling

Challenges in mapping transcription factors to the genome

- Background correction is an open problem.
- DNA fragments can be much larger than the binding site.
- Sequencing read location does not follow any statistical distribution.
- Over 50 different methods are used for mapping, which produce different results.
ChIPathlon

Evaluate the performance of all transcription factor mapping (peak calling) methods.

To this end, we will develop a scalable and easy to use super computing pipeline to stage data, compare many different peak calling and differential binding site tools, and store all results into a single database.
MongoDB

- Works well with large data sets.
- Can handle incomplete data.
Pegasus

- Too much data for manual processing, need to create workflows.
- Built on condor, which is used by a variety of super computing centers.
Python

- Many bioinformatics packages already managed in python under bioconda.
- Has interfaces for both MongoDB & Pegasus.
YAML to Workflows

Each individual job is defined in a plain text YAML file.

```yaml
zcat_awk_sort_peaks:
  inputs:
    - bed:
      type: file
      additional_inputs: null
  outputs:
    - bed:
      type: file
  command: zcat
  arguments:
    - "$inputs.0":
      changeable: false
      required: true
      has_value: false
    - "$outputs.0":
      changeable: false
      required: true
      has_value: false
  walltime: 2000
  memory: 2000
  cores: 1
```
Jobs are chained together by using module YAML files.
Users need to input a file selecting ENCODE experiment id’s for files to process, tools they want to use, and a path to a genome.

```yaml
1  runs:
2   - experiment: "ENCSR605MFS"
3       align: bwa
4       peak: spp
5   - experiment: "ENCSR605MFS"
6       align: bowtie2
7       peak: spp
8   - experiment: "ENCSR000ERE"
9       align: bwa
10      peak: spp
11  - experiment: "ENCSR000EGZ"
12      align: bowtie2
13      peak: macs2
14  genomes:
15   bwa:
16     grch38p6: "/path/to/genome/base"
17   bowtie2:
18     grch38p6: "/path/to/genome/base"
```
Conclusion

• The current pipeline handles all downloads, alignment of single or paired end reads, and peak calling.
• The modularity of the underlying architecture makes it very easy to add additional tools or processing steps without changing the workflow generation code.
• Workflows can be generated for any ENCODE experiment, making this a very versatile pipeline for comparing bioinformatics tools.
Database Architecture

Experiments
- _id
- accession
- aliases : list
- alternate_accessions : list
- assay_synonyms : list
- assay_term_id
- assay_term_name
- assembly : list
- award
- biosample_synonyms : list
- biosample_term_id
- biosample_term_name
- biosample_type
- contributing_files : list
- date_created
- date_released
- dbxrefs : list
- description
- developmental_slims : list
- documents : list
- files : list
- hub
- id
- lab
- month_released
- organ_slims : list
- original_files : list
- possible_controls : list
- references : list
- related_files : list
- related_series : list
- replicates : list
- replication_type
- revoked_files : list
- schema_version
- status
- submitted_by
- system_slims : list
- type : list
- uuid

Samples
- _id
- experiment_id
- organism
- genome
- transcription_factor
- histone_modification
- cell_type
- condition
- url
- control

Results
- _id
- gridfs_id
- control_ids : list
- experiment_ids : list
- alignParams : list
- duplicateParams : list
- peakParams : list
- dfbsParams : list
- peak_tool
- dfbs_tool
- align_tool
- duplicate_level
- result_type

Bed
- _id
- result_id
- chromosome
- start
- end
- name
- score
- strand

Peak
- _id
- result_id
- type
- chromosome
- start
- end
- name
- score
- strand
- signal_value
- p_value
- q_value
- peak

GridFS
- _id
- result_id

DFBS
Regulation of cRPGs (outer circle) and mRPGs (circle next from the outside).

Yellow circles: RPGs
Green diamonds: regulators
Icon size is proportional to the number of regulatory relationships.

Note the extreme density of regulatory relationships.