

Spring 4-15-2016

ChIPathlon: A competitive assessment for gene regulation tools.

Avi Knecht

University of Nebraska-Lincoln, avi@kurtknecht.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/ucareresearch>

 Part of the [Bioinformatics Commons](#), [Cancer Biology Commons](#), [Other Computer Sciences Commons](#), and the [Software Engineering Commons](#)

Knecht, Avi; Caprez, Adam; and Ladunga, Istvan, "ChIPathlon: A competitive assessment for gene regulation tools." (2016). *UCARE Research Products*. 76.

<http://digitalcommons.unl.edu/ucareresearch/76>

This Presentation is brought to you for free and open access by the UCARE: Undergraduate Creative Activities & Research Experiences at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in UCARE Research Products by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

ChIPathlon: a competitive assessment for gene regulation tools

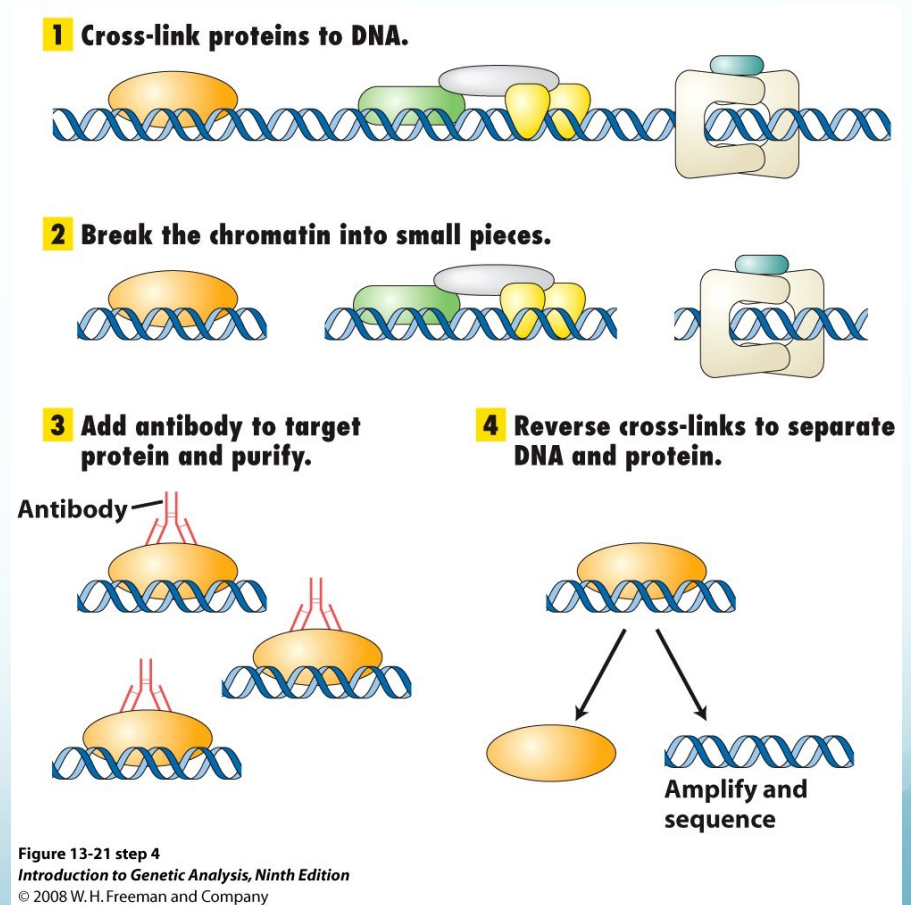
Avi Knecht, Adam Caprez, Istvan Ladunga

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

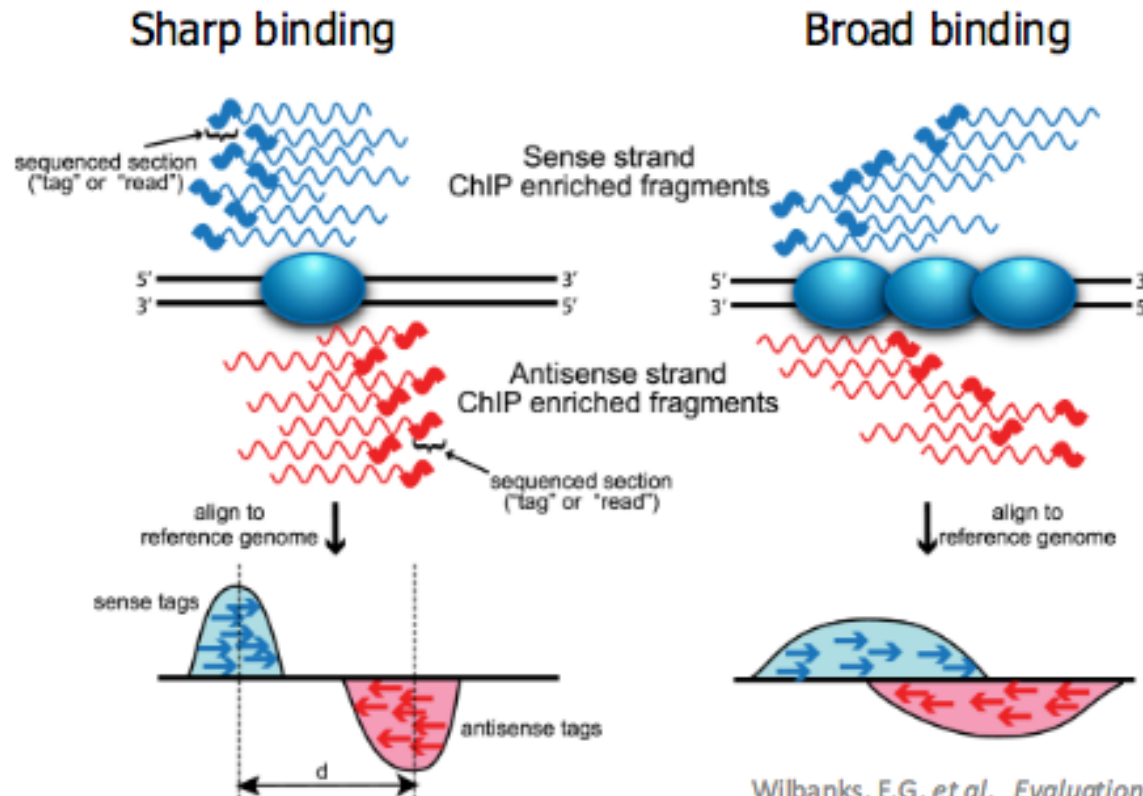
- Genes, are regulated by transcription factors and proteins, that bind to specific sequences on the DNA.
- Transcription factors are mapped to the DNA by chromatin immunoprecipitation followed by next-generation sequencing.



Critical steps in data analysis

Peak calling

Using strand dependent bimodality in peak calling



Wilbanks, E.G. et al. *Evaluation of Algorithm Performance in ChIP-Seq Peak Detection*. PLoS ONE July (2010)

Challenges in mapping transcription factors to the genome

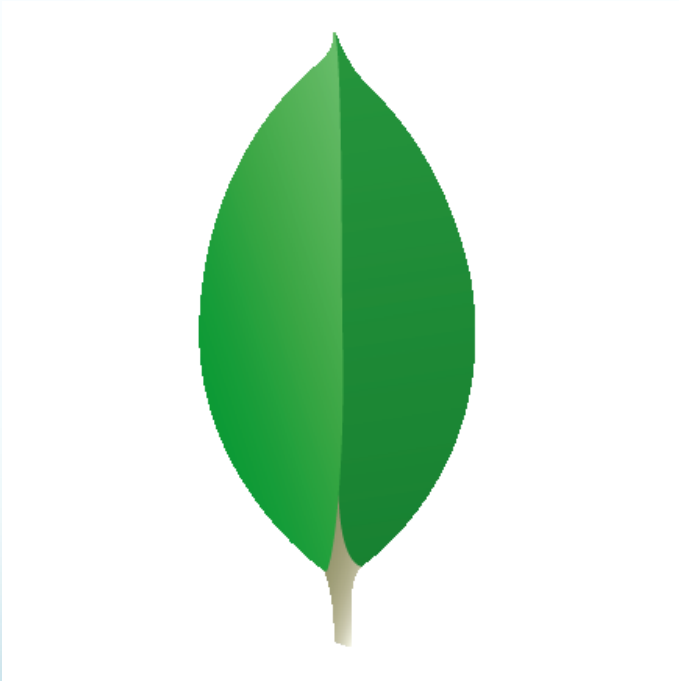
- Background correction is an open problem.
- DNA fragments can be much larger than the binding site.
- Sequencing read location does not follow any statistical distribution.
- Over 50 different methods are used for mapping, which produce different results.

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 I

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

```
1  zcat_awk_sort_peaks:
2    inputs:
3      - bed:
4          type: file
5    additional_inputs: null
6    outputs:
7      - bed:
8          type: file
9    command: zcat
10   arguments:
11     - "$inputs.0":
12         changeable: false
13         required: true
14         has_value: false
15     - "$outputs.0":
16         changeable: false
17         required: true
18         has_value: false
19   walltime: 2000
20   memory: 2000
21   cores: 1
22
```

YAML to Workflows II

Jobs are chained together by using module YAML files.

```
1 1 v peak_call:
2 2 v   - spp[tool]:
3 3 v     - r_spp_nodups:
4 4 v       inputs:
5 5 v         - exp.bed:
6 6 v           type: file
7 7 v         - control.bed:
8 8 v           type: file
9 9 v       additional_inputs: null
10 10 v     outputs:
11 11 v       - results.narrowPeak:
12 12 v         type: file
13 13 v       - results.pdf:
14 14 v         type: file
15 15 v       - results.ccscore:
16 16 v         type: file
17 17 v     - zcat_awk_sort_peaks:
18 18 v       inputs:
19 19 v         - results.narrowPeak:
20 20 v           type: file
21 21 v       additional_inputs: null
22 22 v     outputs:
23 23 v       - results_sorted.narrowPeak:
24 24 v         type: file
```

YAML to Workflows III

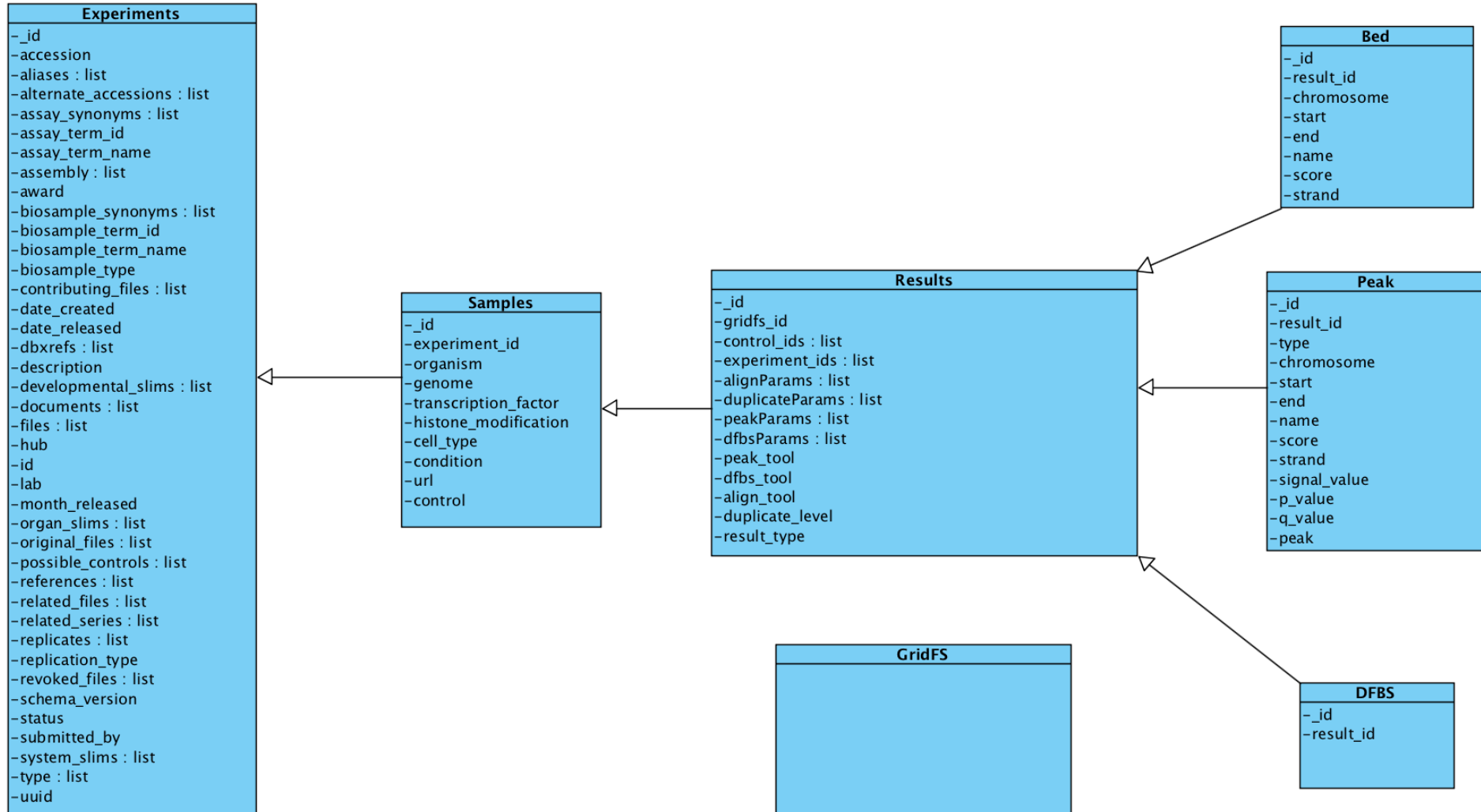
Users need to input a file selecting ENCODE experiment id's for files to process, tools they want to use, and a path to a genome.

```
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"
```

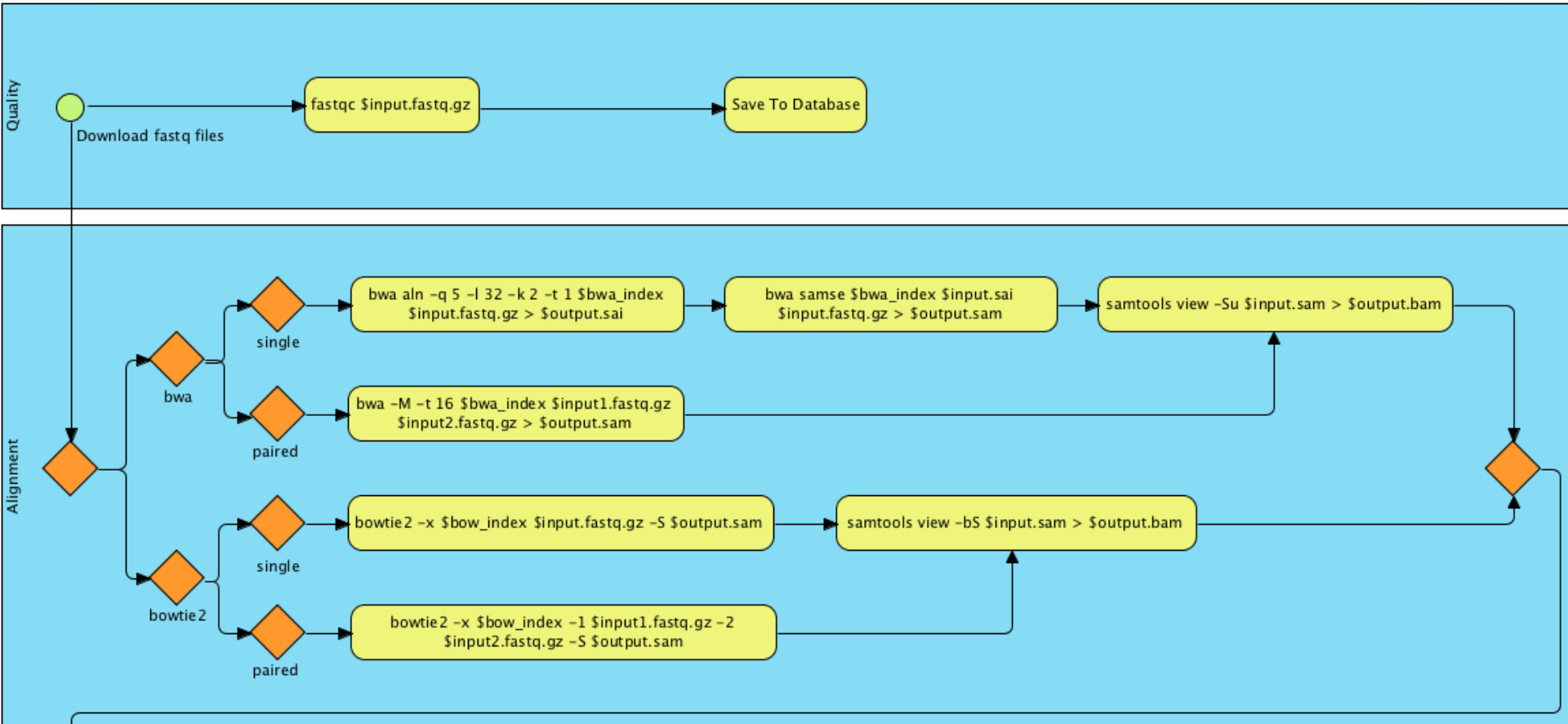
Conclusion

- 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.

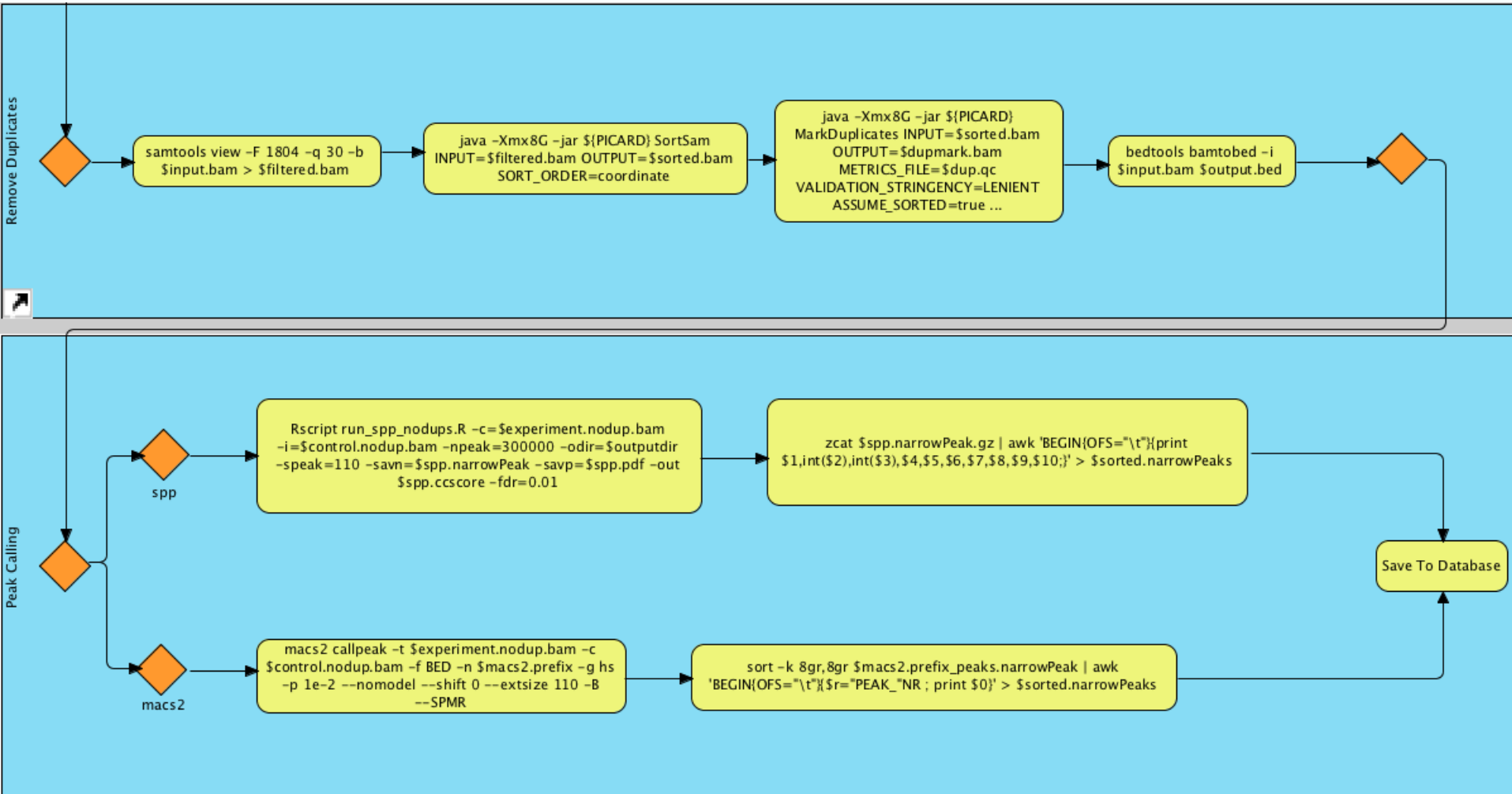
Database Architecture



Workflow I



Workflow II



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.

