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

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.

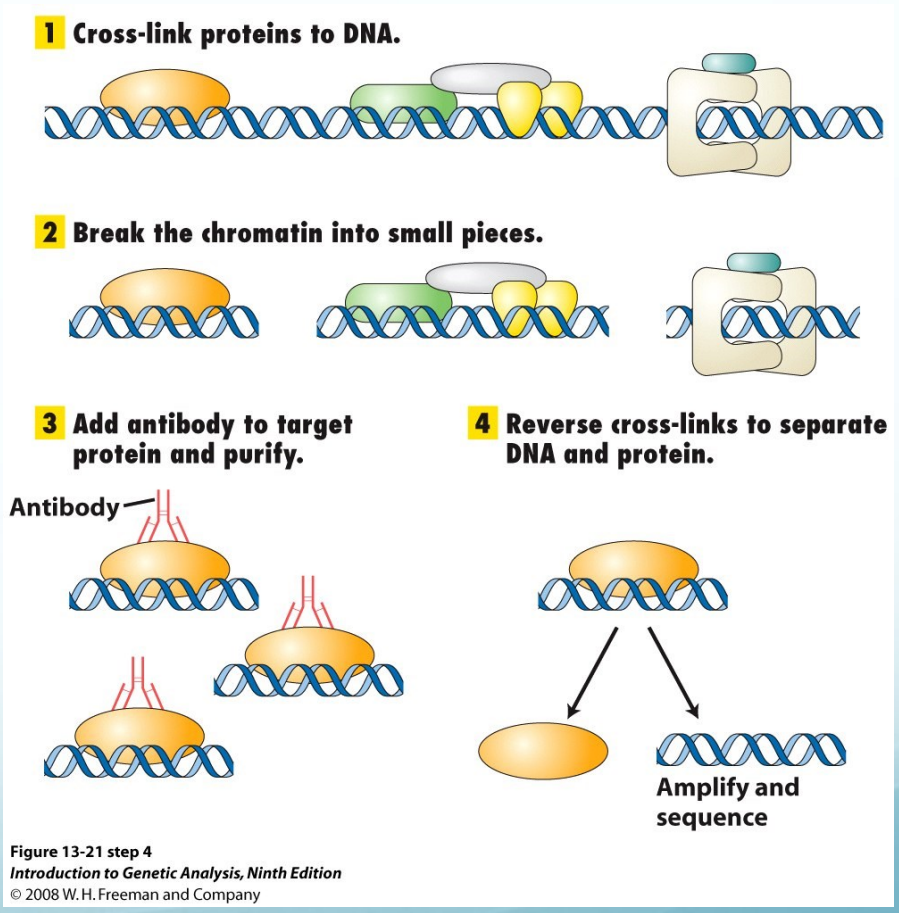
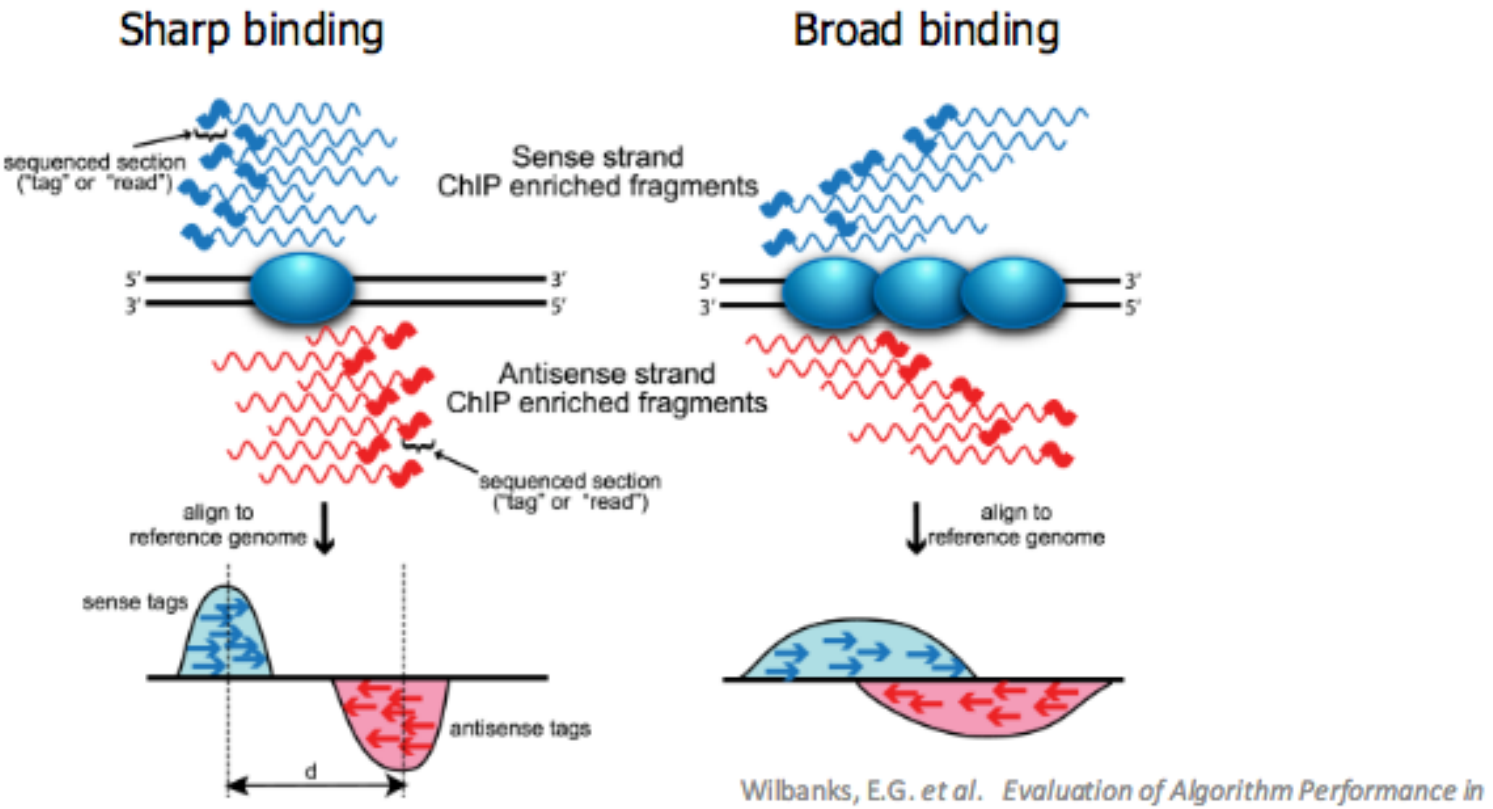


Figure 13-21 step 4
Introduction to Genetic Analysis, Ninth Edition

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.

```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
```

YAML to Workflows II

Jobs are chained together by using module YAML files.

```
peak_call:
  - spp[tool]:
    - r_spp_nodups:
        inputs:
          - exp.bed:
              type: file
          - control.bed:
              type: file
        additional_inputs: null
        outputs:
          - results.narrowPeak:
              type: file
          - results.pdf:
              type: file
          - results.ccscore:
              type: file
    - zcat_awk_sort_peaks:
        inputs:
          - results.narrowPeak:
              type: file
        additional_inputs: null
        outputs:
          - results_sorted.narrowPeak:
              type: file
```

YAML to Workflows III

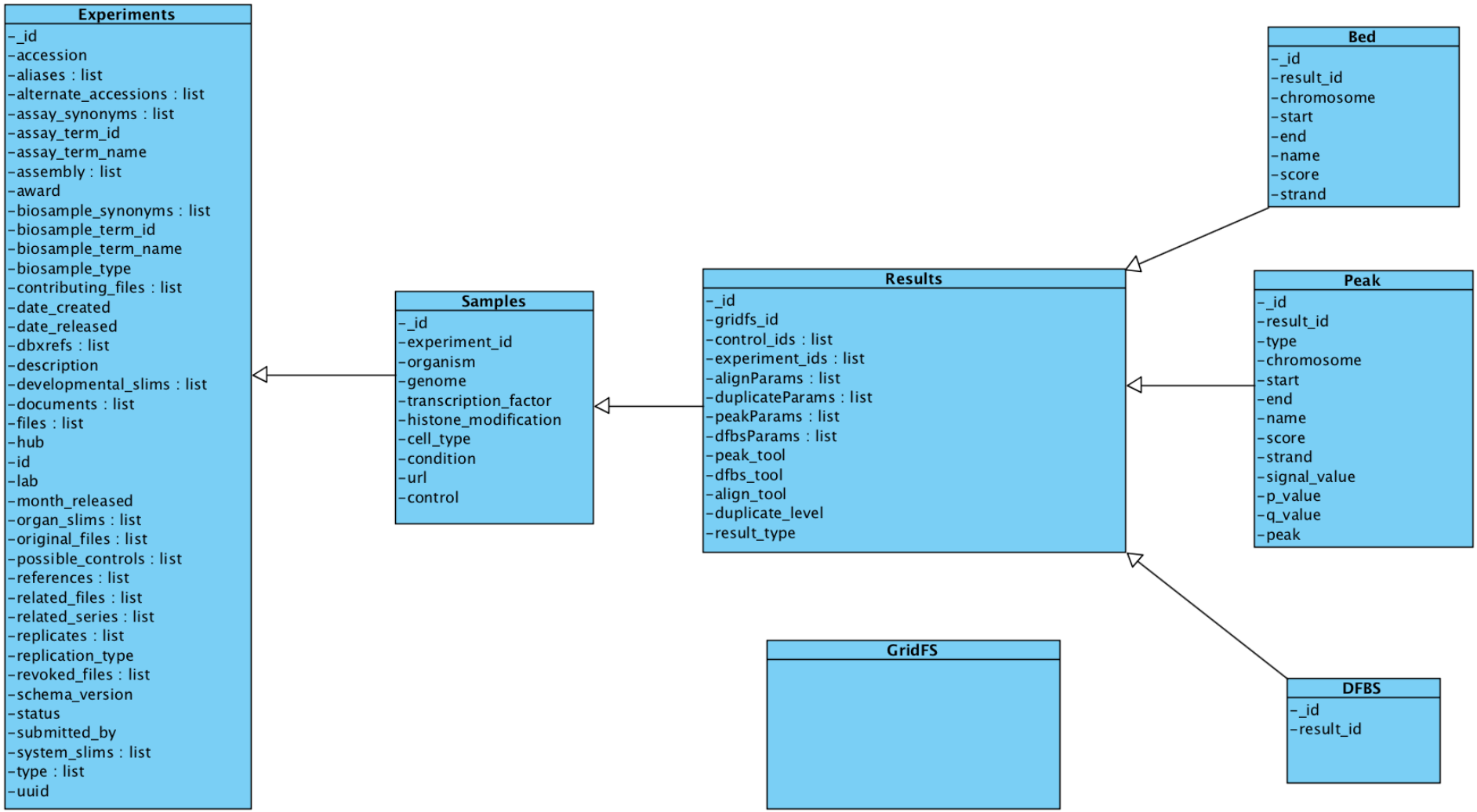
Users need to input a file selecting ENCODE experiment id's for files to process, tools they want to use, and a path to a genome.

```yaml
runs:
  - experiment: "ENCSR605MFS"
    align: bwa
    peak: spp
  - experiment: "ENCSR605MFS"
    align: bowtie2
    peak: spp
  - experiment: "ENCSR000ERE"
    align: bwa
    peak: spp
  - experiment: "ENCSR000EGZ"
    align: bowtie2
    peak: macs2
genomes:
  bwa:
    grch38p6: "/path/to/genome/base"
  bowtie2:
    grch38p6: "/path/to/genome/base"
```

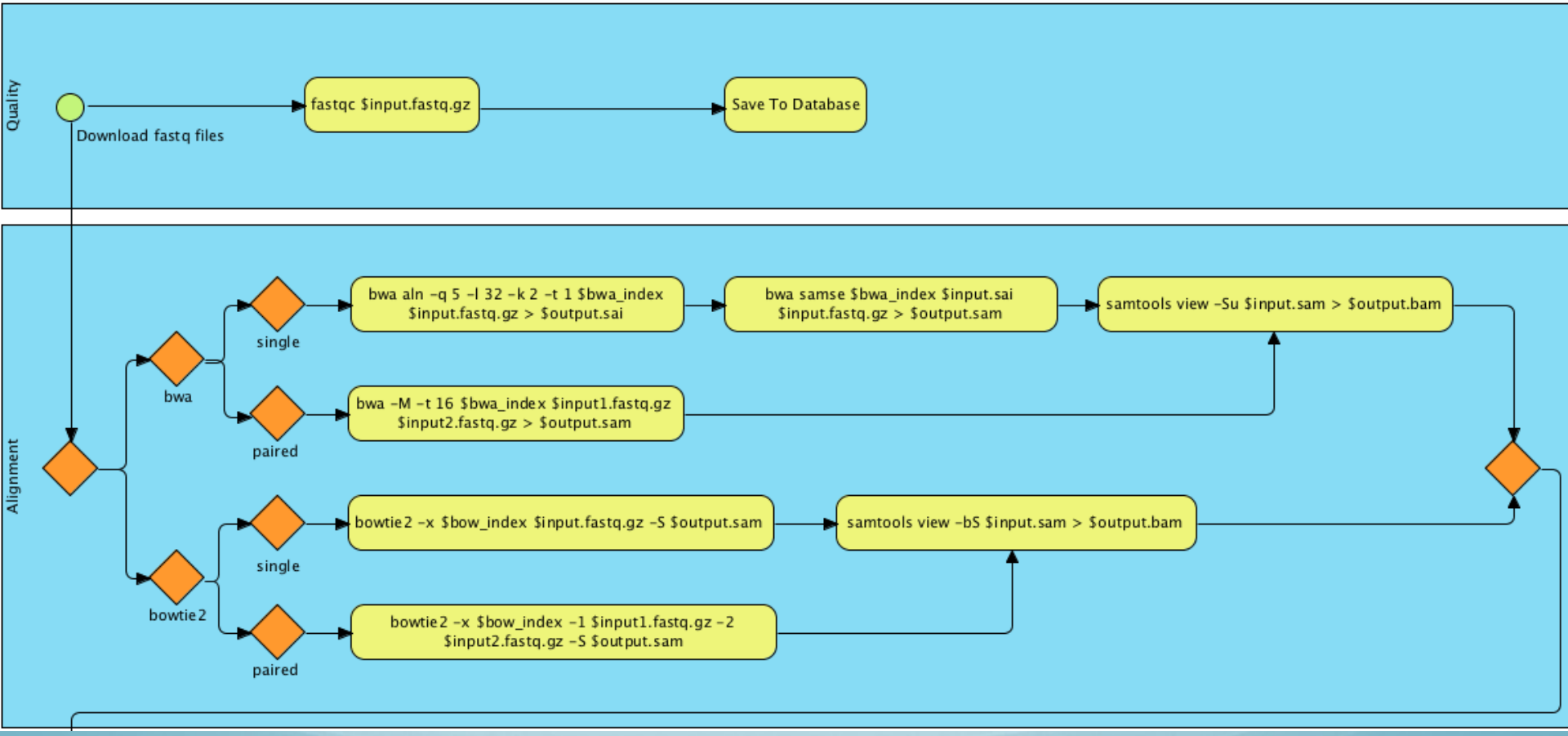
Conclusion

- 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.

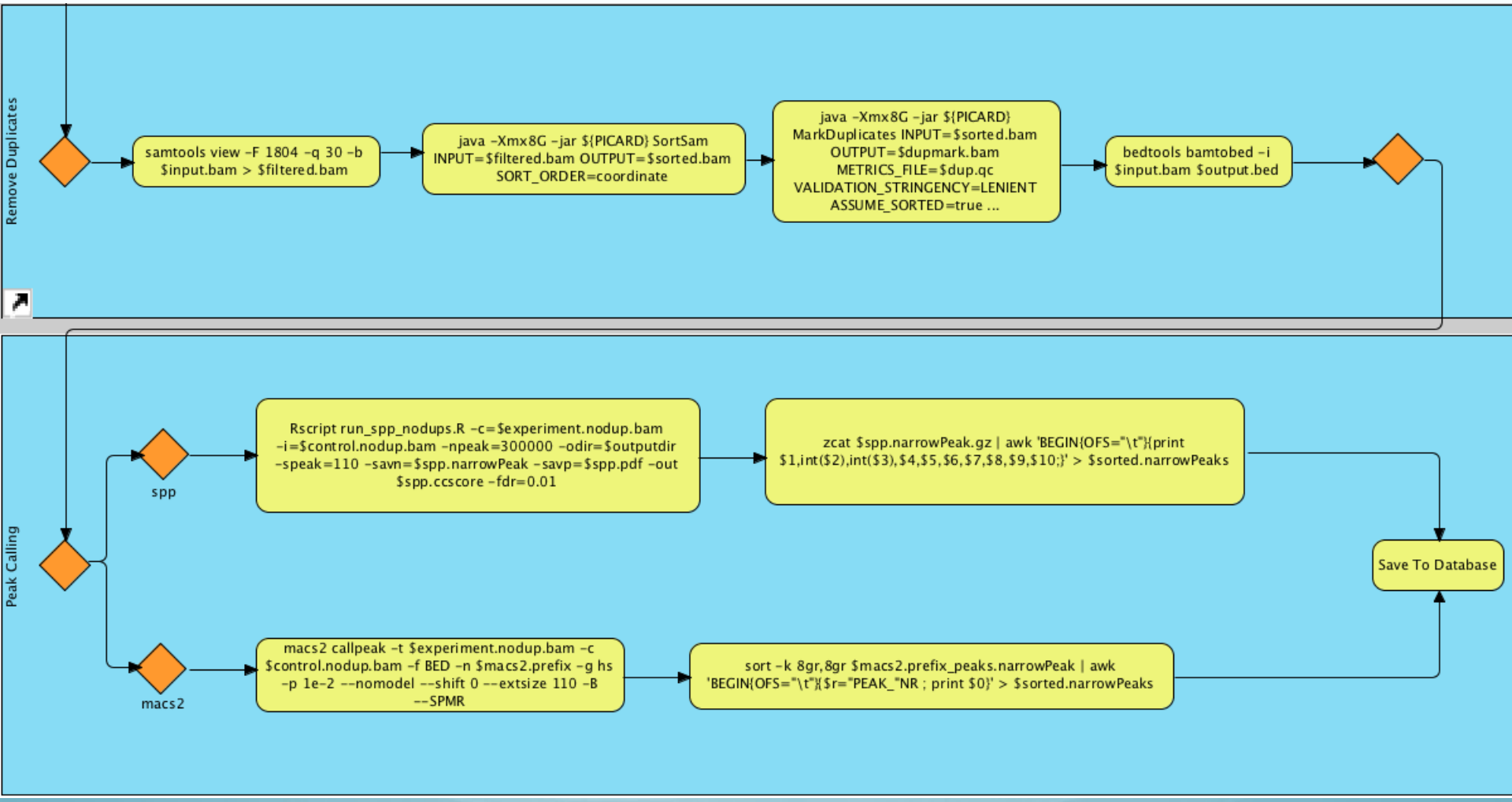
Database Architecture



Workflow I


Quality
Download fastq files
fastqc $input.fastq.gz
Save To Database
Alignment
bwa
single
bwa aln -q 5 -l 32 -k 2 -t 1 $bwa_index $input.fastq.gz > $output.sai
bwa samse $bwa_index $input.sai $input.fastq.gz > $output.sam
samtools view -Su $input.sam > $output.bam
paired
bwa -M -t 16 $bwa_index $input1.fastq.gz $input2.fastq.gz > $output.sam
bowtie2
single
bowtie2 -x $bow_index $input.fastq.gz -S $output.sam
samtools view -bS $input.sam > $output.bam
paired
bowtie2 -x $bow_index -1 $input1.fastq.gz -2 $input2.fastq.gz -S $output.sam

Workflow II


Remove Duplicates
samtools view -F 1804 -q 30 -b $input.bam > $filtered.bam
java -Xmx8G -jar ${PICARD} SortSam INPUT=$filtered.bam OUTPUT=$sorted.bam SORT_ORDER=coordinate
java -Xmx8G -jar ${PICARD} MarkDuplicates INPUT=$sorted.bam OUTPUT=$dupmark.bam METRICS_FILE=$dup.qc VALIDATION_STRINGENCY=LENIENT ASSUME_SORTED=true ...
bedtools bamtobed -i $input.bam $output.bed
Peak Calling
spp
Rscript run_spp_nodups.R -c=$experiment.nodup.bam -i=$control.nodup.bam -npeak=300000 -odir=$outputdir -speak=110 -savn=$spp.narrowPeak -savp=$spp.pdf -out $spp.ccscore -fdr=0.01
zcat $spp.narrowPeak.gz | awk 'BEGIN{OFS="\t"}{print $1,int($2),int($3),$4,$5,$6,$7,$8,$9,$10;}' > $sorted.narrowPeaks
Save To Database
macs2
macs2 callpeak -t $experiment.nodup.bam -c $control.nodup.bam -f BED -n $macs2.prefix -g hs -p 1e-2 --nomodel --shift 0 --extsize 110 -B --SPMR
sort -k 8gr,8gr $macs2.prefix_peaks.narrowPeak | awk 'BEGIN{OFS="\t"}{$r="PEAK_"NR ; print $0}' > $sorted.narrowPeaks

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.