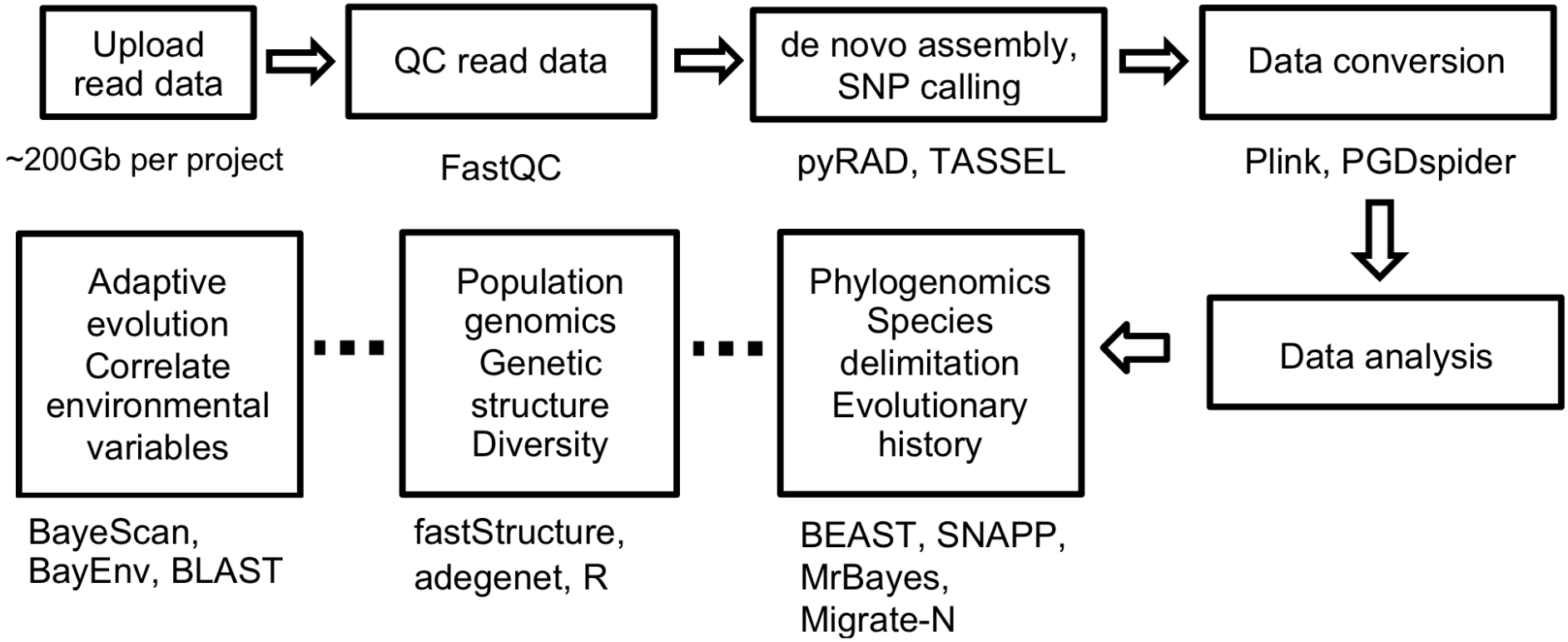
PAWSEY supercomputing centre

Managing Bioinformatics Software Stack on Magnus

Charlene Yang

- Associated projects
  - Kym Ottewell, Developing optimised workflows for phylogenomic and population genomic analyses of Australian native species
  - Grant Morahan, Enabling personalized medicine by predicting genetic signatures of disease

Overview



PAWSEY supercomputing centre

Overview

- Software list
  - FastQC, Plink, Structure, FastStructure, Stacks, Treemix
  - Migrate-n, MUSCLE, PyRAD, Beagle, Beast2, PGDSpider
- Scripts all working
  - Structure – job packing – 20 times faster
  - Stacks – OpenMP – 10 times faster [memory limit]
- No rigorous benchmarking
  - documentation on Portal/ script repository
  - a list of software with brief introduction, build logs, template job scripts and usage notes

Overview

- Compile GPU, MPI, OpenMP versions
  - Beagle, Beast2 – GPU
  - Migrate-n, Mothur – MPI
  - Stacks – OpenMP
- Look for parallel options
  - Plink-1.9 – flags that support parallel runs
  - –r/--r2, --distance, --genome, --make-rel, --make-grm-gz/--make-grm-bin, --epistasis, --fast-epistasis
- Extract parallelism in the workflow
  - Same task on different data/ different tasks without dependency
  - Job-packing/ Job array

Parallelism in workflow

- Structure-2.3.4:
  - a model-based clustering method for inferring population structure using genotype data consisting of unlinked markers
  - structure -m mainparams -e extraparams -K $K_value >> output_$K_value
- Problem:
  - each run scans a range of K's, K=1-15
  - repeat each run for 20 times to get an average
- Attempt 1:
  - pack 15 K's onto the same node using a wrapper
  - repeat the run on 20 nodes but using different random numbers

Parallelism in workflow

- cyang@galaxy-2:/scratch/director100/cyang/> cat job_script
- #!/bin/bash -l
- #SBATCH --account=director100
- #SBATCH --partition=workq
- #SBATCH --time=01:00:00
- #SBATCH --nodes=1
- #SBATCH --export=NONE
- #SBATCH --array=0-19

- module use /ivec/cle52/galaxy/modulefiles/bio-apps
- module load structure
- cd /scratch/director100/cyang/help_user/structure/para
- mkdir output-$SLURM_JOB_ID
- aprun -n 15 -N 15 ./multirun.sh

Parallelism in workflow

- cyang@galaxy-2:/scratch/director100/cyang/help_user/structure/para> cat multirun.sh
- #!/bin/bash
- K_index=$((ALPS_APP_PE+1))
- echo "I'm PE $ALPS_APP_PE and K $K_index" > output-$SLURM_JOB_ID/cleomek-$K_index
- structure -m mainparams -e extraparams -K $K_index >> output-$SLURM_JOB_ID/cleomek-$K_index

- On each node, multirun.sh will spawn 15 instances of structure on 15 cores, running a different K on each core, writing to a different file named cleomek-$K_index in the same folder output-$SLURM_JOB_ID
- On different nodes, multirun.sh creates different folders named output-$SLURM_JOB_ID

Parallelism in workflow

- Issues:
  - different runs run for different length of time
  - random number generators generating the same sequence for the same node
- Attempt 2:
  - run same K's on same nodes for 20 times but with different random numbers
  - run different K's on different nodes to cover the K range of 1-15

Parallelism in workflow

- cyang@galaxy-1:/scratch/director100/cyang/> cat job_script
- #!/bin/bash -l
- #SBATCH --account=director100
- #SBATCH --partition=workq
- #SBATCH --time=00:10:00
- #SBATCH --nodes=1
- #SBATCH --export=NONE
- #SBATCH --array=0-14

- module use /ivec/cle52/galaxy/modulefiles/bio-apps
- module load structure
- cd /scratch/director100/cyang/help_user/structure/new_para
- mkdir output-$SLURM_ARRAY_JOB_ID
- aprun -n 20 -N 20 ./multirun.sh

Parallelism in workflow

- cyang@galaxy-1:/scratch/director100/cyang/help_user/structure/new_para> cat multirun.sh
- #!/bin/bash
- sed -i 's/#define RANDOMIZE 1/#define RANDOMIZE 0/g' extraparams
- SEED_value=$((ALPS_APP_PE+1))
- K_value=$((SLURM_ARRAY_TASK_ID+1))
- echo "I'm Task $SLURM_ARRAY_JOB_ID Subtask $SLURM_ARRAY_TASK_ID PE $ALPS_APP_PE, using K $K_value SEED $SEED_value" > output-$SLURM_ARRAY_JOB_ID/cleomek-K$K_value-PE$ALPS_APP_PE
- structure -m mainparams -e extraparams -D $SEED_value -K $K_value >> output-$SLURM_ARRAY_JOB_ID/cleomek-K$K_value-PE$ALPS_APP_PE

Parallelism in workflow

- On each node, multirun.sh will spawn 20 instances of structure on 20 cores, running the same K on all 20 cores, each writing to a different file named cleomek-K$K_value-PE$ALPS_APP_PE in the same folder output-$SLURM_ARRAY_JOB_ID
- On different nodes, multirun.sh writes different files suffixed by environment variable $ALPS_APP_PE, ranging from 0 to 300

- Cores on a node are running the same K – finishing at the same time
- SEED number is changed to $ALPS_APP_PE+1 so that each core is using different random numbers for the same K

File I/O

- cyang@galaxy-2:/group/bppp006/syoung/LIBSVM/libsvm-3.18> cat testnew.slurm
- #!/bin/bash -l
- #SBATCH --account=bppp006
- #SBATCH --partition=workq
- #SBATCH --nodes=1
- #SBATCH --time=05:00:00
- #SBATCH --export=NONE

- module use /ivec/cle52/galaxy/modulefiles/bio-apps
- module load plink/1.9

- aprun -n 1 -N 1 ./testnew.sh

File I/O

- cyang@galaxy-2:/group/bppp006/syoung/LIBSVM/libsvm-3.18> cat testnew.sh
- #!/bin/bash
- file=$1
- total_lines=$2
- total_PEs=$3
- lines_per_PE=$4
- index_start=$(($ALPS_APP_PE*$lines_per_PE+1))
- index_end=$(($(($ALPS_APP_PE+1))*$lines_per_PE))

- for marker in `awk "NR>= $index_start && NR<= $index_end" $file`
- Do
- sed "1 i\\$marker" thebest${previous}snp_chisq.txt > ${marker}.txt
- plink --bfile 1train --extract ${marker}.txt --make-bed --out $marker-trainsubsnps
- sed -i '1d' $marker-tmp.raw
- cat $marker-tmp.raw | sed 's/NA/0/g' | sed 's/3/0/g' | cut -d ' ' -f7- > $marker-geno.txt
- rm -rf $marker-tmp.*
- …….

File I/O

- …….
- ./dcln2libsvm $marker-subsnps_CASE $marker-subsnps_CONTROL $marker-subsnps_CASE_TEST $marker-subsnps_CONTROL_TEST > $marker-svmtmp
- cat $marker-svmtmp | grep "train" | cut -d' ' -f2- > $marker-train_data
- ./svm-train -s 0 -t 0 $marker-train_data > $marker-train_data.model
- echo `./svm-predict $marker-test_data $marker-train_data.model` ${marker} > linearRes_${marker}.txt
- rm -rf $marker-train_data.model
- ……..
- rm -rf $marker-train_data $marker-test_data $marker-train_data.model $marker-svmtmp linearRes_${marker}.txt polyRes_${marker}.txt rbfRes_${marker}.txt Re${marker}.txt ${marker}.txt sorted${marker}.txt ${marker}-subsnps*
- done

File I/O

- cat: tmp: No such file or directory
- ERROR: fscanf failed to read model
- ERROR: fscanf failed to read model
- can't open model file train_data.model
- can't open model file train_data.model
- warning: No negative true label.
- warning: No positive true label.
- warning: Too few postive true labels or negative true labels
- warning: No positive predict label.
- warning: No postive true label.
- warning: Divide by zero in MCC calculation.

Gotchas

- Random number generator
  - Different processes/threads using different seeds
- File I/O
  - Different processes/thread reading/writing different files/directories
- Memory limit per node
  - Different datasets may need different parallelism

Gotchas

- Consistency in
  - export OMP_NUM_THREADS=20 (after module swap)
  - aprun -n1 -d20
- Load correct PrgEnvs/compilers
- Check path/pythonpath
  - distruct.py -K 2 --input=Callifilt5_output --output=Callifilt5_K2distruct --popfile=callipops.txt
  - Pyrad being installed to solve the 'Could not find sortandcheck2' error.

To-dos

- Continue with the workflow
- Checkpointing – DMTCP, BLCR, Nectar
- Memory limit – Zeus [GPU codes]

- Documentation – script repository

Thank you!