Reads2Map

Practical and reproducible workflows to build polyploid linkage maps from sequencing data

Cris Taniguti
Lucas Taniguti
Jeekin Lau
Augusto Garcia
Marcelo Mollinari
David Byrne
Oscar Riera-Lizarazu

TOOLS FOR POLYPLOIDS
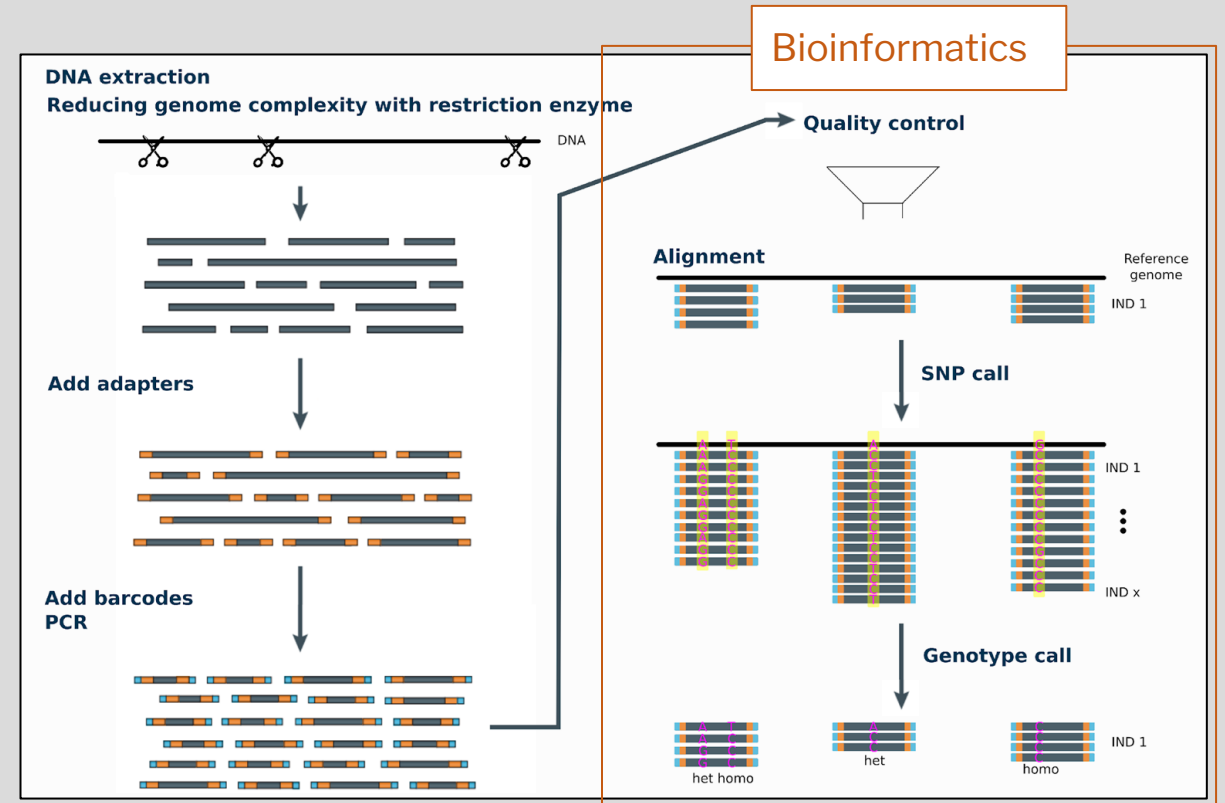
TEXAS A&M UNIVERSITY

Motivation
A genotyping-by-sequencing story

- My first GBS data (2015)
  - Biology background

Motivation

A genotyping-by-sequencing story

- ## My first GBS data (2015)
  - Biology background
  - File formats

FASTQ

VCF

Motivation

A genotyping-by-sequencing story

- My first GBS data (2015)
  - Biology background
  - File formats
  - High Performance Computing
  - Linux
  - Surviving without sudo permissions

Personal computer

core
core
core
core
core
core
core

4GB RAM; 8 cores; 1 node

High Performance Computing

384GB; 48 cores per node; 900 nodes

TEXAS A&M UNIVERSITY

Motivation

A genotyping-by-sequencing story

- ## My first GBS data (2015)
  - Biology background
  - File formats
  - High Performance Computing
  - Linux
  - Surviving without sudo permissions
  - Many software and parameters
  - Many programming languages
  - Different Operational Systems
  - Updates

TOOLS FOR POLYPLOIDS

TEXAS A&M UNIVERSITY®
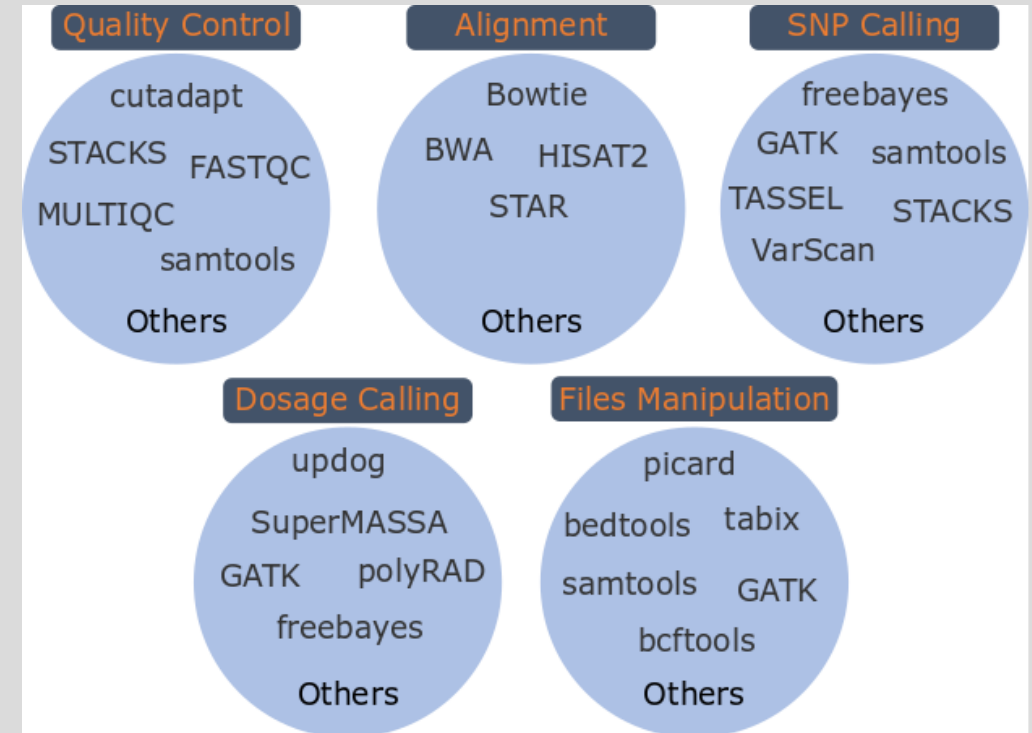
Motivation

A genotyping-by-sequencing story

- My first diploid outcrossing population linkage map
  - Estimating also phases
  - Hidden Markov Model multipoint approach
  - OneMap
  - R

- Diploid species
- Bi-parental populations
- Backcross, RILs, F2 and outcrossing
- Biallelic and Multiallelic markers

Recombination fraction matrix

markers (y-axis)

markers (x-axis)

rf
- 0.5
- 0.4
- 0.3
- 0.2
- 0.1
- 0.0
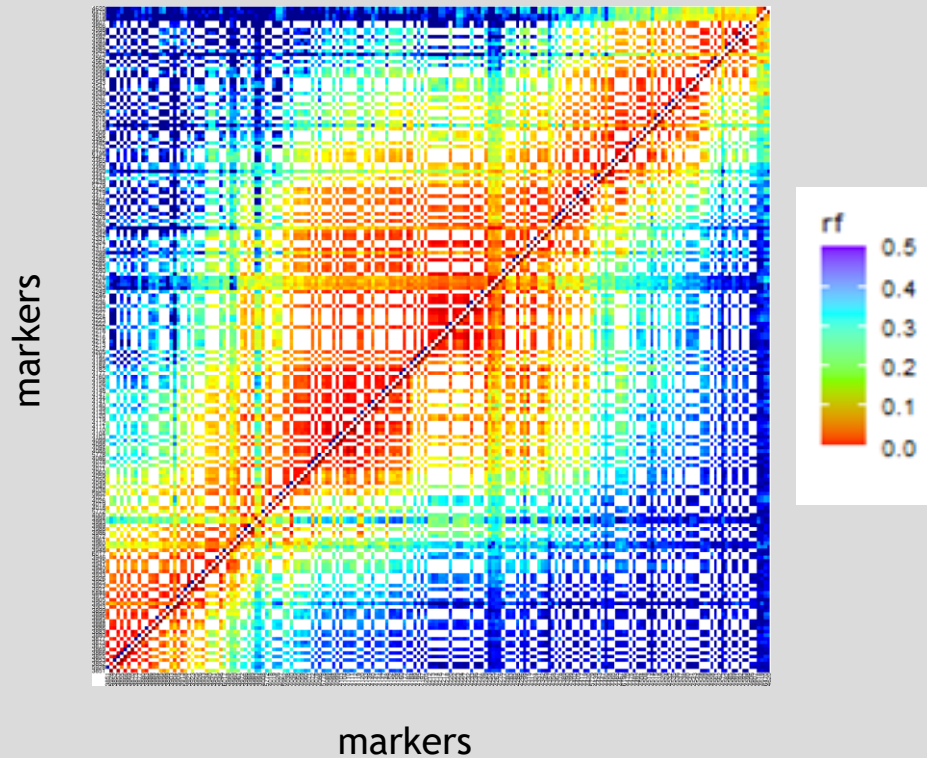
- # recombinations between two markers during meiosis

- Genomic order
  - Example: Eucalyptus chromosome 10

Recombination fraction matrix

Inversion

- Aspen chromosome 12

Most likely bad data
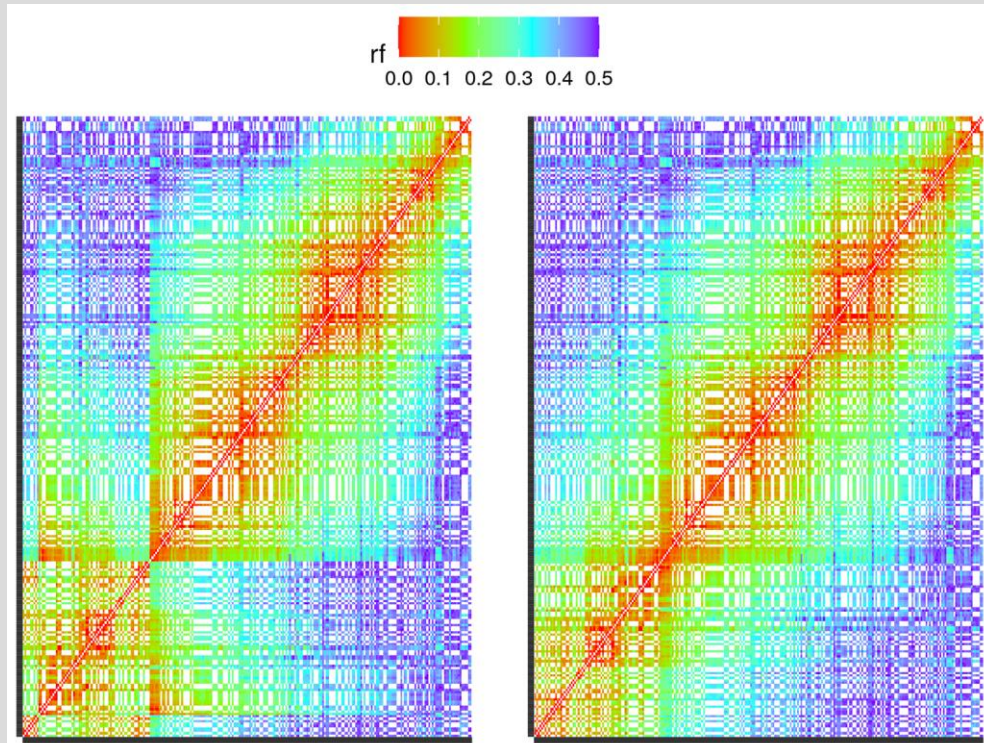
Genomic order

Order fixed
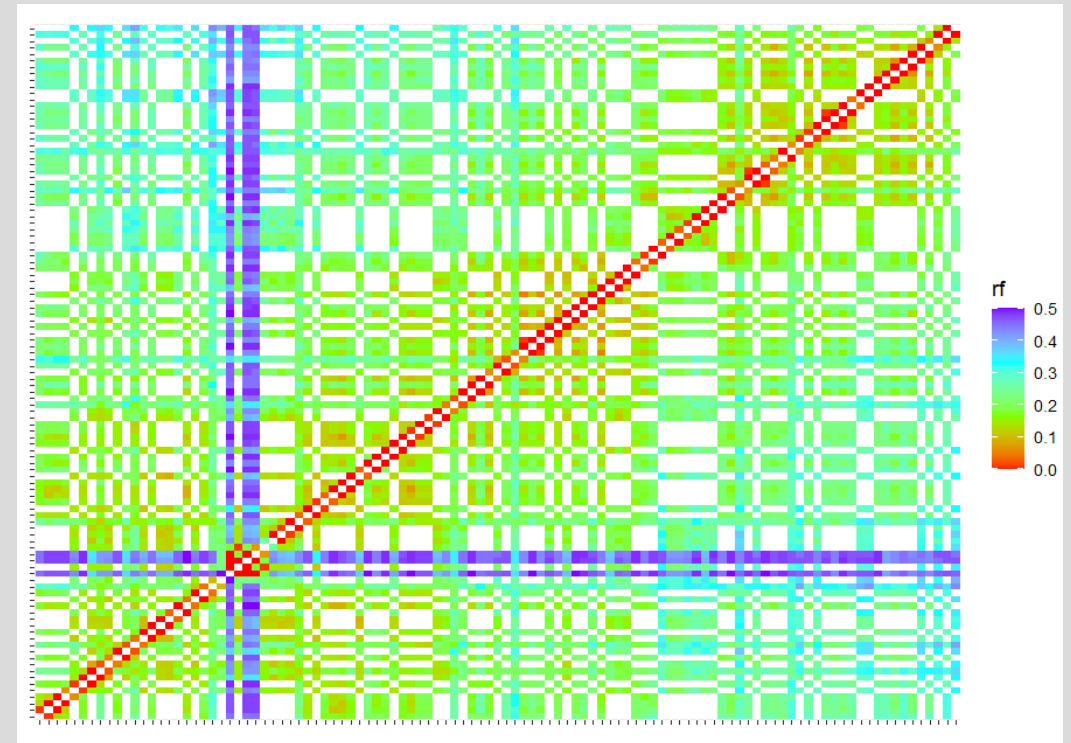
TOOLS FOR
POLYPLOIDS

TEXAS A&M
UNIVERSITY.

Motivation
Experiences building Linkage Maps

- Using sequencing markers
  - Recombination fraction matrix not always pretty
  - Slow analysis
  - Inflated map size

- OneMap - Since 2007

- OneMap maintainer - Since 2017

OneMap

3.0

- Updates in version 3.0

Motivation

Experiences building Linkage Maps

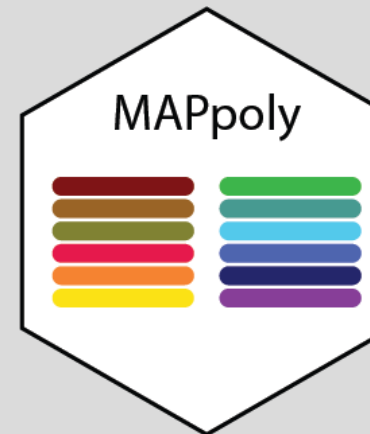- ## Using sequencing markers
  - Recombination fraction matrix not always pretty
  - Slow analysis
  - Inflated map size

- ## OneMap - Since 2007

- ## OneMap maintainer - Since 2017

- ## MAPpoly – Since 2018

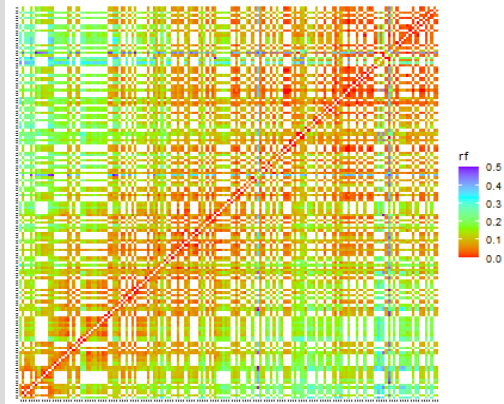**OneMap**

3.0

- Updates in version 3.0

MAPpoly

- Diploid and polyploid species
- Bi-parental populations
- Outcrossing
- All dosages markers
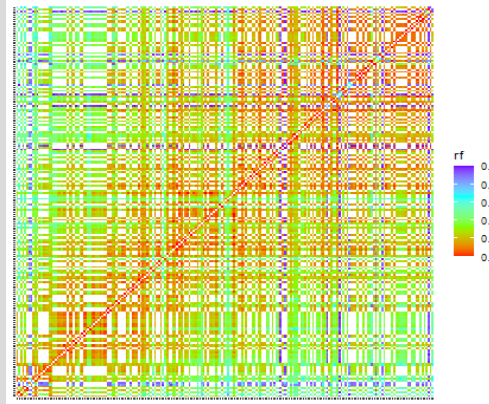- Updates – Marcelo and Gabriel talk

Diploid roses
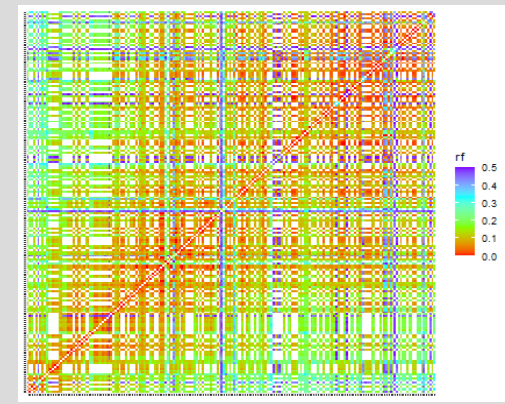
- 37% of chromosome 1

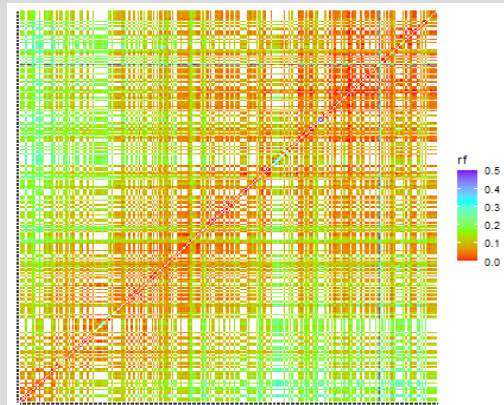- ~ 38 cM

- Sequencing depth ~ 94X

OneMap

GATK

GATK + polyRAD

GATK + updog

freebayes

freebayes + polyRAD
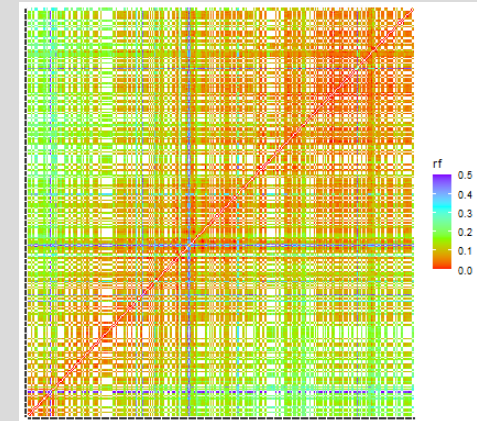
freebayes + updog

GATK + updog
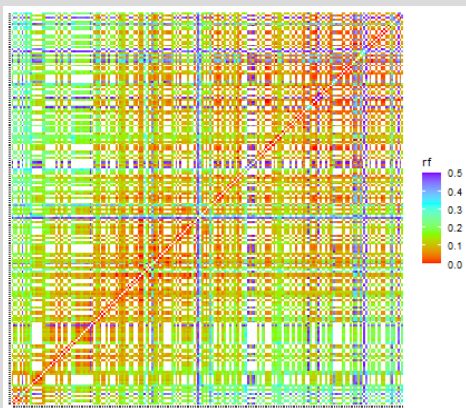
GATK + polyRAD

freebayes + polyRAD
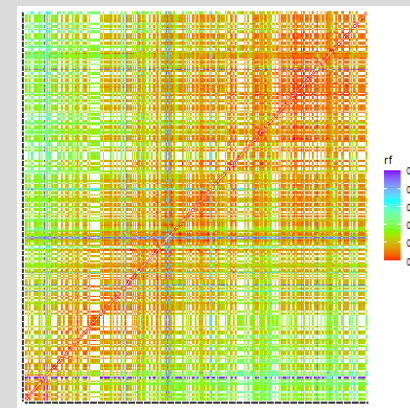
freebayes + updog

Simulation studies: 10,880 maps
Empirical studies: 816 maps

Filters:
- genotype probabilities
- non-informative markers
- replace AD by missing when GT is missing
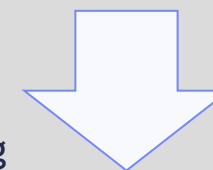
GATK + updog

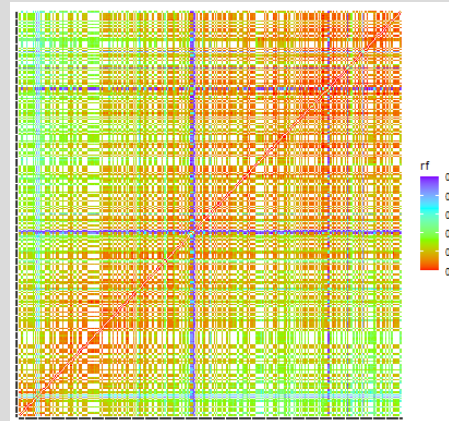GATK + polyRAD
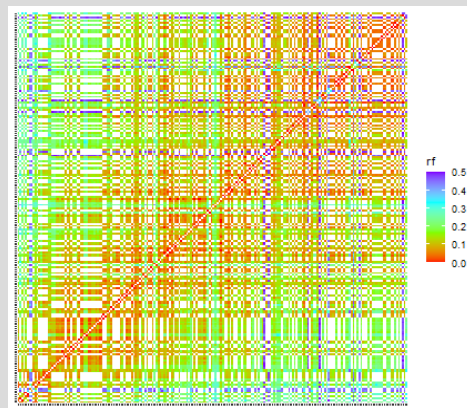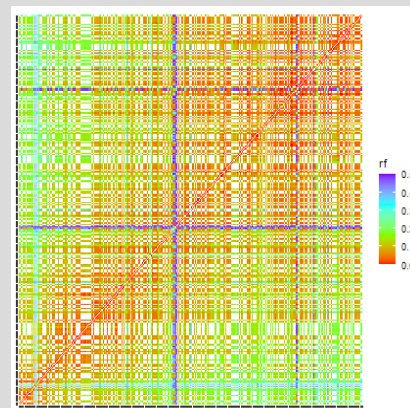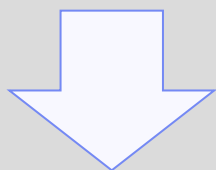
freebayes + polyRAD

freebayes + updog

TOOLS FOR POLYPLOIDS

TEXAS A&M UNIVERSITY
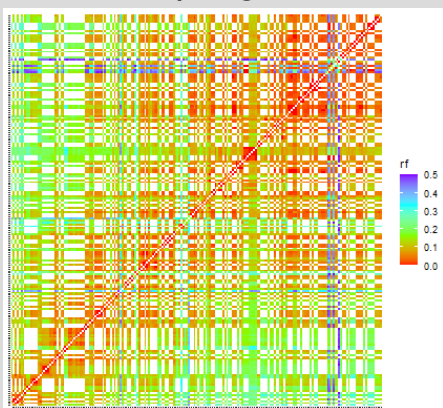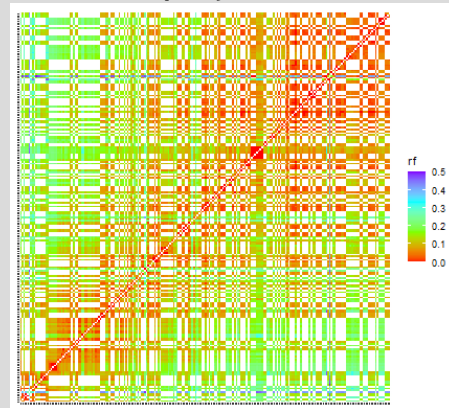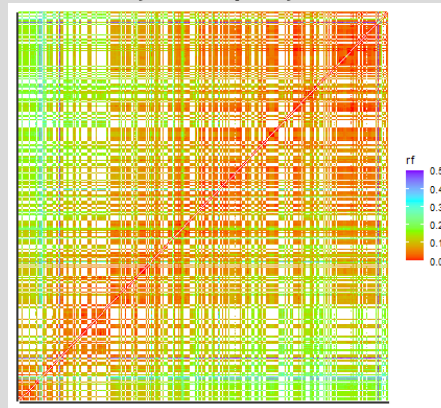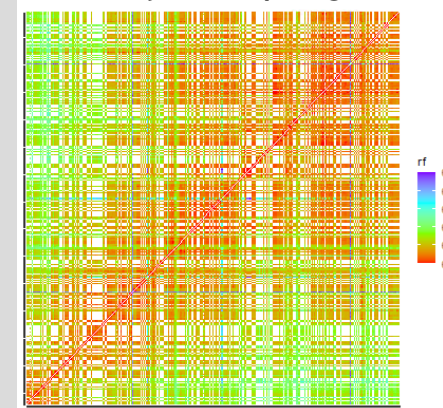
Map size
37% of chromosome 1 ~38cM
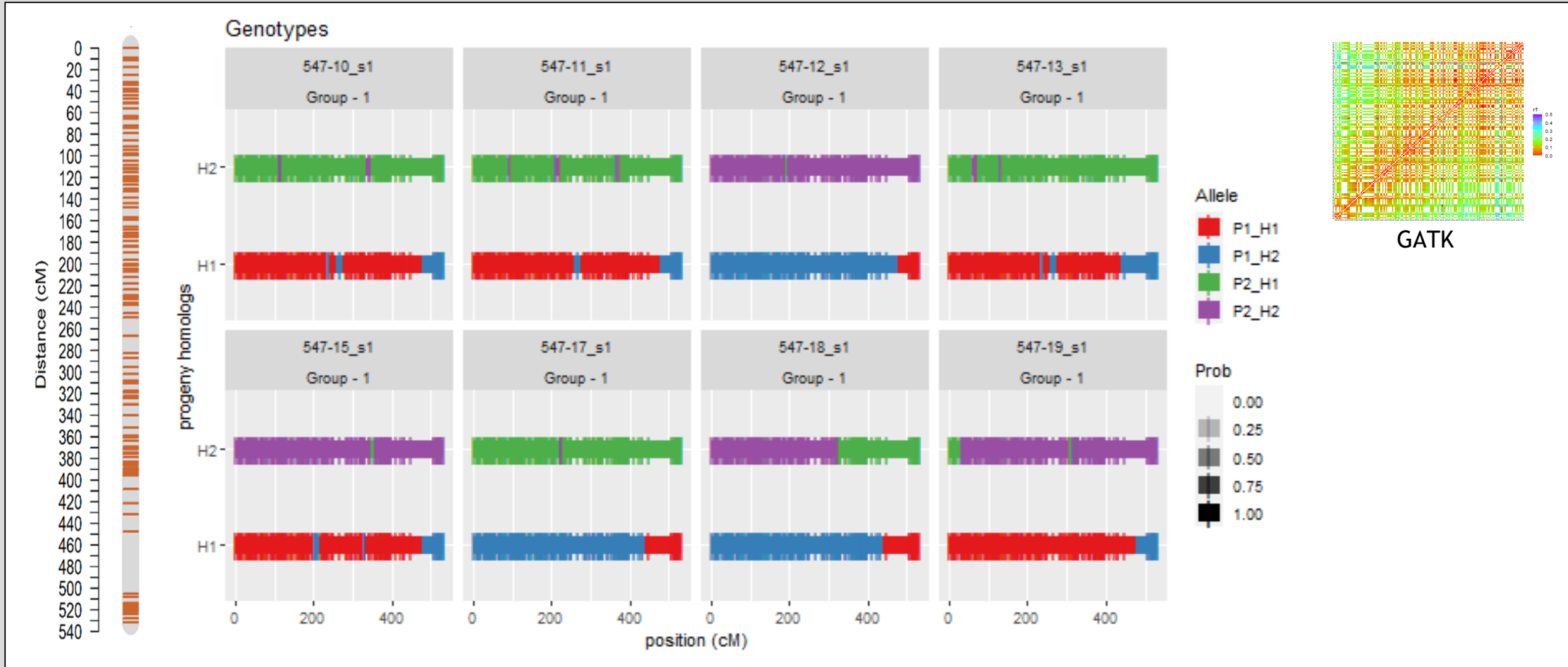
Map size

- Hidden Markov Model Emission Function
  - global error rate
  - genotype probabilities (PL)

red: changed = 0.25%

green: imputed = 1.1%

black: unchanged = 98.65%

Figure by Jeekin Lau

TOOLS FOR POLYPLOIDS

TEXAS A&M UNIVERSITY
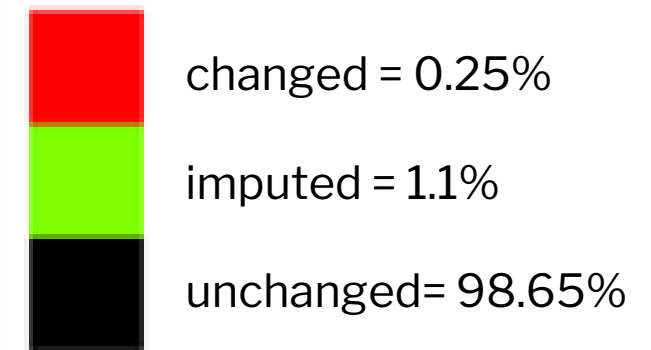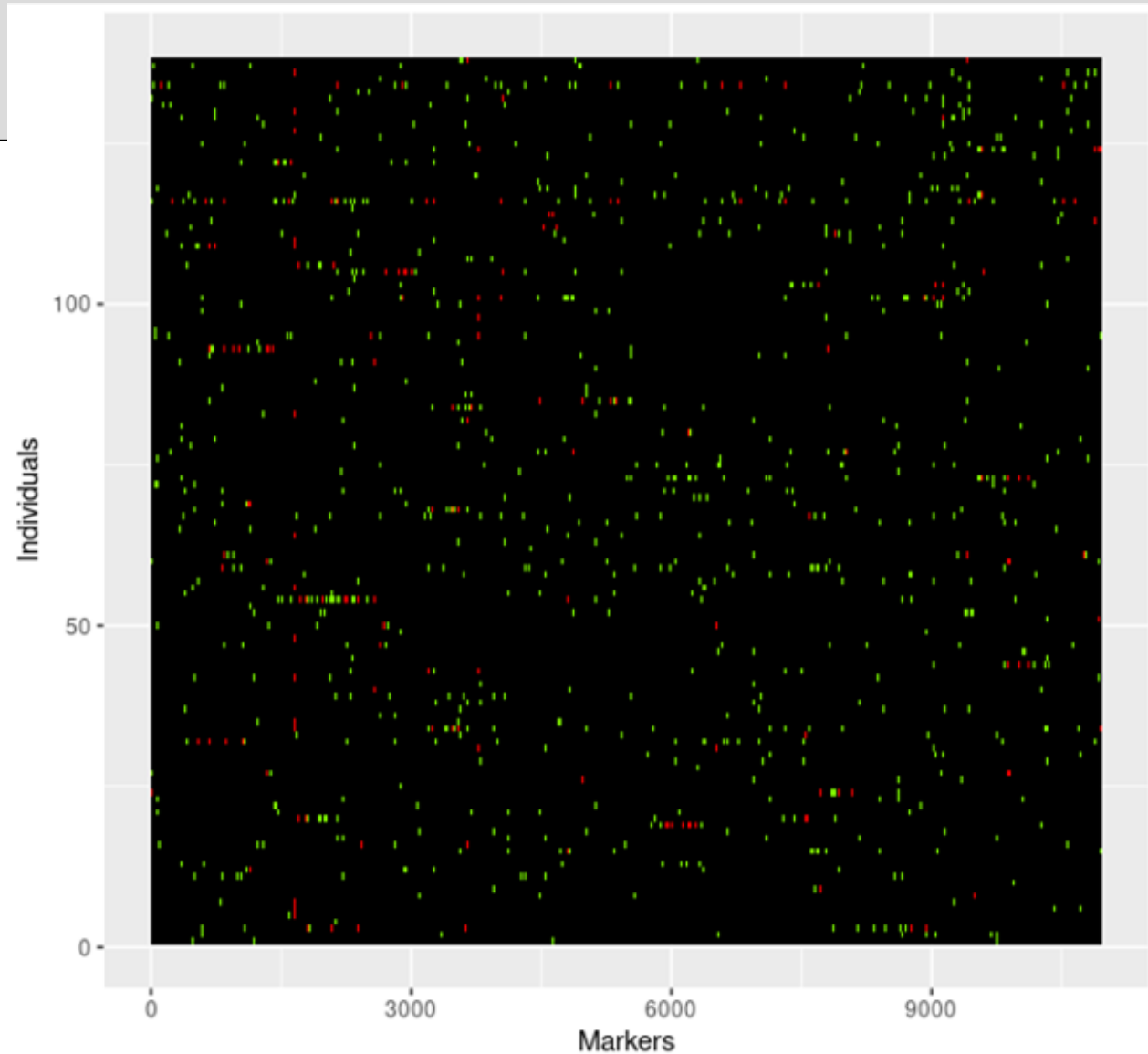
Map size
37% of chromosome 1 ~38cM

- Hidden Markov Model Emission Function
  - global error rate
  - genotype probabilities (PL)

Map size
37% of chromosome 1 ~38cM

TOOLS FOR POLYPLOIDS

TEXAS A&M UNIVERSITY
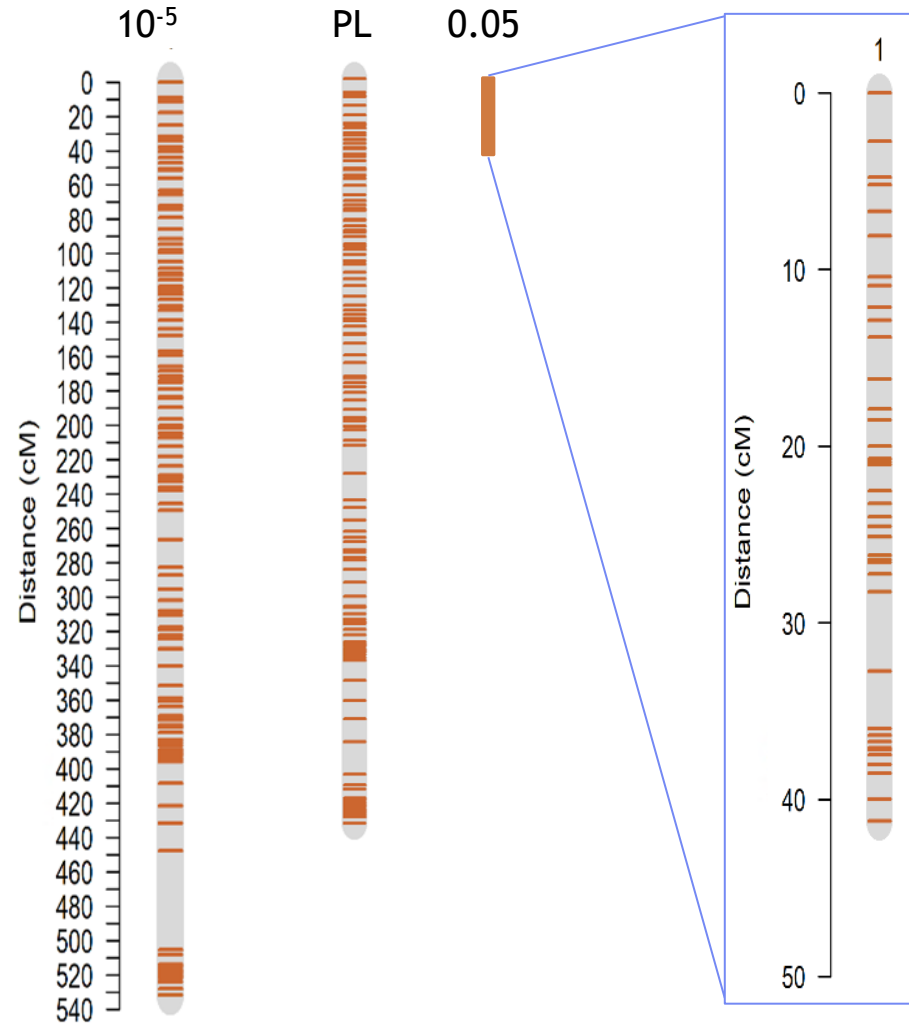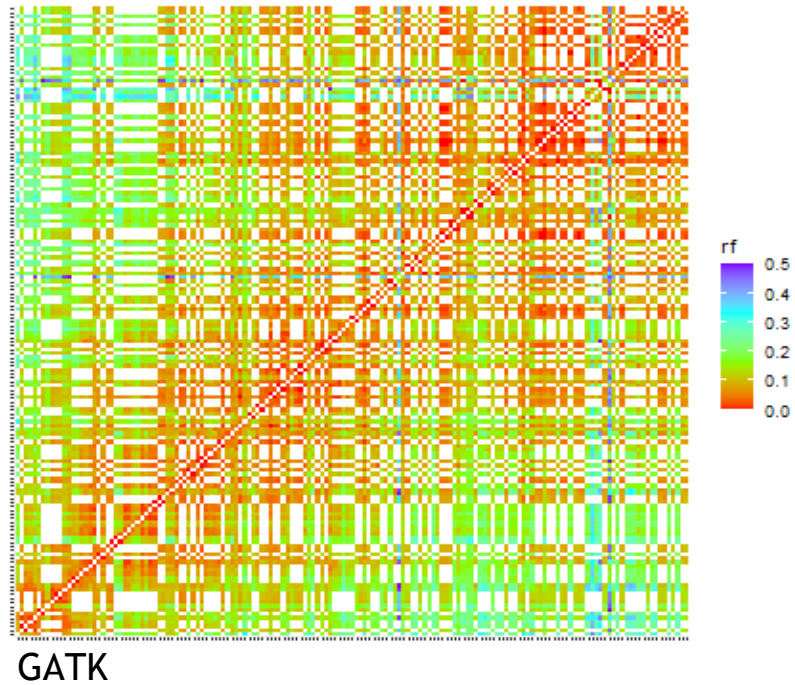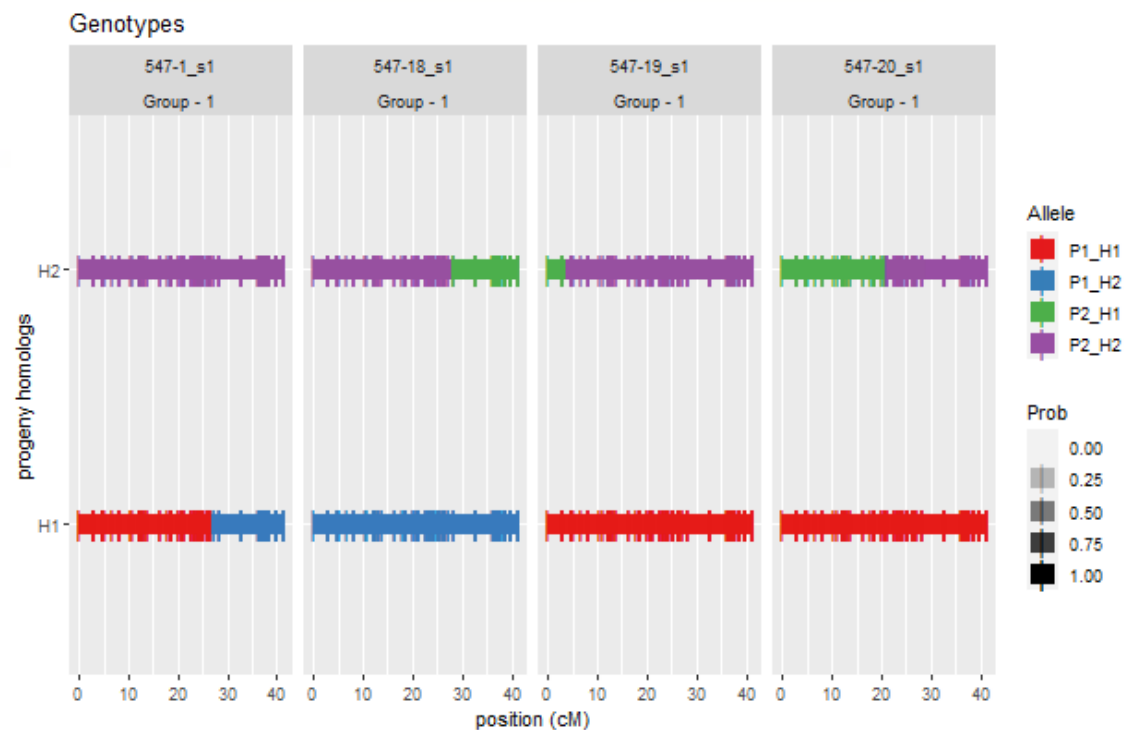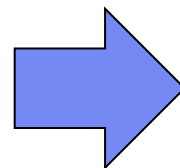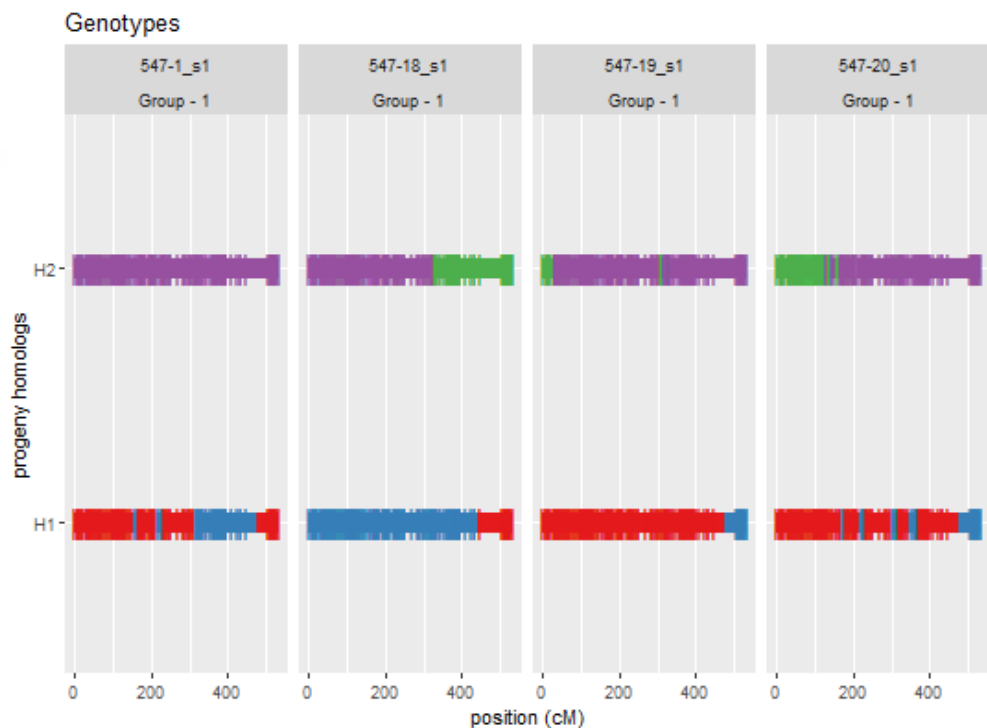
Simulation study

- Match recombinations breakpoints
  - Large maps - always bad
  - Small maps - not always good

- Other tested scenarios with:
  - Segregation distortion
  - Contaminants samples
  - Multiallelic markers
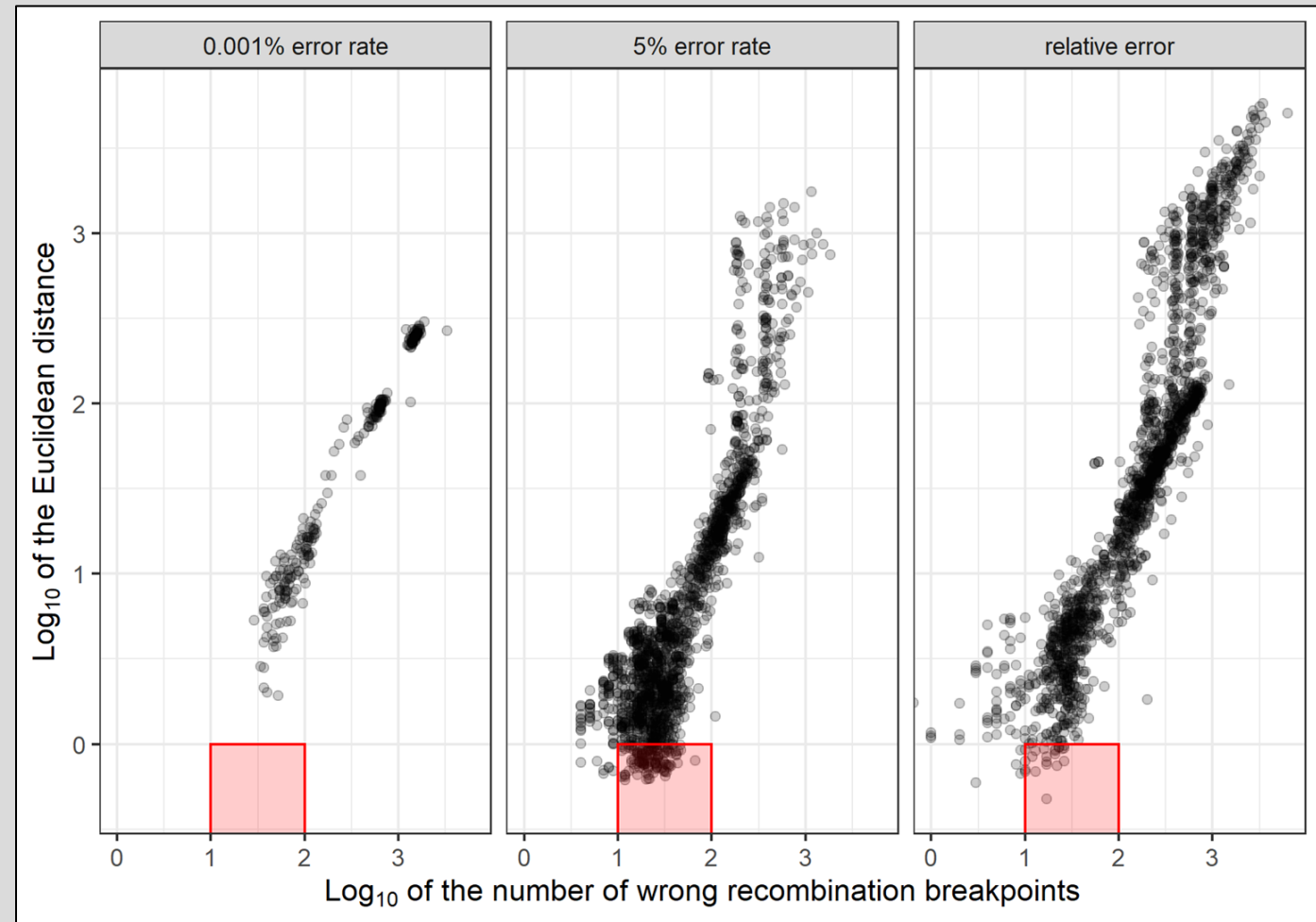
Red square: no inflated size (1 or less Euclidean distance) but have from 10 to 100 wrong recombination breakpoints

Preprint

Diploid Aspen

- 37% of chromosome 10

- Sequencing depth ~ 6X

OneMap
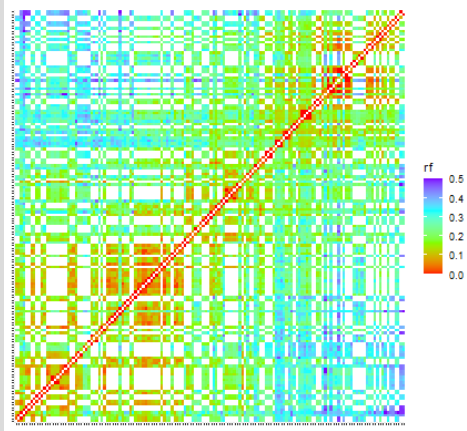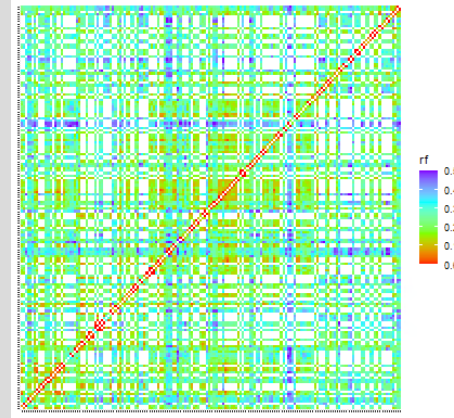
GATK

GATK + polyRAD
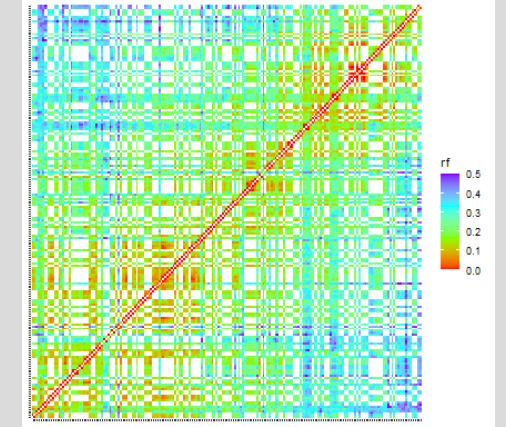
GATK + updog
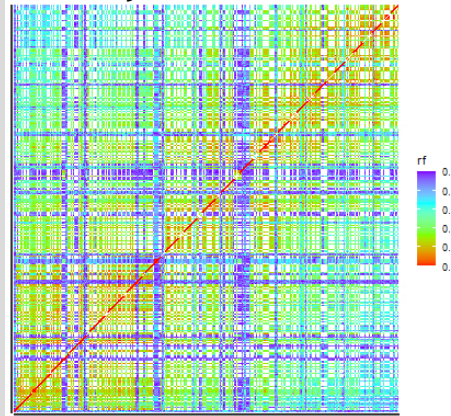
freebayes
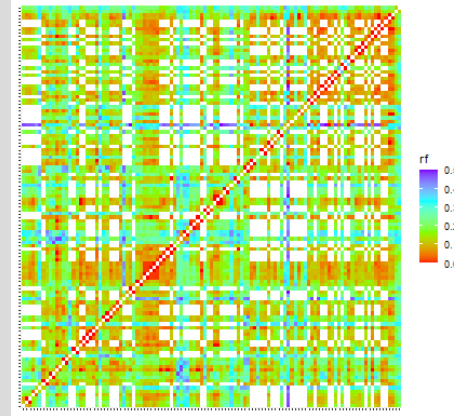
freebayes + polyRAD
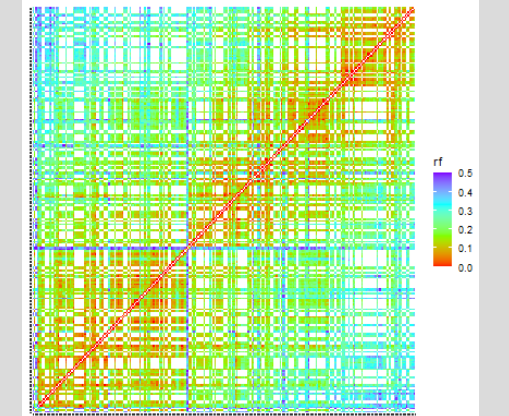
freebayes + updog

Tetraploid rose

- Chromosome 2

- Sequencing depth ~ 50X
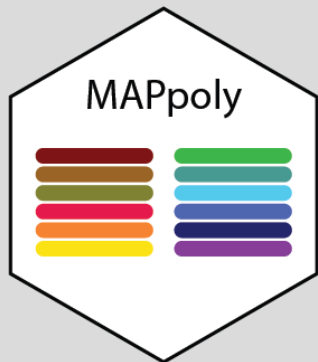
MAPpoly
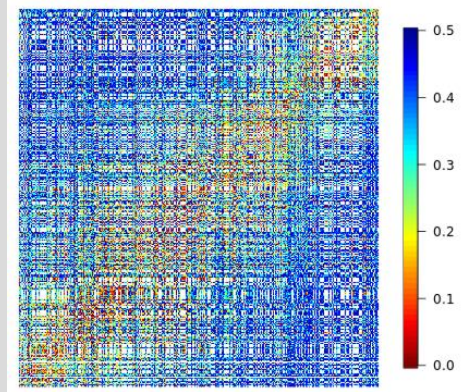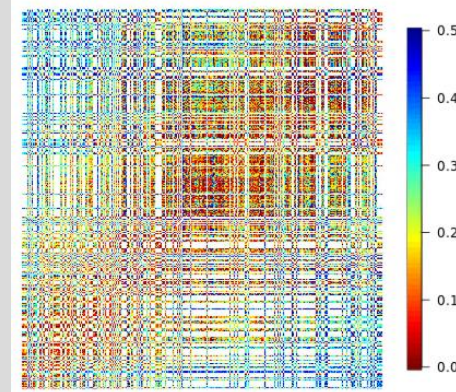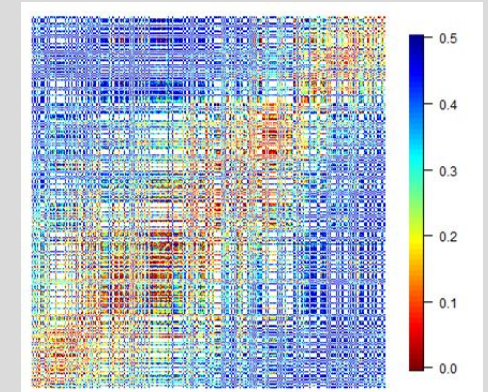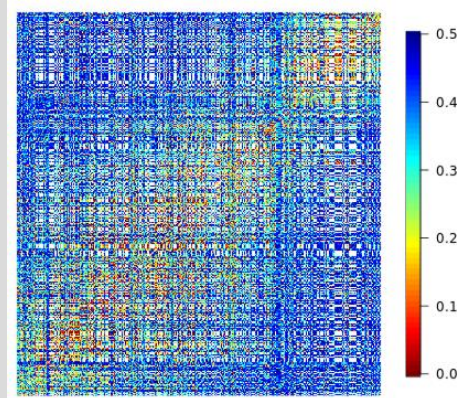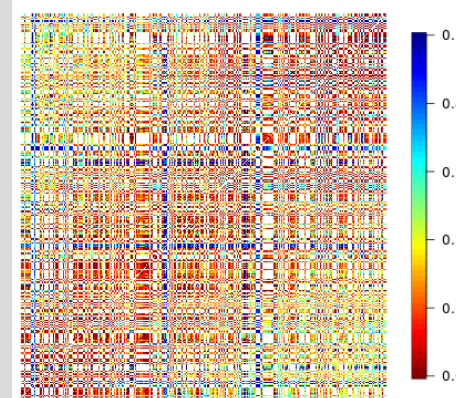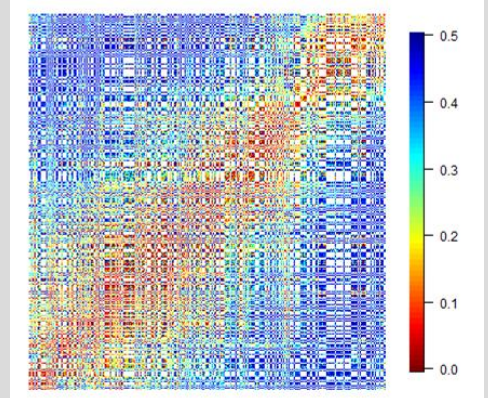
Map size

- Chromosome 2

- Sequencing depth ~ 50X

GATK

TEXAS A&M UNIVERSITY
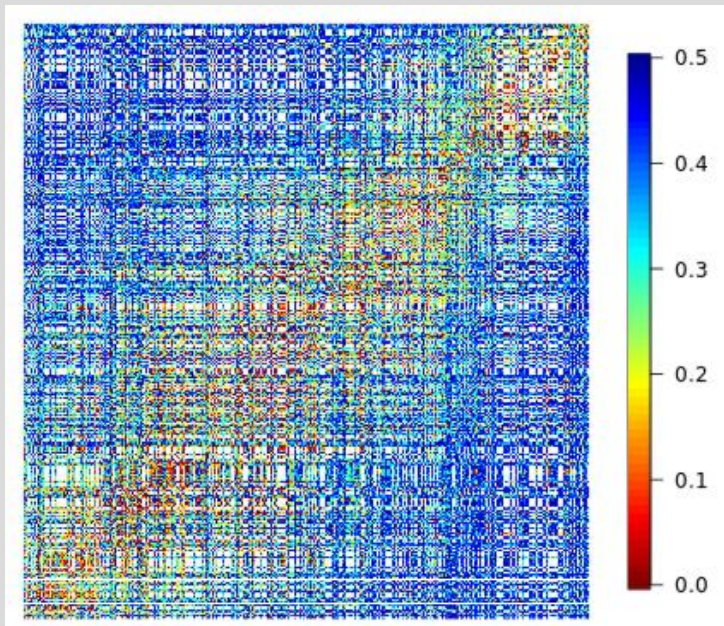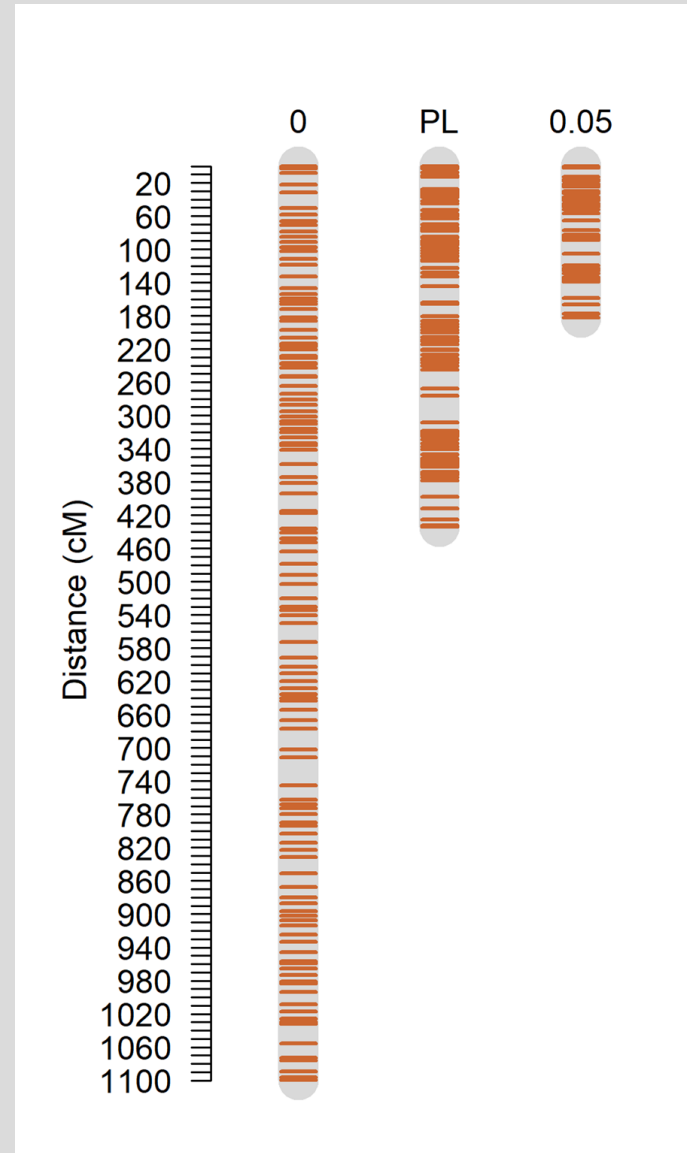
Overcoming Technical Difficulties

- Large files
  - High Performance Computing (HPC)
  - Management systems (SLURM, SGE)
  - Cloud (Google, Amazon)

- Many software
- Many programming languages
- Different Operational Systems
- Updates
  - Containers
    - Docker
    - Singularity (usually available in HPC)
    - BioContainers

TOOLS FOR POLYPLOIDS

TEXAS A&M UNIVERSITY
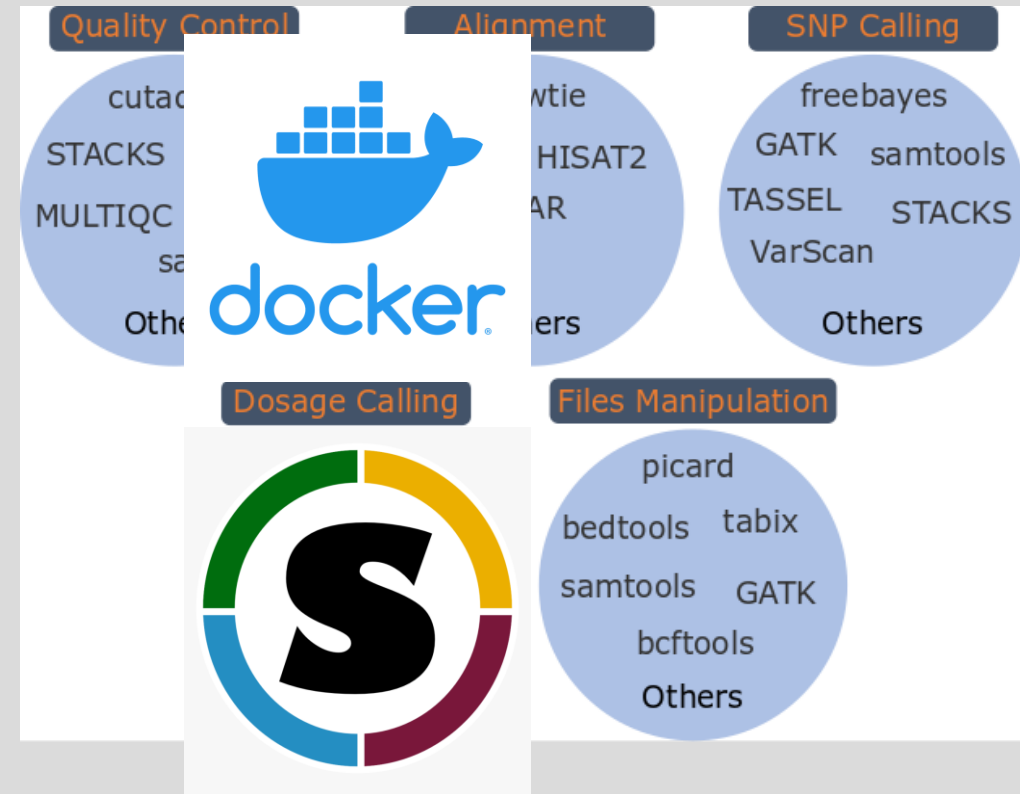
Overcoming Technical Difficulties

- Many steps
- Many file formats
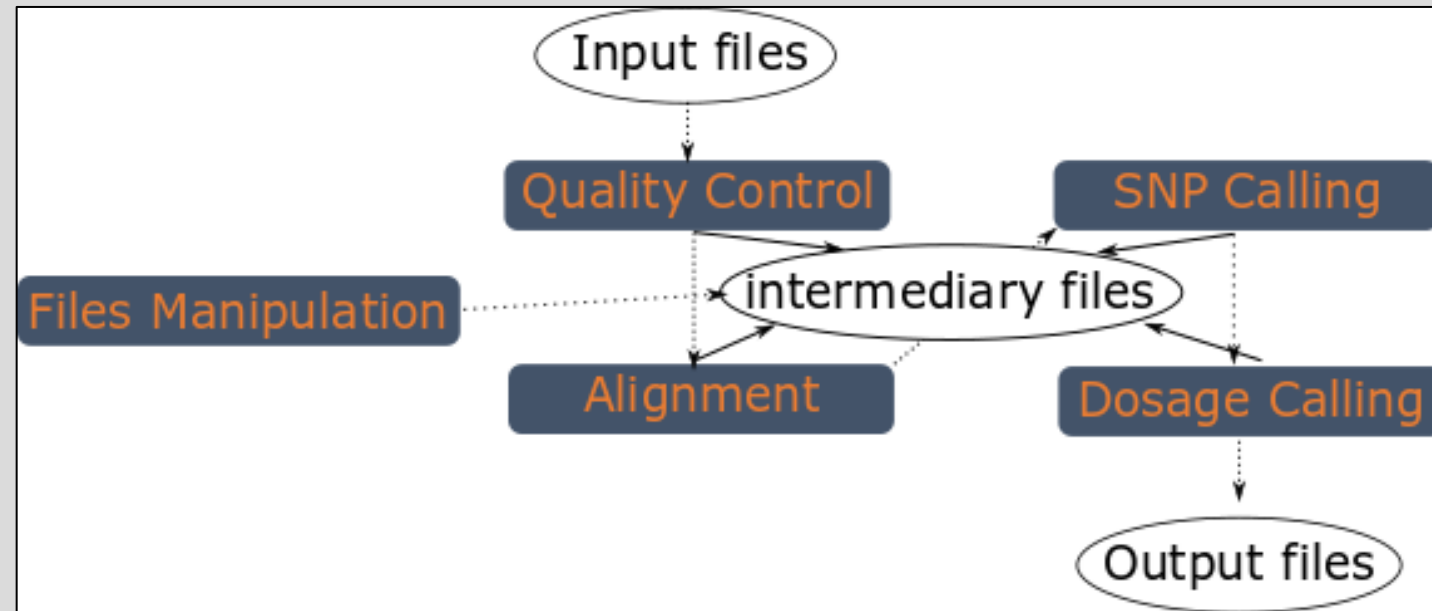
Overcoming Technical Difficulties

- Many steps
- Many file formats
  - Workflows systems
    - Galaxy; Nextflow; Snakemake; CWL; **WDL**
  - Workflows repositories
    - Dockerstore; WorkflowHub
  - Run workflows on Cloud
    - Galaxy; DNAnexus; Terra; AnVIL; SevenBridges

TOOLS FOR POLYPLOIDS

TEXAS A&M UNIVERSITY

Reads2Map

Reads2Map

- Cloud environments
  - terra.bio
- HPC
  - Cromwell
  - MiniWDL
  - dxWDL

```json
{
  "SNPCalling.max_cores": 2,
  "SNPCalling.ploidy": 4,
  "SNPCalling.rm_dupli": false,
  "SNPCalling.replaceAD": false,
  "SNPCalling.run_gatk": true,
  "SNPCalling.run_freebayes": true,
  "SNPCalling.hardfilters": true,
  "SNPCalling.n_chrom": 1,
  "SNPCalling.chunk_size": 2,
  "SNPCalling.samples_info": "tests/data/polyploid/fastq/samples_info.txt",
  "SNPCalling.gatk_mchap": false,
  "SNPCalling.references": {
    "ref_fasta": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.fasta",
    "ref_dict": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.dict",
    "ref_ann": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.fasta.ann",
    "ref_sa": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.fasta.sa",
    "ref_amb": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.fasta.amb",
    "ref_pac": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.fasta.pac",
    "ref_bwt": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.fasta.bwt",
    "ref_fasta_index": "tests/data/polyploid/RchinensisV1.0/Chr04_sub.fasta.fai"
  }
}
```

```
$ java -jar /path/to/cromwell.jar run -i EmpiricalSNPCalling/inputs.json EmpiricalSNPCalling.wdl
```

TOOLS FOR
POLYPLOIDS

task

main ▾   Reads2Map / tasks /

Cristianetaniguti joint wf

..

📄 BWA.wdl

📄 JointReports.wdl

📄 bcftools.wdl

📄 chunk_lists.wdl

📄 cutadapt.wdl

📄 freebayes.wdl

📄 gatk.wdl

📄 gusmap.wdl

📄 mchap.wdl

📄 pedigree_simulator.wdl

📄 gusmap.wdl

📄 mchap.wdl

📄 pedigree_simulator.wdl

📄 pedigree_simulator_utils.wdl

📄 pirs.wdl

📄 radinitio.wdl

📄 simuscop.wdl

📄 stacks.wdl

📄 utils.wdl

📄 utilsR.wdl

📄 vcf2diploid.wdl

TEXAS A&M UNIVERSITY

task
example: freebayes.wdl


```
task RunFreebayes {

    input { ⋯
    }

    Int disk_size = ceil(size(reference, "GiB") + size(bam, "GiB") +  50)
    Int memory_size = ceil(size(bam, "MiB") * 25 * max_cores + 10000)

    command <<< ⋯
    >>>

    runtime { ⋯
    }

    meta { ⋯
    }

    output { ⋯
    }
}
```

```
input {
    File reference
    File reference_idx
    File bam
    File bai
    Int max_cores
    Int ploidy
}

Int disk_size = ceil(size(reference, "GiB") + size(bam, "GiB") +  50)
Int memory_size = ceil(size(bam, "MiB") * 25 * max_cores + 10000)
```

inputs.json

```
{
    "reference": "tests/data/PtrichocarpaV3.0/Chr10.2M.fa",
    "reference_idx": "tests/data/PtrichocarpaV3.0/Chr10.2M.fa.fai",
    "bam": "tests/data/Ptremula_PRJNA395596_subset/merged.bam",
    "bai": "tests/data/Ptremula_PRJNA395596_subset/merged.bam.bai",
    "max_cores": 2,
    "ploidy": 4
}
```

task
example: freebayes.wdl


```
task RunFreebayes {

    input { …
    }

    Int disk_size = ceil(size(reference, "GiB") + size(bam, "G:
    Int memory_size = ceil(size(bam, "MiB") * 25 * max_cores +

    command <<< …
    >>>

    runtime { …
    }

    meta { …
    }

    output { …
    }
}
```

```
command <<<

ln -s ~{bam} .
ln -s ~{bai} .

freebayes-parallel <(fasta_generate_regions.py ~{reference_idx} 100000) ~{max_cores} \
--genotype-qualities --ploidy ~{ploidy} -f ~{reference} *bam > "freebayes.vcf"

>>>
```

TEXAS A&M UNIVERSITY.

task
example: freebayes.wdl


```
task RunFreebayes {

  input {...
  }

  Int disk_size = ceil(size(reference, "GiB") + size(bam, "GiB") +  50)
  Int memory_size = ceil(size(bam, "MiB") * 25 * max_cores + 10000)

  command <<< ...
  >>>

  runtime {...
  }

  meta {...
  }

  output {...
  }
}
```

```
runtime {
  docker: "cristaniguti/freebayes:0.0.1"
  cpu: max_cores
  # Cloud
  memory:"~{memory_size} MiB"
  disks:"local-disk " + disk_size + " HDD"
  # Slurm
  job_name: "RunFreebayes"
  mem:"~{memory_size}M"
  time:"48:00:00"
}
```

task
example: freebayes.wdl


```
task RunFreebayes {

    input { ⋯
    }

    Int disk_size = ceil(size(reference, "GiB") + size(bam, "GiB") + 50)
    Int memory_size = ceil(size(bam, "MiB") * 25 * max_cores + 10000)

    command <<< ⋯
    >>>

    runtime { ⋯
    }

    meta { ⋯
    }

    output { ⋯
    }
}
```

```
meta {
    author: "Cristiane Taniguti"
    email: "chtaniguti@tamu.edu"
    description: "Split genomic regions and runs [freebayes](https://github.com/freebayes/freebayes) parallelized."
}
```

TOOLS FOR POLYPLOIDS

TEXAS A&M UNIVERSITY

task
example: freebayes.wdl

```
task RunFreebayes {

    input { ⋯
    }

    Int disk_size = ceil(size(reference, "GiB") + size(bam, "GiB") +  50)
    Int memory_size = ceil(size(bam, "MiB") * 25 * max_cores + 10000)

    command <<< ⋯
    >>>

    runtime { ⋯
    }

    meta { ⋯
    }

    output { ⋯
    }
}
```

```
output {
    File vcf = "freebayes.vcf"
}
}
```

inputs.json

```
{
    "reference": "tests/data/PtrichocarpaV3.0/Chr10.2M.fa",
    "reference_idx": "tests/data/PtrichocarpaV3.0/Chr10.2M.fa.fai",
    "bam": "tests/data/Ptremula_PRJNA395596_subset/merged.bam",
    "bai": "tests/data/Ptremula_PRJNA395596_subset/merged.bam.bai",
    "max_cores": 2,
    "ploidy": 4
}
```

```
miniwdl run --task RunFreebayes -i tests/tasks/freebayes/inputs.json tasks/freebayes.wdl
```

subworkflow

example


```
workflow FreebayesGenotyping {
  input {...
  }

  call chunk_lists.CreateChunksBamByChr {...
  }

  scatter (chunk in zip(CreateChunksBamByChr.bams_chunks, CreateChunksBamByChr.bais_chunks)) {

    call freebayes.RunFreebayes {...
    }
  }

  call utils.mergeVCFs {...
  }

  call norm_filt.Normalization {...
  }

  Map[String, Array[File]] map_bams = {"bam": CreateChunksBamByChr.bams_chunks, "bai": CreateChunk

  if(replaceAD){...
  }

  Array[File] freebayes_vcfs = select_all([Normalization.vcf_norm, ReplaceAD.bam_vcf])
  Array[String] freebayes_software = select_all([Normalization.software, ReplaceAD.software])
  Array[String] freebayes_counts_source = select_all([Normalization.source, ReplaceAD.source])

  output {...
  }
}
```

```
$ java -jar /path/to/cromwell.jar run -i freebayes_genotyping/inputs.json freebayes_genotyping.wdl
```

pipeline
example

```
workflow SNPCalling {

  input { ...
  }

  call fam.CreateAlignmentFromFamilies { ...
  }

  if(run_gatk){
    call gatk.GatkGenotyping { ...
    }
  }

  if(run_freebayes){
    call freebayes.FreebayesGenotyping { ...
    }
  }

  Array[Array[File]] vcfs_sele = select_all([GatkGenotyping.vcfs, FreebayesGenotyping.vcfs])
  Array[Array[String]] software_sele = select_all([GatkGenotyping.vcfs_software, FreebayesGenotyping.vcfs_software
  Array[Array[String]] source_sele = select_all([GatkGenotyping.vcfs_counts_source, FreebayesGenotyping.vcfs_count

  output { ...
  }
}
```

```
$ java -jar /path/to/cromwell.jar run -i EmpiricalSNPCalling/inputs.json EmpiricalSNPCalling.wdl
```

Reads2Map
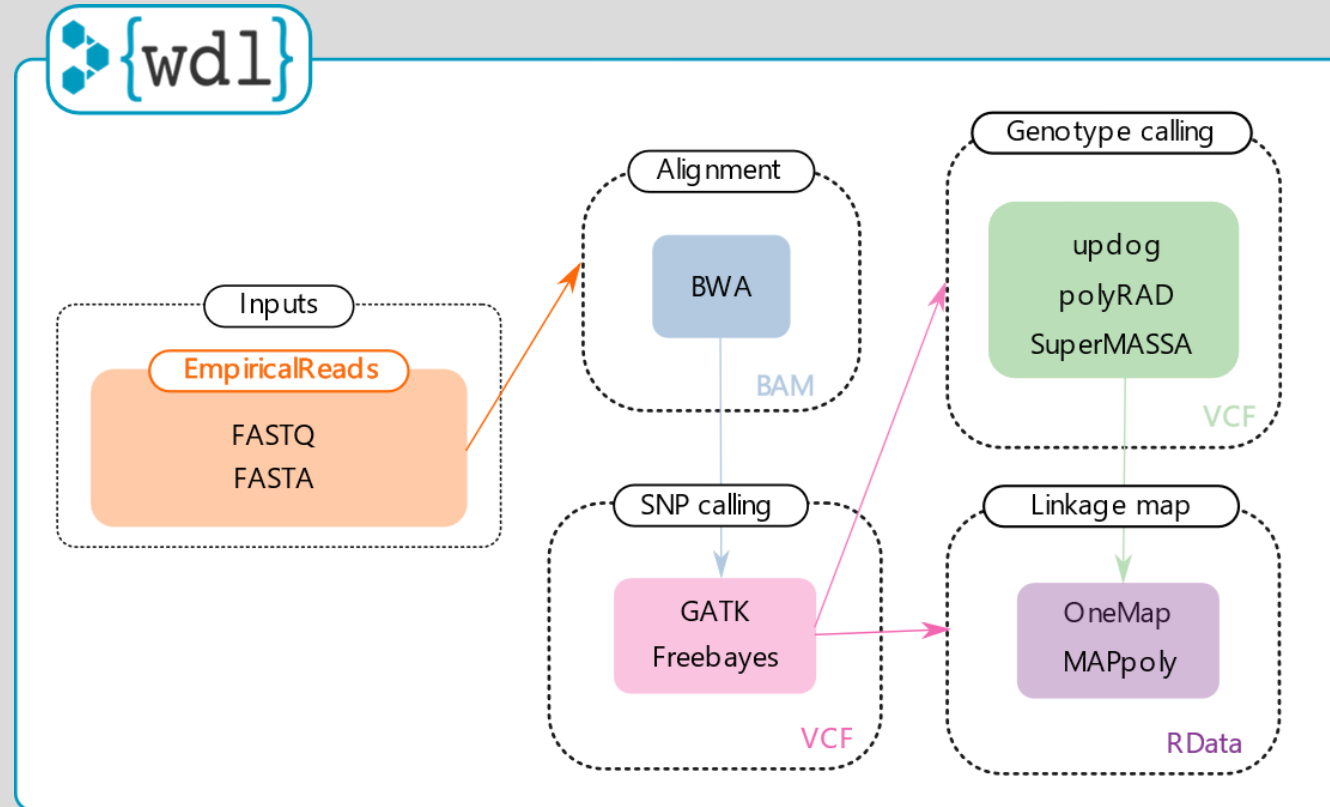
Available in Github, Dockerstore and WorkflowHub

Summary

- Linkage map quality is very sensitive to upstream choices
- There is no single suggestion for every data set
- Reads2Map uses software's default/tutorials parameters
- Developers, sorry if I used any of the parameters wrongly (let me know)
- Easy to test updates or modification of parameters
- Reads2Map is an organized structure for reproducibility and scalable production

TEXAS A&M UNIVERSITY®

Thank you

Oscar Riera-Lizarazu
David Byrne
Jeekin Lau
Tessa Hochhaus

Augusto Garcia
Rodrigo Amadeu
Getulio Caixeta

Marcelo Mollinari
Gabriel Gesteira

Lucas Taniguti

Guilherme Pereira

Thiago Oliveira

Project Members

Other Collaborators