

User Manual for Nextflow Pipeline-DIVIA

Written by Wenchao Zhang

Center for Applied Bioinformatics, St. Jude Children Research Hospital

02/23/2024

Introduction

Here, we introduce our developed nextflow pipeline DIVIA.nf. It takes the RNAseq fastq files as input, and can sequentially deliver each functional module's output, including the STAR-Mapping results, Fusion-Catcher, STAR-Fusion and Arriba-Fusion detection tables, iAdmix's estimating results as the admixture proportions for an individual's ancestry, Pindel detection result as a VCF file, RNAIndel, RSEM based RNAseq gene/transcript measurable matrix and the downstream ML(machine-learning) based Gene expression classifier's result, RNAseq based GATK variant calling results and the downstream variant annotation result and RNAseq CNV analysis result.

The pipeline mainly includes a nextflow script file- DIVIA.nf, nextflow configurable file -nextflow.config, and external R script package including ML-Classifer and RNASeqCNV. We also consider to run the pipeline in a physical or a cloud HPC cluster, singularity container files DIVIA.def and DIVIA.sif are also provided.

To run the pipeline-DIVIA.nf, user need to 1) prepare a RNAseq fastq file samplelist, 2.) modify the nextflow.config, and 3.) invoke the pipeline either at -profile "cluster" or -profile "singularity".

Design Architecture, Workflow, and Implementation

Our previous bench study work shows that STAR-Mapping is time consuming, and the latest STAR at version 2.7.9 can output two types of BAM files as "xx.STAR.Aligned.sortedByCoord.out.bam", "xx.STAR.Aligned.toTranscriptome.out.bam" and a junction file as xx.STAR.Chimeric.out.junction, which can be used in the downstream modules for Arriba fusion detection and RNAseq GATK variant calling, RSEM count matrix calculation, and STAR-fusion detection respectively. Therefore, one STAR-Mapping result, in principle, can be re-used and adapted to the downstream modules.

Considering the flexibility, we set a bunch of "Y/N" switch variables, by which the whole pipeline can be customized into one or several functions specific pipeline, such as STAR-Mapping only, STAR-Mapping+ Arriba /STAR-Fusion fusion detection, STAR-Mapping + RSEM, STAR-Mapping+ RSEM & ML_Rank, STAR_Mapping+ RSEM & ML_Rank +GATK+RNAseq_CNV, STAR-Mapping +Pindel+iAdmix + RSEM

Nextflow pipeline usually includes the nextflow script .nf file and a nextflow.config. To run the pipeline, you need a physical/cloud HPC cluster to parallelly speed up. The memory quota, the cpu number for each functional application module can be specifically configured in the file nextflow.config; The most important thing is that the executable application program at the specific version can be simply deployed by method 'module load'. Containerization of workloads has become popular, by which, all the functional applications and the dependency packages can be packed into the container and the results can be reproduced. Of all the container options, Docker is not suitable for HPC applications due to its requirement for super root privilege for security reasons, although it's very popular in micro-service. Here, the singularity is adopted for its advantages in HPC, the singularity image file DIVIA.sif is built based on a singularity container definition file DIVIA.def. In short, the nextflow pipeline provides two run profiles as '-profile cluster' and '-profile singularity', calling the executable application by "module load" or loading from singularity image DIVIA.sif respectively. Both of the two running profiles have been tested in HPC LSF cluster.

Based on these, the architecture is designed and illustrated in Fig.1, which can be summarized as the following three main features.

- 1.) Maximum reusability of the upstream' output channel
- 2.) Maximum flexible customization by a series of 'Y/N' switch variables.
- 3.) Two running profiles which can allow user to run the pipeline at a HPC cluster and/or deploy the singularity container.

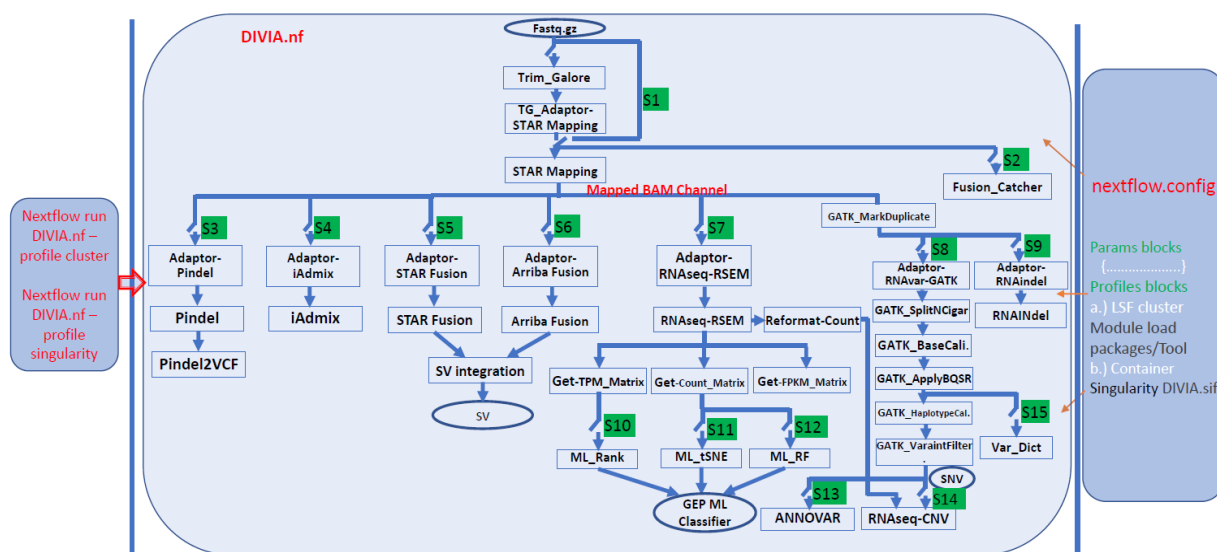


Fig.1 The architecture of nextflow pipeline-DIVIA.nf and the relationship among the main script-DIVIA.nf, the configuration file nextflow.config and the run cmds in terminal.

In our further application test, we found that the GATK Haplotype_Caller can be terribly slow when running some challenging sample level at whole genome level. However, the deployment of multiple computing nodes for running chromosome level Haplotype_Caller can achieve 20 more times speed up. Given hundred samples, the nextflow running Haplotypecaller in whole genome level just require hundred of cpu nodes, but it will demand several thousands cpu nodes running in mode of per-chromosome level haplotype_caller, which occupy too much HPC cluster resource and may affect other HPC users. Therefore, we design it as a flexible option (Fig.2).

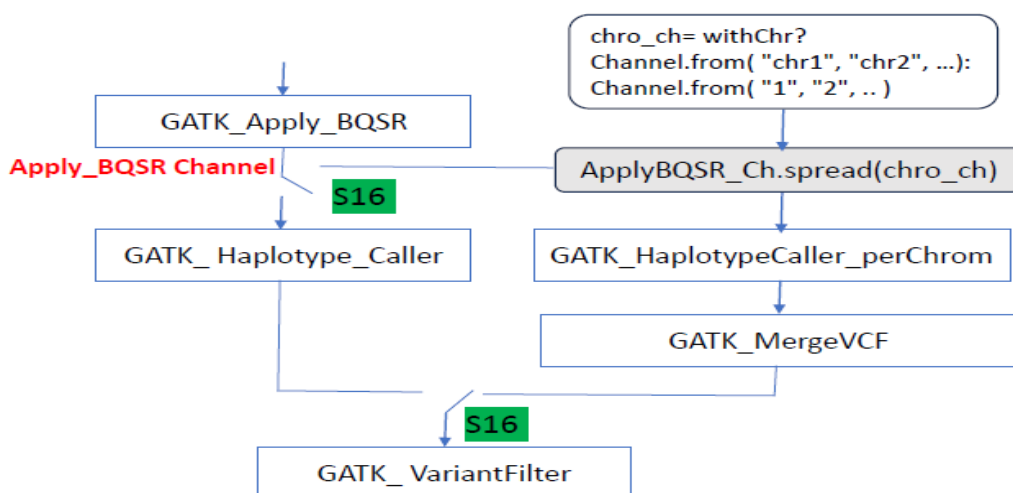


Fig.2 Workflow to run sample level HaplotypeCaller or sample' Chromosome level HaplotypeCaller

In short, the whole workflow of pipeline DIVIA.nf is illustrated in fig.3.

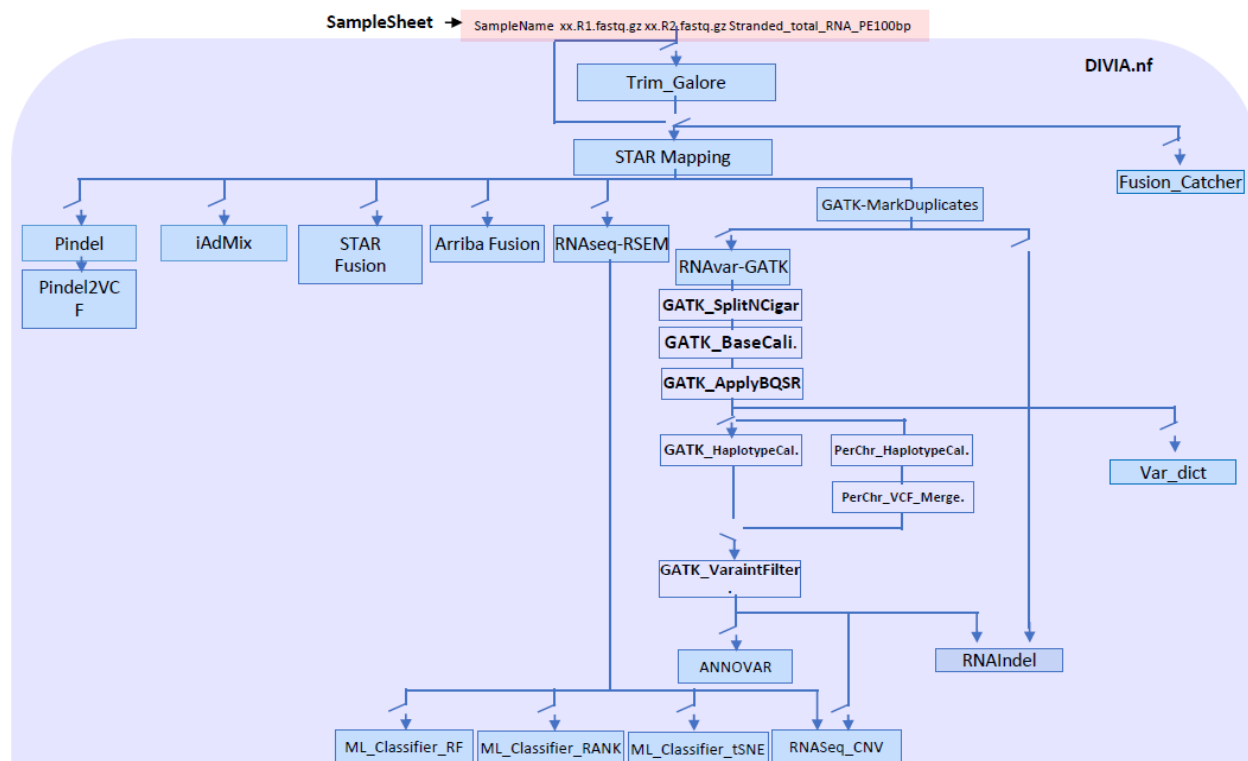


Fig.3 Workflow of Pipeline DIVIA.nf

Reusability implemented by forking output channel

Fig.1 indicate several places that are reused and the upstream outputs are forked into several channels for their downstream process modules, including the sample level' STAR-Mapping result, sample level' RNAseq-RSEM result, count-matrix across samples, and sample level filtered GATK called variants. For example, STAR-Mapping result are forked into four copies and then sent to the downstream arriba-fusion, STAR-fusion, RSEM, and GATK respectively (see the red circle part in Fig.4).

```
/*Fork the STAR Mapping stream into several downstream channels*/
STAR Mapping_Ch.into{ STAR Mapping_Ch1; STAR Mapping_Ch2; STAR Mapping_Ch3; STAR Mapping_Ch4; STAR Mapping_Ch5; STAR Mapping_Ch6; STAR Mapping_Ch7 }

/******
* This Process is be responsible for adjusting/editting STAR Mapping's outputs to adapt to the downstream STAR Fusion
*****
Adaptor_STAR_Fusion_In_Ch = (params.Select_STAR_Fusion == "N" ) ? Channel.empty() : STAR Mapping_Ch1
process STARMapping_Adaptor_STAR_Fusion {
// publishDir "${params.outdir}/${params.project}/STARMapping_Adaptor_STAR_Fusion/${SampleName}", mode: 'copy', overwrite: true
```

Fig.4 Illustration of reusability by forking STAR-Mapping output channel into 7 channels (check with DIVIA.nf)

The downstream input maybe requires only a portion of the upstream output, additionally, the format may also need to be reformatted to adapt the downstream module. For example, the 21-columns Chimeric junction file from STAR-2.7.9a need to be converted into 16- column Chimeric junction file in STAR-Fusion detection module. Therefore, an adaptor module named as STARMapping_Adaptor_STAR_Fusion must be developed, by which the original junction file will be converted into the suitable format, and two STAR-Mapped BAM files will be ignored during the STAR-

fusion detection. Fig.5 illustrate this adaption process, which essentially is implemented by a shell pipe of 'grep' and 'awk'.

Similarly, process STARMapping_Adaptor_Arriba_Fusion, STARMapping_Adaptor_RNAseq_RSEM, and STARMapping_Adaptor_RNAvar_GATK are also implemented to select only the suitable part from the STAR-Mapping's output and adapt to the downstream arriba fusion detection, RNS-seq-RSEM, and RNA-seq-GATK process respectively.

```
Adaptor_STAR_Fusion_In_Ch = (params.Select_STAR_Fusion == "N" ) ? Channel.empty() : STAR_Mapping_Ch1
process STARMapping_Adaptor_STAR_Fusion {
  publishDir "${params.outdir}/${params.project}/STARMapping_Adaptor_STAR_Fusion/${SampleName}", mode:
  'copy', overwrite: true

  input:
    set SampleName, file(Align_sortedByCoord_Bam), file(Align_toTranscriptome_Bam),
    file(ReadsPerGene_out_tab), file(SJ_Out_Tab), file(Chimeric_out_junction) from Adaptor_STAR_Fusion_In_Ch

  output:
    set SampleName, file("${SampleName}.Chimeric.out.junction") into Adaptor_STAR_Fusion_Ch
    //ln -s ${Chimeric_out_junction} ${SampleName}.Chimeric.out.junction

  script:
    """
    echo "need to convert 21-column Chimeric junction file to 16-column Chimeric junction file"
    cat ${Chimeric_out_junction}| grep "^#" > Last_twolines_comment.txt
    cat ${Chimeric_out_junction}| grep -v "^#" | awk 'BEGIN {OFS="\t"}; NR>1 { print $1,$2,$3,$4,$5,$
    $6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$21}' > ${SampleName}.Chimeric.out.junction
    cat Last_twolines_comment.txt >> ${SampleName}.Chimeric.out.junction
    """
}
```

Fig.5 Illustration of formatting and adapting the STAR-Mapping's output to the downstream STAR-Fusion (check with DIVIA.nf).

Flexibility implemented by 'Y/N' binary switch variables

Fig.1 illustrate 10 more switch variables, by which the whole pipeline can be customized into only one or several functions specific pipeline. This strategy greatly enhance the flexibility in debugging and function testing. All the switch variables can be configured in nextflow.config (see fig.6). If one switch variable is configured with "N", the corresponding data channel will be assigned with Channel.empty() and the corresponding process module will not be invoked.

```
Select_Trim_Galore = "Y"
// Whether select Trim_Galore for trimming reads and FastQC before STAR Mapping. Y(Default) | N

Select_STAR_Fusion = "Y"
// Whether select STAR_Fusion for Fusion Detection. Y(Default) | N

Select_Arriba_Fusion = "Y"
// Whether select Arriba for Fusion Detection. Y(Default) | N

Select_GATK = "Y"
// Whether select GATK for rna variant calling. Y(Default) | N

Select_Variant_ANNOVAR = "Y"
// Whether select ANNOVAR for Variant Annotation. Y(Default) | N

Select_RSEM = "Y"
// Whether select RSEM for rnaseq quantification. Y(Default) | N

Select_ML_Classifier_RANK = "Y"
// Whether select RSEM' Gene Expression data for downstream ML_Classifier RANK . Y(Default) | N

Select_ML_Classifier_tsNE = "Y"
// Whether select RSEM' Gene Expression data for downstream ML_Classifier tsNE. Y(Default) | N

Select_ML_Classifier_RF = "Y"
// Whether select RSEM' Gene Expression data for downstream ML_Classifier RandomForest. Y(Default) | N

Select_RNASeqCNV = "Y"
// Whether select RNASeqCNV for RNAseq CNV calling. Y(Default) | N
```

Fig.6 The Y/N switch variables are defined in nextflow.config, and the different confirmation can ensure the pipeline's flexibility.

For example, the line “Adaptor_STAR_Fusion_In_Ch = (params.Select_STAR_Fusion == "N") ? Channel.empty() : STAR_Mapping_Ch1” in DIVIA.nf mean that if you configure the switch variable Select_STAR_Fusion with “N”, then the input channel for process Adaptor_STAR_Fusion will be assigned as empty, and the process Adaptor_STAR_Fusion will not be invoked, and no output will be delivered, finally, the downstream process STAR_Fusion will also not be invoked.

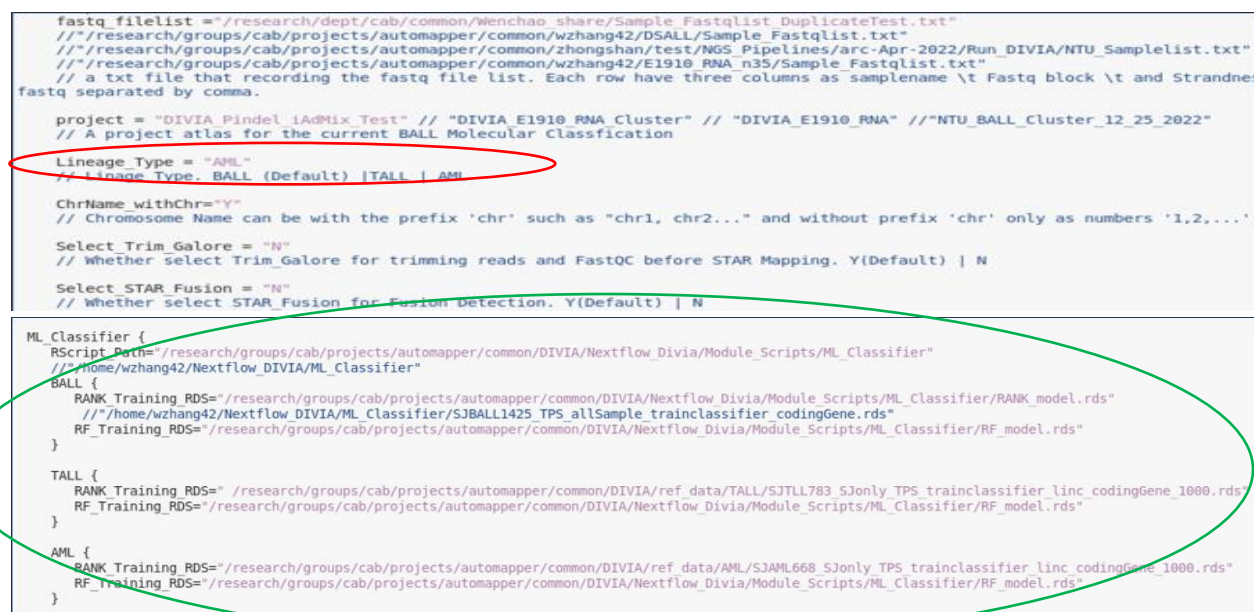
Two run profiles as HPC module loading and deploying the singularity image container.

Nextflow script file DIVIA.nf essentially is a HPC cluster wrapper for the parallelly and continuously run a series of bioinformatic application programs such as STAR, STAR-Fusion, arriba-fusion, RSEM, GATK,.... In HPC cluster, the application programs can be called by “module load xx”. However, as mentioned before, this strategy highly depends on the available programs that have been installed in HPC cluster, which can not ensure the reproductive result in two HPC cluster environment. Singularity container is a good option, which can solve this issue by pre-installing the bioinformatic programs in a packed container. Therefore, we developed a singularity definition file DIVIA.def file and built the corresponding singularity image file DIVIA.sif.

In Nextflow.config, it provides two profiles as “cluster” and “singularity”, which can be configured separately. Here, we want to remind that machine learning classifier scripts such as ML_Classifier_RF.R, ML_Classifier_RANK.R, ML_Classifier_tSNE.R, and RNASeqCNV.R need more R functional packages and dependency packages, which should completely be installed in the container. We have installed the required packages required in the current R scripts. In future, if the R script are modified and require other R packages, the container should be modified and built accordingly.

User flexible configuration for handling pan-ALL classification

in our understanding, pan-ALL either as BALL, TALL or AML can share most of processing module. In DIVIA pipeline, only the ML training module for ML_Classifier_RANK and ML_Classifier_RF are different. Fig.7 illustrates the lineage configuration and .rds module files for BALL, TALL and AML.



```

fastq_filelist = "/research/dept/cab/common/Wenchao_share/Sample_FastqList_DuplicateTest.txt"
// "/research/groups/cab/projects/automapper/common/wzhang42/DSALL/Sample_FastqList.txt"
// "/research/groups/cab/projects/automapper/common/zhongshan/test/NGS_Pipelines/arc-Apr-2022/Run_DIVIA/NTU_SampleList.txt"
// "/research/groups/cab/projects/automapper/common/wzhang42/E1910_RNA_n35/Sample_FastqList.txt"
// a txt file that recording the fastq file list. Each row have three columns as sampleName \t Fastq block \t and Strandness
fastq separated by comma.

project = "DIVIA Pindel iA@Mix_Test" // "DIVIA E1910_RNA_Cluster" // "DIVIA E1910_RNA" // "NTU BALL_Cluster_12_25_2022"
// A project atlas for the current BALL Molecular Classification

Lineage_Type = "AML"
// Lineage Type: BALL (Default) | TALL | AML

ChrName withChr="Y"
// Chromosome Name can be with the prefix 'chr' such as "chr1, chr2..." and without prefix 'chr' only as numbers '1,2,...'

Select_Trim_Galore = "N"
// Whether select Trim_Galore for trimming reads and FastQC before STAR Mapping. Y(Default) | N

Select_STAR_Fusion = "N"
// Whether select STAR_Fusion for Fusion Detection. Y(Default) | N

ML_Classifier {
  RScript_Path = "/research/groups/cab/projects/automapper/common/DIVIA/Nextflow_Divia/Module_Scripts/ML_Classifier"
  // "/home/wzhang42/Nextflow_DIVIA/ML_Classifier"
  BALL {
    RANK_Training_RDS = "/research/groups/cab/projects/automapper/common/DIVIA/Nextflow_Divia/Module_Scripts/ML_Classifier/RANK_model.rds"
    // "/home/wzhang42/Nextflow_DIVIA/ML_Classifier/SJBALL1425_TPS_allSample_trainclassifier_codingGene.rds"
    RF_Training_RDS = "/research/groups/cab/projects/automapper/common/DIVIA/Nextflow_Divia/Module_Scripts/ML_Classifier/RF_model.rds"
  }

  TALL {
    RANK_Training_RDS = "/research/groups/cab/projects/automapper/common/DIVIA/ref_data/TALL/SJTLL783_SJonly_TPS_trainclassifier_linc_codingGene_1000.rds"
    RF_Training_RDS = "/research/groups/cab/projects/automapper/common/DIVIA/Nextflow_Divia/Module_Scripts/ML_Classifier/RF_model.rds"
  }

  AML {
    RANK_Training_RDS = "/research/groups/cab/projects/automapper/common/DIVIA/ref_data/AML/SJAML668_SJonly_TPS_trainclassifier_linc_codingGene_1000.rds"
    RF_Training_RDS = "/research/groups/cab/projects/automapper/common/DIVIA/Nextflow_Divia/Module_Scripts/ML_Classifier/RF_model.rds"
  }
}

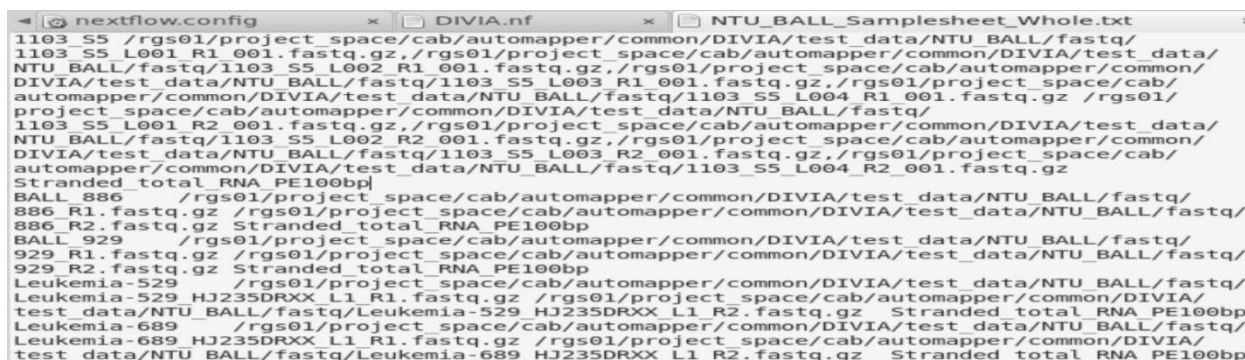
```

Fig.7 Nextflow configuration for user to flexible run either BALL , TALL or AML .

User Application

All procedure to run the DIVIA.nf pipeline can be simply summarized as the followings:

- 1.) prepare a fastq samplesheet as the same format of AutoMapper. That is, each row representing one sample contain three columns as the sample_id, and read fastq file path, strandness delimited as TAB. In column 2, the fastq files for pair-end reads are delimited by space and the fastq files for multiple LANEs are delimited by comma. The strandness should be configured per sample and only can be as one of the four cases: Stranded_total_RNA_PE100bp, Stranded_total_RNA_PE75bp, Unstranded_mRNA_PE100bp and Unstranded_mRNA_PE75bp. Fig.8 show one example sample sheet.



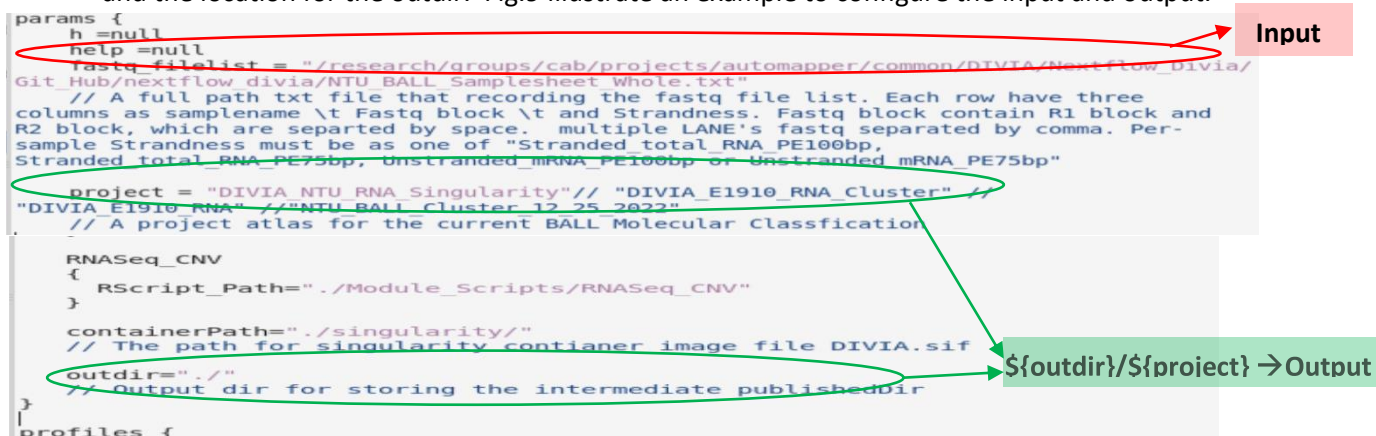
```

1103 S5 /rgs01/project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
1103 S5 L001 R1 001.fastq.gz,/rgs01/project_space/cab/automapper/common/DIVIA/test_data/
NTU_BALL/fastq/1103 S5 L002 R1 001.fastq.gz,/rgs01/project_space/cab/automapper/common/
DIVIA/test_data/NTU_BALL/fastq/1103 S5 L003 R1 001.fastq.gz,/rgs01/project_space/cab/
automapper/common/DIVIA/test_data/NTU_BALL/fastq/1103 S5 L004 R1 001.fastq.gz /rgs01/
project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
1103 S5 L001 R2 001.fastq.gz,/rgs01/project_space/cab/automapper/common/DIVIA/test_data/
NTU_BALL/fastq/1103 S5 L002 R2 001.fastq.gz,/rgs01/project_space/cab/automapper/common/
DIVIA/test_data/NTU_BALL/fastq/1103 S5 L003 R2 001.fastq.gz,/rgs01/project_space/cab/
automapper/common/DIVIA/test_data/NTU_BALL/fastq/1103 S5 L004 R2 001.fastq.gz
Stranded total RNA PE100bp
BALL 886 /rgs01/project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
886 R1.fastq.gz /rgs01/project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
886 R2.fastq.gz Stranded total RNA PE100bp
BALL 929 /rgs01/project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
929 R1.fastq.gz /rgs01/project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
929 R2.fastq.gz Stranded total RNA PE100bp
Leukemia-529 /rgs01/project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
Leukemia-529 HJ235DRXX L1 R1.fastq.gz /rgs01/project_space/cab/automapper/common/DIVIA/
test_data/NTU_BALL/fastq/Leukemia-529 HJ235DRXX L1 R2.fastq.gz Stranded total RNA PE100bp
Leukemia-689 /rgs01/project_space/cab/automapper/common/DIVIA/test_data/NTU_BALL/fastq/
Leukemia-689 HJ235DRXX L1 R1.fastq.gz /rgs01/project_space/cab/automapper/common/DIVIA/
test_data/NTU_BALL/fastq/Leukemia-689 HJ235DRXX L1 R2.fastq.gz Stranded total RNA PE100bp
  
```

Fig.8 Example of a sample sheet.

- 2.) Acquire the pipeline package from http://phoebe.stjude.org/wzhang42/nextflow_divia.git (BALL) or https://phoebe.stjude.org/wzhang42/nextflow_pan_divia_rhel8 (PAN-ALL either as BALL, TALL, or AML)
- 3.) Modify the nextflow.config
 - a.) Configure the input and output in the param block

The input for pipeline DIVIA.nf is a full path fastq file samplesheet. User need to provide samplelist with the correct format. Additionally, user need to provide an unique Project name and the location for theoutdir. Fig.9 illustrate an example to configure the input and output.



```

params {
  h = null
  help = null
  fastq_filelist = "/research/groups/cab/projects/automapper/common/DIVIA/nextflow_divia/
Git_Hub/nextflow_divia/NTU_BALL_Samplesheet_Whole.txt"
  // A full path txt file that recording the fastq file list. Each row have three
  columns as samplename \t Fastq block \t and Strandness. Fastq block contain R1 block and
  R2 block, which are separated by space. multiple LANE's fastq separated by comma. Per-
  sample Strandness must be as one of "Stranded total RNA PE100bp,
  Stranded total RNA PE75bp, Unstranded_mRNA PE100bp or Unstranded mRNA PE75bp"

  project = "DIVIA NTU RNA Singularity" // "DIVIA_E1910_RNA Cluster" //
  "DIVIA E1910 RNA" // "NTU BALL Cluster 12 25 2022"
  // A project atlas for the current BALL Molecular Classification

  RNaseq_CN
  {
    RScript_Path = "../Module_Scripts/RNaseq_CN"
  }

  containerPath = "../singularity/"
  // The path for singularity container image file DIVIA.sif

  outdir = "/"
  // Output dir for storing the intermediate publishedDir
}
profiles {
  
```

Fig.9 Illustration to configure the parameters for input and output

- b.) Configure the local R_LIBS. There are 4 R scripts : ML_Classifier_RANK.R, ML_Classifier_RF.R, ML_Classifier_tSNE.R, and RNAseqCNV.R, the first three machine-learning classifier script is developed and tested work in R4.2.0, but the RNAseqCNV.R only can work in R4.1.0. They may conflict each other. To make the whole run smooth. Beside that you need to configure the different R version in nextflow.config, you also need to configure your two R_LIBS related parameters which can be dynamically exported and applied to ML_Classifier_xx.R and RNAseqCNV.R respectively. Fig.10 illustrate the configuration of R_LIBS in nextflow.config.

```

RNASeq_CN
{
  RScript_Path="/research/groups/cab/projects/automapper/common/DIVIA/Nextflow_Divia/
  Module_Scripts/RNASeq_CN
}

containerPath="/research/groups/cab/projects/automapper/common/DIVIA/Nextflow_Divia"
//containerPath="/home/wzhang42/Nextflow_DIVIA"
//"/home/wzhang42/Nextflow_DIVIA"
//"/DIVIA_Container"
// The path for container

TMP_DIR="." // "/lustre scratch/user scratch/wzhang42/tmp"
// Temporary Directory that used to store the intermediate files.

R 4 2 LIBS="/home/wzhang42/R/tmp/x86_64-pc-linux-gnu-library/4.2"
R 4 1 LIBS="/home/wzhang42/R/tmp/x86_64-pc-linux-gnu-library/4.1"

outdir="."
//"/research/groups/cab/projects/automapper/common/wzhang42/E1910_RNA" //"/DIVIA_Result"
// Output dir for storing the intermediate publishedDir
}

```

Fig.10 Illustration to configure the parameters for R_LIBS

- c.) Configure the Y/N switch variables
According to your analysis requirement, configure the Y/N switch variable for a customized run (See fig.6).
- d.) If you put the Moudle script files, module reference or singularity container at different places, you also need to configure the correct their path.
- 4.) Module load nextflow/21.10.5
- 5.) Run one of the cmds 1) **nextflow run ./DIVIA.nf -profile cluster** or 2.) **nextflow run ./DIVIA.nf -profile singularity** to initiate the pipeline in profile cluster by “module load” or profile singularity by deploying container image. Notes, the user can combine step 4 and 5 and submit only a batch job, e.g **bsub -P DIVIA -q standard -M 100000 -e err%.err -o out%.out -J DIVIA "module load nextflow/21.10.5 && nextflow run ./DIVIA.nf -profile cluster"**

\$ nextflow run ./DIVIA.nf -profile cluster

```

(base) [wzhang42@splprhpc04 Nextflow_Divia] nextflow run ./DIVIA.nf -profile cluster
Picked up JAVA_OPTIONS: -Djava.io.tmpdir=/research/rgs01/scratch_lsf/java -XX:Parallel
N E X T F L O W ~ version 21.10.6
Launching "/DIVIA.nf" [voluminous_gates] - revision: 7ea1322ea2

Welcome to run Nextflow Pipeline DIVIA.nf
Your configuration are the following:
project      : NTU_BALL_Singularity 12.2.2022
fastq_filelist : NTU_BALL_Samplesheet_Whole.txt
outdir       : ./

Select Trim Galore : Y
Select STAR Fusion : Y
Select Arriba Fusion : Y

Select GATK : Y
Select Variant_ANNVAR : Y

Select RSEM : Y
Select ML_RANK : Y
Select ML_tSNE : Y
Select ML_RF : Y

Select RNASeqCNV : Y
The following detail information is based your configuration:

executor > lsf (191)
[2a/276a2f] process > Zcat MergeFastq (3) [100%] 8 of 8 ✓
[e8/161b73] process > Trim Galore (8) [100%] 8 of 8 ✓
[ae/b6172f] process > Trim Galore Adaptor_STARMapping (8) [100%] 8 of 8 ✓
[d2/4bbcf8] process > STAR Mapping (8) [100%] 8 of 8 ✓
[4b/a419dd] process > STARMapping Adaptor_STAR_Fusion (8) [100%] 8 of 8 ✓
[b9/9b91f8] process > STARMapping Adaptor_Arriba_Fusion (8) [100%] 8 of 8 ✓
[f5/b220fd] process > STARMapping Adaptor_RNAseq_RSEM (8) [100%] 8 of 8 ✓
[14/5f1506] process > STARMapping Adaptor_RNAvar_GATK (8) [100%] 8 of 8 ✓
[00/9a340e] process > STAR_Fusion (8) [100%] 8 of 8 ✓
[f75/c59cfc] process > Arriba Fusion (8) [100%] 8 of 8 ✓
[2a/276a2f] process > Zcat MergeFastq (3) [100%] 8 of 8 ✓
[e8/161b73] process > Trim Galore (8) [100%] 8 of 8 ✓
[ae/b6172f] process > Trim Galore Adaptor_STARMapping (8) [100%] 8 of 8 ✓
[d2/4bbcf8] process > STAR Mapping (8) [100%] 8 of 8 ✓
[4b/a419dd] process > STARMapping Adaptor_STAR_Fusion (8) [100%] 8 of 8 ✓
[b9/9b91f8] process > STARMapping Adaptor_Arriba_Fusion (8) [100%] 8 of 8 ✓
[f5/b220fd] process > STARMapping Adaptor_RNAseq_RSEM (8) [100%] 8 of 8 ✓
[14/5f1506] process > STARMapping Adaptor_RNAvar_GATK (8) [100%] 8 of 8 ✓
[00/9a340e] process > STAR_Fusion (8) [100%] 8 of 8 ✓
[f75/c59cfc] process > Arriba Fusion (8) [100%] 8 of 8 ✓
[10/22430c] process > RNAseqCNV (8) [100%] 8 of 8 ✓
Completed at: 03-Dec-2022 06:09:11
Duration : 18h 27m 16s
CPU hours : 175.5
Succeeded : 191

```


\$ nextflow run ./DIVIA.nf -profile singularity

```
Picked up JAVA_OPTIONS: -Djava.io.tmpdir=/research/rgs01/scratch_lsf/java -XX:ParallelGC
N E X T F L O W ~ version 21.10.6
Launching './DIVIA.nf' [deadly_mercator] - revision: c6107806ea

Welcome to run Nextflow Pipeline DIVIA.nf
Your configuration are the following:
project      : NTU BALL Singularity 12 24 2022
fastq_filelist : NTU BALL Samplesheet_Whole.txt
outdir       : ./

Select Trim_Galore : Y
Select STAR_Fusion : Y
Select Arriba_Fusion : Y

Select GATK : Y
Select Variant_ANNVAR : Y

Select RSEM : Y
Select ML_RANK : Y
Select ML_tSNE : Y
Select ML_RF : Y

Select RNASeqCNV : Y
The following detail information is based your configuration:

executor > lsf (191)
[a2/bec6e9] process > Zcat MergeFastq (3) [100%] 8 of 8 ✓
[db/0297e7] process > Trim Galore (8) [100%] 8 of 8 ✓
[f2/aea9d6] process > Trim Galore Adaptor_STARMapping (8) [100%] 8 of 8 ✓
[98/726b0e] process > STAR Mapping (8) [100%] 8 of 8 ✓
[6b/aae1f7] process > STARMapping Adaptor_STAR_Fusion (8) [100%] 8 of 8 ✓
[f7/79c11e] process > STARMapping Adaptor_Arriba_Fusion (8) [100%] 8 of 8 ✓
[88/a81c8c] process > STARMapping Adaptor_RNAseq_RSEM (8) [100%] 8 of 8 ✓
[5f/b15db4] process > STARMapping Adaptor_RNAvar_GATK (8) [100%] 8 of 8 ✓
[ed/6140b2] process > STAR_Fusion (8) [100%] 8 of 8 ✓
[49/5029e0] process > Arriba_Fusion (8) [100%] 8 of 8 ✓
[5c/8cc380] process > RNAseq_RSEM (8) [100%] 8 of 8 ✓
[2e/67a7ed] process > ExpectedCount_GenesResults_Aadtpr (8) [100%] 8 of 8 ✓
[17/716515] process > ModifiedCount_GenesResults_Aadtpr (8) [100%] 8 of 8 ✓
[e2/a0efaf5] process > TPM_GenesResults_Aadtpr (8) [100%] 8 of 8 ✓
[78/16c30a] process > FPKM_GenesResults_Aadtpr (8) [100%] 8 of 8 ✓
[56/94e3b2] process > RSEM Reformatted_Count (8) [100%] 8 of 8 ✓
[58/51bdc4] process > Get_ExpectedCount_Matrix [100%] 1 of 1 ✓
[7d/457df2] process > Get_ModifiedCount_Matrix [100%] 1 of 1 ✓
[0e/0e6c63] process > Get_TPM_Matrix [100%] 1 of 1 ✓
[fe/9becca] process > Get_FPKM_Matrix [100%] 1 of 1 ✓
[45/d825fa] process > ML_Classifier_RANK [100%] 1 of 1 ✓
[1c/edd631] process > ML_Classifier_tSNE [100%] 1 of 1 ✓
[83/44616d] process > ML_Classifier_RF [100%] 1 of 1 ✓
[a8/5ccf53] process > GATK_SplitNCigarReads (8) [100%] 8 of 8 ✓
[e8/584709] process > GATK_BaseRecalibrator (8) [100%] 8 of 8 ✓
[c7/b453d0] process > GATK_ApplyBQSR (8) [100%] 8 of 8 ✓
[e3/00ac22] process > GATK_HaplotypeCaller (8) [100%] 8 of 8 ✓
[7f/e4b1c5] process > GATK_VariantFiltration (8) [100%] 8 of 8 ✓
[5a/5bfe69] process > Variant_ANNVAR (8) [100%] 8 of 8 ✓
[76/2f2d7a] process > RNASeqCNV (8) [100%] 8 of 8 ✓
Completed at: 24-Dec-2022 22:46:08
Duration : 1d 3m 7s
CPU hours : 152.8
Succeeded : 191
```

Fig.11 nextflow interface for progress status updating

6.) Check and navigate the result at \${outdir}/\${project}

A.) Pipeline configured with Trim_Galore, Arriba_Fusion, STAR_Fusion, RSEM & ML_RF& ML_RANK & ML_tSNE, and GATK & RNAseqCNV

```
(base) [wzhang42@splprhpc04 Nextflow Divia]$ ls NTU BALL Singularity 12 24 2022
Arriba_Fusion          RNAseq_RSEM
ExpectedCount_Matrix  RSEM Reformatted_Count
FPKM_Matrix           STAR_Fusion
GATK_ApplyBQSR        STAR Mapping
GATK_BaseRecalibrator STARMapping Adaptor_Arriba_Fusion
GATK_HaplotypeCaller  STARMapping Adaptor_RNAseq_RSEM
GATK_SplitNCigarReads STARMapping Adaptor_RNAvar_GATK
GATK_VariantFiltration STARMapping Adaptor_STAR_Fusion
ML_Classifier_RANK    TPM_Matrix
ML_Classifier_RF      Trim Galore
ML_Classifier_tSNE    Variant_ANNVAR
ModifiedCount_Matrix  Zcat_MergeFastq
RNASeqCNV
```

B.) Pipeline configured with RSEM & ML_RANK & ML_tSNE, Pindel, and iAdmix

```
(base) [wzhang42@splprhpc04 DIVIA Pindel iAdmix Test]$ ls
ExpectedCount_Matrix ML_Classifier_tSNE RNAseq_RSEM STARMapping Adaptor_Pindel
FPKM_Matrix          ModifiedCount_Matrix RSEM Reformatted_Count STARMapping Adaptor_RNAseq_RSEM
iAdmix               Pindel              STAR Mapping              TPM_Matrix
ML_Classifier_RANK   Pindel2VCF          STARMapping Adaptor iAdmix Zcat_MergeFastq
```

Fig.12 Pipeline DIVIA.nf run result folder structure varied with different customized

Notice or Tips to User

In our multiple rounds test, we summarized the following practical rules that the users should notice.

Preparation of fastq samplesheet.

- 1.) If you start from aligned BAM, and convert the aligned BAM to fastq, then run DIVIA.nf, you may suffer the STAR mapping fail (use up all the allocated memory even at 320G per sample).

To solve this issue, you need to shuffle (**samtools bamshuf**) the BAM or sort the BAM by name (**Samtools sort -n**, then convert the shuffled.bam or sorted.bam to fastq.

- 2.) If the sample name contains pure number, such as 929, 1103, the RNAseqCNV module may fail. To solve this issue, please avoid the sample names with the pure number, such as 929→Sample_929 or 929_S, 1103→ Sample_1103 or 1103_S, ...
- 3.) The strandness per sample only can be one of the 4 cases: Stranded_total_RNA_PE100bp, Stranded_total_RNA_PE75bp, Unstranded_mRNA_PE100bp and Unstranded_mRNA_PE75bp. This is determined by the ML_Classifier_tSNE
- 4.) Currently, ML_Classifier_RF.R can be failed for the special application cases as only one sample or only duplicated samples.
- 5.) The fastq samplelist must be configured as a full path file. Because, fastq samplelist not only will be used as the entry cahnnel for DIVIA.nf (can be relative path file), but also will be used as a parameter that used in ML_Classifier_tSNE.R, which must be configured as a full path file.

DIVIA.nf

```
Read_Fastq_Ch = Channel
    .fromPath(params.fastq_filelist)
    .splitText()
    .splitCsv(sep: '\t') // use as data channel
```

nextflow.config

```
tSNE {
    LibraryType="Stranded_total_RNA_PE100bp"
    Metadata="${fastq_filelist}"
    ...
}

DIVIA.nf
Rscript ${params.ML_Classifier.RScript_Path}/ML_Classifier_tSNE.R \
    ${params.project} \
    ${Count_Matrix} \
    ${params.ML_Classifier.tSNE.Metadata} \
```

Conflict of R version and R_LIBS for R scripts.

The three ML_Classifier_xx.R are developed in R4.2.0, while RNAseqCNV.R developed in R4.1.0, and only can run in R.4.1.0. If not correctly configured, it will cause one of the R module fail. To solve this issue, you need to find the R_LIBS for R4.2.0 and R4.1.0, and configure them as the following:

```
R_4_2_LIBS="/home/wzhang42/R/tmp/x86_64-pc-linux-gnu-library/4.2"
R_4_1_LIBS="/home/wzhang42/R/tmp/x86_64-pc-linux-gnu-library/4.1"
```

Frustration from Rhel8 and its special technical configuration

In our running DIVIA.nf pipeline and other nextflow pipeline in rhel8 , we meet some module exit and throw out the following error message “**No space left on device**”, e.g

- 1.) " **samtools sort: failed to create temporary file**
"/tmp/206420236.tmpdir/samtools.444460.9241.tmp.0117.bam": No space left on device
- 2.) **gzip: SJAML031035_D1.Merge_Read_R1.fastq.gz: No space left on device**

tee: .command.err: No space left on device

3.Java gatk program

Java IOException -- No space left on device

Caused by: java.io.IOException: No space left on device

at sun.nio.ch.FileDispatcherImpl.write0(Native Method)

at sun.nio.ch.FileDispatcherImpl.write(FileDispatcherImpl.java:60)

We reasoned that the /tmp in rhel8 per user is strictly restricted for storing the intermediate files for some executable programs. Our solution is define the following environment variable in .bashrc_profile, which allow the executable program not to store the intermediate file in /tmp but in the user defined tmp folder.

```
export TMPDIR=/scratch_space/wzhang42/tmp
```

```
export NXF_TEMP=/scratch_space/wzhang42/tmp
```

```
export _JAVA_OPTIONS=-Djava.io.tmpdir=/scratch_space/wzhang42/tmp
```