得到测序数据f1,f2

1. 先用软件fastqc看测序情况，再用fastp用来测序数据除杂和接头。

用法：fastqc f1

fastp -f 10 -F 10 --detect\_adapter\_for\_pe -x -i R1.fq.gz -I R2.fq.gz -o R1.out.fq.gz -O R2.out.fq.gz.

得到clean数据r1,r2。这步用集群做吧

基因组用NGSQCToolkit

Conda activate toolkit

perl /public/home/yyang/src/NGSQCToolkit\_v2.3.3/QC/IlluQC\_PRLL.pl -pe 2\_L4\_1.fq 2\_L4\_2.fq 2 5 -l 40 -s 20

1. jellyfish && GenomeScope 评估基因组

用法；jellyfish count -C -m 21 -s 1000000000 -t 10 \*.fastq -o reads.jf

jellyfish histo -t 10 reads.jf > reads.histo

然后数据上传到http://genomescope.org/

当然为了防止网站出意外，我弄了本地化。

具体方法： 打开cmd F: cd F:\D盘微信文件\R语言练习\genomescope

F:\R-3.6.1\bin\Rscript genomescope.R reads.histo 31 100 F:\D盘微信文件\R语言练习\genomescope\wc 1000 结果文件在F:\R-3.6.1\bin\Rscript genomescope.R reads.histo 31 100 F:\D盘微信文件\R语言练习\genomescope\wc 。

Linux安装R语言：conda install r-essentials --yes

1. 二代数据拼接，估计基因组大小软件SPAdes

spades.py -1 1.fastq -2 2.fastq --only-assembler -o /dat1/bgq/Genome/illumina/Rawdata/assembly -t 10 -k 77

1. 三代测序拼接基因组（最好的软件应该是falcon，但是参数复杂，我还没搞出来）

这里用的是软件wtdbg2

三代数据的存储是subreads.bam文件，我们可以利用工具把它转为fasta文件然后进行后续分析。

wtdbg2 -t 10 -i /dat1/bgq/Genome/sequel/pacbio.fastq -fo fungi -g 80m

wtpoa-cns -t 16 -i dbg.ctg.lay.gz -fo dbg.raw.fa

检验一下：用busco为90.0，这个是在没有利用二代数据纠错的情况下。我觉得还行。

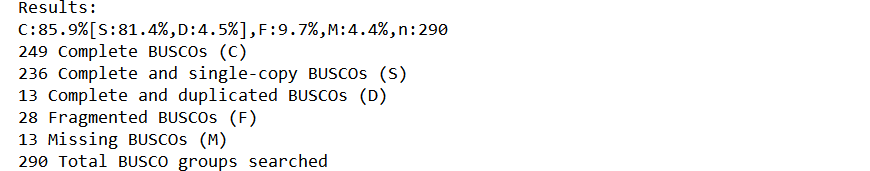
我拼出来反正和公司的差了4m，我觉得差不多。。

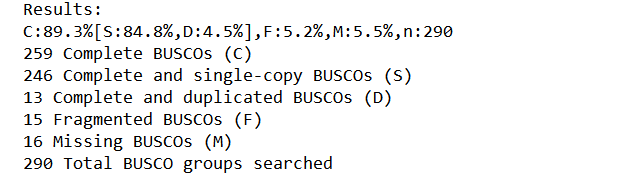
然后用软件mecat2

步骤mecat.pl correct nephila\_config\_file.txt

mecat.pl assemble nephila\_config\_file.txt

Mecat2的busco的结果是





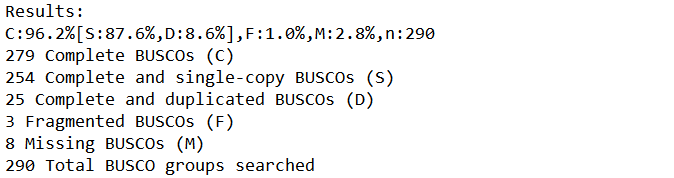
然后拼接出来的三代数据需要经过二代数据纠错以及矫正，用的软件是nextpolish。

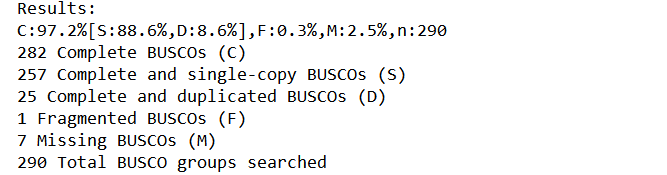
配置sgs.fofn文件以及run2.cfg文件，这些在我的服务器里面都有。

Cd /mnt/usb/sandai/polish

nohup /mnt/usb/sandai/NextPolish/nextPolish run2.cfg &

最后nextPolish run2.cfg 。得到结果如下。





很明显busco真菌完整度好多了。

运行busco:

nohup run\_BUSCO.py -i dbg.raw.fa -o result -l /root/root/busco/library/fungi\_odb9 --mode geno &

FinisherSC升级基因组：

/public/home/yyang/miniconda3/envs/py2.7/bin/python /public/home/yyang/finishingTool-master/finisherSC.py up /public/home/yyang/mummer4/bin

得到的improve3.fasta就行我们需要的简单的序列。

1. 重复屏蔽。

1.先用repeatmask预测

/dat1/bgq/software/RepeatMasker/RepeatMasker -norna -pa 60 -e ncbi -dir /dat1/bgq/Genome/repeat/repeatmask -gff Wolfiporia\_cocos.fasta

2.再用repeatmoder挑出重复序列，然后用repeatmask用其建的库进行预测

BuildDatabase -name Fusarium\_tricinctum -engine ncbi GCA\_900382705.2\_FTRI.INRA104.GCA2018.2\_genomic.fna

RepeatModeler -database wc -engine ncbi -pa 46

RepeatMasker -pa 4 -gff -lib Fusarium\_tricinctum-families.fa -dir repeat2 GCA\_900382705.2\_FTRI.INRA104.GCA2018.2\_genomic.fna

3.用脚本把两个结果合并

perl $0 repeatMasker/genome.fasta.out repeatModeler/genome.fasta.out > genome.repeat.stats

1. braker2流程进行基因预测。

注意服务器安装软件gmes的话一定要按照说明书上走

Perl scripts are configured with default Perl location at "/usr/bin/perl".

Default path to Perl can be changed by script "change\_path\_in\_perl\_scripts.pl"

from GeneMark\* distribution folder.

比如这句话，我们就要把perl环境换成我们的不能用默认的usr/bin/perl。

Braker流程看《如何配置云服务器》

1. go。Kegg等等基因功能注释
2. Interpro注释：在自己的或者服务器运行脚本
3. Pfam注释 因为网站现在限定一次最多只能500个基因注释了。所以只能比对弄了。

Conda activate pseudofinder

建库：hmmpress Pfam-A.hmm

比对： hmmscan --cpu 8 --tblout TrinotatePFAM.out $Pfam $proteins > pfam.log

python interpro.py --multifasta test\_proteins.fasta --maxJobs 25 --useSeqId --email life333@webmail.hzau.edu.cn --outformat tsv 可以用nohop 一次运行多个

interpro也可用于go注释

Go注释整理两个关键脚本 go.pl 以及 根据一个列表对应.pl脚本

在

/public/home/yyang/database/nr 有自己的nr数据库，14分钟大概注释30个基因。12线程。1h大概128个基因

hic流程

1. 提取单倍体序列。 用 软件purge\_haplogs

minimap2 -ax map-pb genome.fa subreads.fasta.gz \

| samtools view -hF 256 - \

| samtools sort -@ 8 -m 1G -o aligned.bam -T tmp.ali ---------得到mapping。

purge\_haplotigs readhist -b aligned.bam -g genome.fasta -t 20 得到统计图

由图可以看出是不是二倍体

1. purge\_haplotigs contigcov -i cns\_p\_ctg.aligned.sd.bam.gencov -o coverage\_stats.csv -l 20 -m 75 -h 190 20 ，75，190这几个参数自己定

purge\_haplotigs purge -g cns\_p\_ctg.fasta -c coverage\_stats.csv -b cns\_p\_ctg.aligned.sd.bam -t 4 -a 60 得到结果

curated.fasta：新的单倍型组装

Python共线性图制作不讲了

/root/gongxianxing/amewc/12/e 虚拟机下的

以及 https://github.com/hexin010101/MCscan-Analysis-process

基因组TRNA，rRNA，ssr注释

tRNAscan-SE -o tRNA.out -f rRNA.ss -m tRNA.stats /root/TRNA/Wolfiporia\_cocos\_genome.fasta（在tRNAscan-SE文件夹里运行）在自己虚拟机上运行。

rRNA注释，看生物信息文件夹下的rRNA注释。

ssr注释：在F:\D盘微信文件\SSR文件夹里面

使用misa.pl seqs.fasta （里面的配置文件我已经配好了，无需更改）

密码子分析 分析软件在

F:\D盘微信文件\R语言练习\R脚本111\双y轴\Win32CodonW\Win32

方法：<https://www.doc88.com/p-38961713546.html>

可视化：D:\云服务器从零开始\codon-usage-tables-master\codon\_usage\_data

和 F:\C盘QQ文件\Wolfiporiacocos\Wolfiporiacocos\Genome\rjchallis-codon-usage-bc184b0

用的是CDS序列。然后就可以作图了，太复杂了我就不重现了。公司那个我觉得就是瞎搞的，所以这个咱们改改数据好交差。

mummer作图: mummer2circos -l -c -r genomes/NZ\_CP008827.fna -q genomes/\*fna

在这之前要先conda activate mummer2circos

转录组可变剪切位点分析

[https://mp.weixin.qq.com/s?src=11&timestamp=1599888389&ver=2579&signature=dh\*wBmVlFXTATfJ9RaclRxpkll1jiq5gUQimf9VZNkhN-qmte\*h3SiECpyjQibq4\*OoBpoS0R4v1QHDLN3oXmXeF9JyPJbFccbgxurCZxmaVylx-Lj\*J9YV5IVhJ\*m-D&new=1](https://mp.weixin.qq.com/s?src=11&timestamp=1599888389&ver=2579&signature=dh*wBmVlFXTATfJ9RaclRxpkll1jiq5gUQimf9VZNkhN-qmte*h3SiECpyjQibq4*OoBpoS0R4v1QHDLN3oXmXeF9JyPJbFccbgxurCZxmaVylx-Lj*J9YV5IVhJ*m-D&new=1)

extract-as \

transcript.gtf \

ref.fa.hdrs > as\_events.txt

下载数据

ascp -v -k 1 -T -l 200m -P33001 -i /public/home/yyang/.aspera/connect/etc /asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR768/SRR7683 16/SRR768316\_2.fastq.gz ./

转录组拼接：

1. 可做可不做（合并基因组：% cat \*.left.fq > reads.ALL.left.fq
2. % cat \*.right.fq > reads.ALL.right.fq

）

1. nohup Trinity --seqType fq --max\_memory 24G --samples\_file sample.txt --SS\_lib\_type FR --CPU 24 --jaccard\_clip --group\_pairs\_distance 500 &
2. 拼接结果的统计

TrinityStats.pl trinity\_out\_dir/Trinity.fasta>./assembly\_report.txt

出现问题：

UTF8 SyntaxError: Non-ASCII character ‘\xe5’ in file on line 12, but no encoding declared

解决：文件的最上方加上

#!/usr/bin/python

# -\*- coding: UTF-8 -\*-

出现问题 2，trinity在用bowtie和rsem统计时出现Can't locate rsem\_perl\_utils.pm

解决：找到Can't locate rsem\_perl\_utils.pm并且放到trinity放perl模块的位置就可以了

1. 比对和统计表达量

align\_and\_estimate\_abundance.pl --transcripts trinity\_out\_dir/Trinity.fasta --seqType fq --left myc2\_1.fq.gz --right myc2\_2.fq.gz --est\_method RSEM --aln\_method bowtie --trinity\_mode --prep\_reference --output\_dir rsem5\_Sp\_log\_outdir --SS\_lib\_type FR（几个样本做几次）。

1. 为了后面的转录本的注释，先提取最长转录本序列。

/root/root/Trinity/opt/trinity-2.11.0/util/misc/get\_longest\_isoform\_seq\_per\_trinity\_gene.pl trinity\_out\_dir/Trinity.fasta >unigene.fasta

去沉余：cd-hit-est -i unigene.fasta -o unigene2.fasta -T 1 -M 1000

1. 获得表达量矩阵：

abundance\_estimates\_to\_matrix.pl --est\_method RSEM --out\_prefix /mnt/usb/trinity/Trinity\_trans rsem1\_Sp\_log\_outdir/cut1RSEM.isoforms.results rsem2\_Sp\_log\_outdir/cut2RSEM.isoforms.results rsem3\_Sp\_log\_outdir/cut3RSEM.isoforms.results rsem4\_Sp\_log\_outdir/myc1RSEM.isoforms.results rsem5\_Sp\_log\_outdir/myc2RSEM.isoforms.results rsem6\_Sp\_log\_outdir/myc3RSEM.isoforms.results

Genemap参数不用管

1. 差异分析

run\_DE\_analysis.pl --matrix Trinity\_trans.counts.matrix --samples\_file samples.txt --method edgeR --output diffent

Sample.txt (格式)

cut cut1 cut1\_1.fq.gz cut1\_2.fq.gz

cut cut2 cut2\_1.fq.gz cut2\_2.fq.gz

Samples.txt（格式）

cut cut1RSEM

看看哪些是差异基因 ：sed '1,1d' diffent/Trinity\_trans.counts.matrix.cut\_vs\_myc.edgeR.DE\_results | awk '{ if ($5 <= 0.05) print;}' > 11111.txt

安装R语言，并下载

Linux安装R语言：conda install r-essentials --yes

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("edgeR")

BiocManager::install("qvalue")

BiocManager::install("Biobase")

然后做差异基因热图 这步要进入diffent文件夹。

/root/root/Trinity/opt/trinity-2.1.1/Analysis/DifferentialExpression/analyze\_diff\_expr.pl --matrix /mnt/usb/trinity/Trinity\_trans.TMM.EXPR.matrix --samples /mnt/usb/trinity/samples.txt -P 1e-3 -C 2

差异基因趋势图：

/root/root/Trinity/opt/trinity-2.1.1/Analysis/DifferentialExpression/define\_clusters\_by\_cutting\_tree.pl --Ptree 60 -R diffExpr.P1e-3\_C2.matrix.RData

run\_DE\_analysis.pl --matrix Trinity\_trans.counts.matrix --samples\_file samples.txt --method edgeR --output diffent 基因图再做一遍。。。。。。。

基因预测：对于得到的unigene2.fasta ，

1.TransDecoder.LongOrfs -t unigene2.fasta

2.diamond makedb --in uniprot\_sprot.fasta --db uniprot\_sprot.fasta

3.diamond blastp -d uniprot\_sprot.fasta -q transcripts.fasta.transdecoder\_dir/longest\_orfs.pep --evalue 1e-5 --max-target-seqs 1 > blastp.outfmt6

transDecoder.Predict -t transcripts.fasta --retain\_blastp\_hits blastp.outfmt6

然后做基因功能注释：

Nr,swissprot,go,kegg等等

真菌线粒体基因组注释拼接注释流程

参考：https://blog.csdn.net/yangl7/article/details/108081571?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522160327145519195188332752%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request\_id=160327145519195188332752&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_v2~rank\_v28-13-108081571.first\_rank\_ecpm\_v3\_pc\_rank\_v2&utm\_term=线粒体拼接&spm=1018.2118.3001.4187

筛选基因并进行拼接，用软件ARC ARC -c ARC\_config.txt

拼接出来的线粒体基因组与参考基因组比对 **这个方法预测的结果很难构成环状的DNA(寻找另外的方法)**

sh mummer.sh .

根据出来的结果进行手动这里数据，注意观察有无成环。拼接最后的基因组

最后的基因组与参考基因组比对验证。

提交到https://chlorobox.mpimp-golm.mpg.de/geseq.html 进行注释.

方法2：（好用）

get\_organelle\_from\_reads.py -1 /public/home/yyang/ibest-ARC-3831cb8/wcmt/2\_L4\_1.fq -2 /public/home/yyang/ibest-ARC-3831cb8/wcmt/2\_L4\_2.fq -F fungus\_mt -R 10 -k 21,45,65,85,105 -o wcmt2

(conda activate mtDNA 后面默认参数即可)

再用mummer.sh检查

有参转录组：1.fastp除杂 [链非特异性建库]

fastp -f 10 -F 10 --detect\_adapter\_for\_pe -x -i myc1\_L2\_1.fq.gz -I myc1\_L2\_2.fq.gz -o 1\_1.fq.gz -O 1\_2.fq.gz

fastp -f 10 -F 10 --detect\_adapter\_for\_pe -x -i myc2\_L2\_1.fq.gz -I myc2\_L2\_2.fq.gz -o 2\_1.fq.gz -O 2\_2.fq.gz

2.hisat2建库比对

hisat2-build -f Wolfiporia\_cocos\_genome.fasta wc

hisat2 -p 12 -x wc -1 1\_1.fq.gz -2 1\_2.fq.gz | samtools sort -@ 10 > 1.bam

hisat2 -p 12 -x wc -1 2\_1.fq.gz -2 2\_2.fq.gz | samtools sort -@ 10 > 2.bam

1. htseq计数

Conda activate zlu gffread old.gff3 -T -o new.gtf(似乎只认识gtf格式)

htseq-count --stranded=no -f 1.bam new.gtf > 1.txt (得到count)

stringtie -p 8 -G new.gtf -o 1.gtf 1.bam

stringtie --merge -p 8 -G new.gtf -o stringtie\_merged.gtf mergelist.txt

gffcompare -r new.gtf -G -o merged stringtie\_merged.gtf

stringtie -e -B -p 8 -G stringtie\_merged.gtf -o ballgown/ERR188044\_chrX/ERR188044\_chrX.gtf ERR188044\_chrX.bam

gffread -w transcripts.fa -g Wolfiporia\_cocos\_genome.fasta stringtie\_merged.gtf(参考基因组提取转录本)

/public/home/yyang/TransDecoder-TransDecoder-v5.5.0/util/gtf\_genome\_to\_cdna\_fasta.pl stringtie\_merged.gtf Wolfiporia\_cocos\_genome.fasta > transcripts1.fasta (然后tbtools提取可能的多的基因)

/public/home/yyang/TransDecoder-TransDecoder-v5.5.0/util/gtf\_to\_alignment\_gff3.pl stringtie\_merged.gtf > stringtie\_merged.gff3 (gtf转为gff3)

/public/home/yyang/TransDecoder-TransDecoder-v5.5.0/TransDecoder.LongOrfs -t transcripts.fasta

diamond寻找同源证据:

diamond blastp -d ~/database/swissprot/uniprot\_sprot.fasta -q transcripts1.fasta.transdecoder\_dir/longest\_orfs.pep --evalue 1e-5 --max-target-seqs 1 > blastp.outfmt6

/public/home/yyang/TransDecoder-TransDecoder-v5.5.0/TransDecoder.Predict \

-t transcripts1.fasta \

--retain\_blastp\_hits blastp.outfmt6

上面的没匹配上的序列和下面的没有编译蛋白的进行比对筛选出序列

最后1.pfam注释看看有没有编码什么基因

2.

<http://cpc.gao-lab.org/programs/cpc.do> cpc预测

1. CNCI

下载安装CNCI

<https://github.com/www-bioinfo-org/CNCI>

python /public/home/yyang/zzzzz/CNCI-master/CNCI.py -f lncnipre.fasta -o CNCI\_out -m pl -p 8

最后经过三个软件筛选得到了2347个可能是lncRNA的基因

最后筛选长度大于200bp的基因 参考文章 <http://interact.majorbio.com/article/detail/296>

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

这个结果与一些文章的结果相似，大于1000bp的基因相当少。所以应该是可信的。