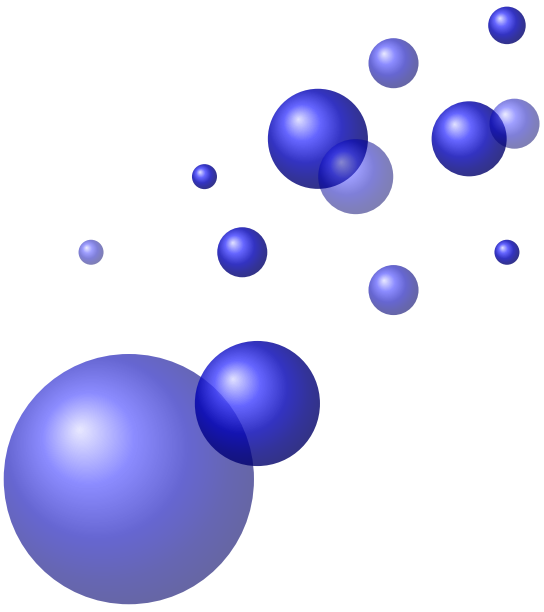
M2 BBS UNIVERSITE PAUL SABATIER

Rapport - NEXTFLOW

Mise en place d'une pipeline automatisée avec NEXTFLOW sur le serveur Genotoul - à partir de données RNAseq -

Auteure :
Julie CAMPOS

Sous la supervision de :
Dr. Sarah Maman Haddab



Octobre 2023

Table des matières

CHAPITRE 1

EXEMPLE TOMATE

Le but de cette partie est de construire une pipeline automatisée grâce à Nextflow pour traiter des données RNAseq. Nous utiliserons un serveur de calcul (pour bénéficier de bonnes ressources informatiques). Nous choisirons d'utiliser une pipeline complète, déjà optimisée et éprouvée par la communauté scientifique : nf core (RNAseq).

1.1 PREPARATION ENVIRONNEMENT DE TRAVAIL (Q1)

1.1.1 CONNEXION AU SERVEUR GENOLOGIN

— **Se déconnecter de conda :** conda deactivate
Pour éviter les conflits. Sur le serveur, un environnement sera créé automatiquement.

— **Se connecter à Genologin :** ssh -XY laurier@genologin.toulouse.inrae.fr
pwd : f1o2r3 !

1.1.2 ORGANISATION DU TRAVAIL

— **Aller dans répertoire de travail 'work' :** cd work
Dans 'home', se trouve 2 répertoires : 'save' et 'work'. Le répertoire 'save' permet de conserver les données sur le long terme, alors que le répertoire 'work' est nettoyé régulièrement. En revanche, les calculs DOIVENT ETRE IMPERATIVEMENT être réalisés dans le répertoire 'work'.

— **Créer les répertoires de travail :**
mkdir PROJET_NEXTFLOW ;
mkdir PROJET_NEXTFLOW/TOMATES ;
cd PROJET_NEXTFLOW/TOMATES ;
mkdir FASTQ ;
mkdir GENOME_REF ;

1.1.3 TÉLÉCHARGEMENT DES DONNÉES

En entrée, la pipeline a besoin des échantillons (format .fastq) et le génome de référence (annotations au format .gtf, et la séquence du génome au format .fasta/.fna/.fa).

Génome de référence

— **Lien :** `http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/`
Aller sur ce lien, clic droit pour copier le lien, pour l'insérer dans la commande suivante.

— **Commande bash :**
wget `https://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/ITAG2.3_genomic_Ch6.fasta`
wget `https://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/ITAG2.3_genomic_Ch6.gtf`

— **Vérification :** ls et/ou ls -lrt

— **Output :** OK voir fig.1.1

Fastq

```
laurier@genologin2 ~/work/PROJET_NEXTFLOW/GENOME_REF $ ls
ITAG2.3_genomic_Ch6.fasta  ITAG2.3_genomic_Ch6.gtf
laurier@genologin2 ~/work/PROJET_NEXTFLOW/GENOME_REF $ ls -lrt
total 47520
-rw-r--r-- 1 laurier formation  2034585 30 sept.  2022 ITAG2.3_genomic_Ch6.gtf
-rw-r--r-- 1 laurier formation 46617169 30 sept.  2022 ITAG2.3_genomic_Ch6.fasta
```

FIGURE 1.1 – Vérification téléchargement - génome de référence .gtf et .fasta - Tomate

Information sur les échantillons ARN

— paired-end

— unstranded

Téléchargement

— **Lien :** `http://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/`
Aller sur ce lien, clic droit pour copier le lien du fichier désiré, pour l'insérer dans les commandes suivantes :

— **Commande bash :**
wget `https://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/MT_rep1_1_Ch6.fastq.gz`
wget `https://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/MT_rep1_2_Ch6.fastq.gz`
wget `https://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_TOMATES/WT_rep1_1_Ch6.fastq.gz`
wget `https://genoweb.toulouse.inra.fr/~sigenae/sarah/UPS/DATA/TP_OMATES/WT_rep1_2_Ch6.fastq.gz`

— **Vérification :** ls ou ls -lrt

— **Output :** OK voir fig.1.2

```
laurier@genologin2 ~/work/PROJET_NEXTFLOW/FASTQ $ ls
MT_rep1_1_Ch6.fastq.gz  WT_rep1_1_Ch6.fastq.gz
MT_rep1_2_Ch6.fastq.gz  WT_rep1_2_Ch6.fastq.gz
laurier@genologin2 ~/work/PROJET_NEXTFLOW/FASTQ $ ls -lrt
total 455984
-rw-r--r-- 1 laurier formation 128793000 30 sept.  2022 MT_rep1_1_Ch6.fastq.gz
-rw-r--r-- 1 laurier formation 106238937 30 sept.  2022 WT_rep1_1_Ch6.fastq.gz
-rw-r--r-- 1 laurier formation 126966181 30 sept.  2022 MT_rep1_2_Ch6.fastq.gz
-rw-r--r-- 1 laurier formation 104918284 30 sept.  2022 WT_rep1_2_Ch6.fastq.gz
```

FIGURE 1.2 – Vérification téléchargement - des 4 échantillons - Tomate

1.1.4 ORGANIGRAMME

voir fig.1.3

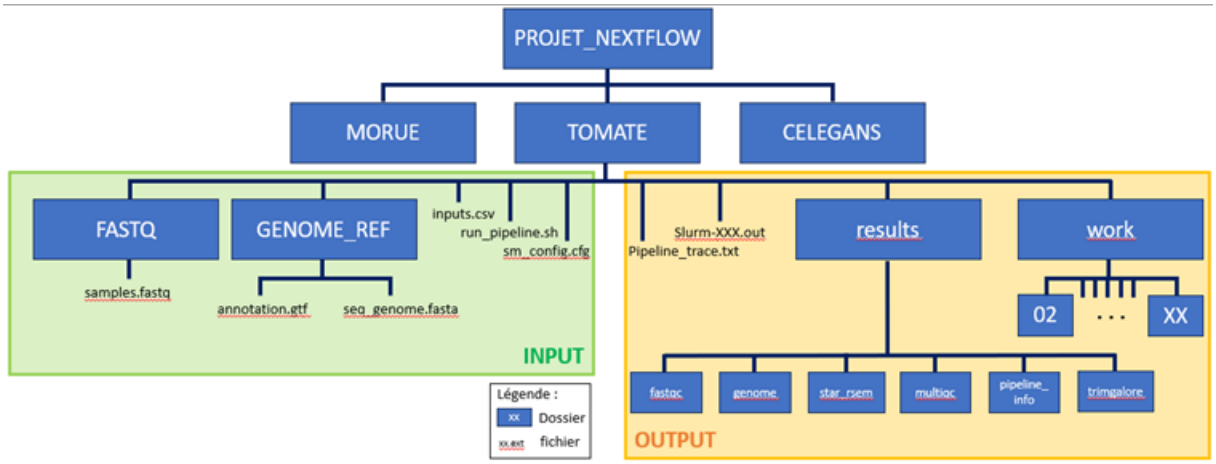


FIGURE 1.3 – Hiérarchie répertoires - Organigramme Tomate

1.2 CREATION ET LANCEMENT DE LA PIPELINE (Q2)

Nous travaillerons tout d'abord sur un seul chromosome (6) pour établir une pipeline (plus rapide).

1.2.1 CHOIX DE LA PIPELINE

Nous testerons la pipeline de nf core/RNAseq. Celle-ci va des données brutes, à l'analyse différentielle de l'expression des gènes (Deseq2, basé sur le comptage des reads) incluant des contrôles qualité à toutes les étapes : fastqc/multiQC (avant, après trimmage, après mapping).

— **Source :** `https://nf-co.re/rnaseq/3.12.0`

— **Pipeline :** voir fig.1.4

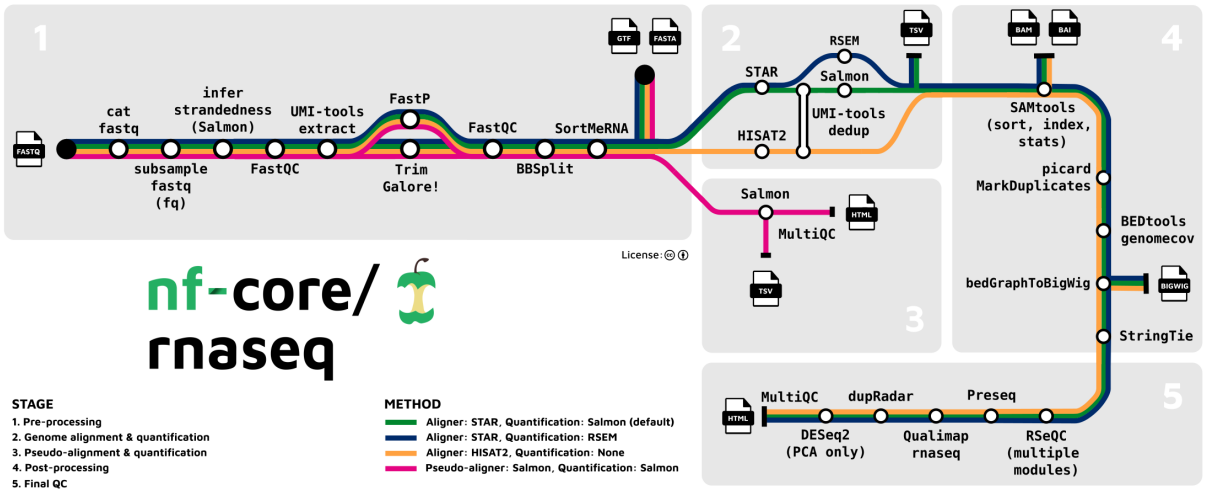


FIGURE 1.4 – Pipeline nf core RNAseq

— **Description, choix des étapes :**

- **Première étape de CQ :** Après avoir compilé les fichiers fastq (cat) avec FastQC
- **Séparation des brins :** selon si brin sens ou antisens avec fq et Salmon
- **Deuxième étape de CQ :** avec FastQC
- **Etape de nettoyage des séquences :**
  — Retire l'adaptateur 3' du séquençage avec Trim Galore
  — Ote les contaminants avec BBSplit
  — Ote l'ARN ribosomal avec SortMeRNA
- **Choix de la méthode d'alignement/mapping :** Nous choisissons la méthode STAR (arbitraire)
- **Choix de la méthode de comptage des reads (séquences) :** Nous choisissons la méthode RSEM intégrées dans les options de STAR (arbitraire)
- **Deuxième comptage :** avec featurecounts
- **Préparation des fichiers de sortie :**
  — Tri(obligatoire), index et produits des statistiques avec Samtools
  — Création des fichiers BigWig avec BedTools, BamCoverage, bedGraphToBigWIg → permettra d'être visualisé avec IVG (par exemple)
- **Stringtie** permet d'analyser les différents isoformes. Il s'agit de différencier les plusieurs transcripts éventuels pour une même position de mapping sur le génome de référence. très utile pour l'étude du transcriptome.
- **Dernières étapes de CQ :**
  — avec RSeqQC,Qualimap,dupRadar,Preseq,DESeq2
  — compile toutes les étapes de CQ avec MultiQC

1.2.2 PREPARATION DE LA PIPELINE NEXTFLOW

Nous travaillons dans le répertoire 'TOMATES'.

Fichier sm_config.cfg

Rôle

Création

— **Création du fichier :** touch sm_config.cfg

— **Ecriture :** nano sm_config.cfg

```
trace {
enabled = true
file = 'pipeline_trace.txt'
fields = 'task_id,name,status,exit,realtime,%cpu,rss,script'
}
```

— **Vérification :** head smconfig.cfg

— **Output :** OK

Fichier inputs.csv

Rôle Le but est de définir le plan d'expérience, selon les noms des fichiers/ ou indiqué sur le bioprojet NCBI/GEO.

— **Information sur les échantillons :**
  - tomate MT → mutant
  - tomate WT → sauvage
  - un un seul réplicat par condition (rep1)
  - ne sont pas orientés (UNSTRANDED)
  - paired-end (2 fichiers par réplicats) [1]

— un echantillon = une ligne!!! surtout ne pas aller à la ligne!! ATTENTION SAUT DE LIGNE ; ESPACE, PATH (chech chacun avec ls)

— Chacune des informations sont séparées par des virgules (format csv)

— A besoin du chemin absolu des fichiers des échantillons (tous à vérifier pour éviter les erreurs simples)

— **format entrée des échantillons :** fastq.gz (compressé, prend moins de place)

Création

— **Création du fichier :** touch inputs.csv

— **Vérifier en amont les chemins absolus :**
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/MT_rep1_1_Ch6.fastq.gz
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/MT_rep1_2_Ch6.fastq.gz
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/WT_rep1_1_Ch6.fastq.gz
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/WT_rep1_2_Ch6.fastq.gz

— **Ecriture :** nano inputs.csv
group,replicate,fastq_1,fastq_2,strandedness
mutant,1,/home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/MT_rep1_1_Ch6.fastq.gz,
/home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/MT_rep1_2_Ch6.fastq.gz,unstranded
wild,1,/home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/WT_rep1_1_Ch6.fastq.gz,
/home/laurier/work/PROJET_NEXTFLOW/TOMATES/FASTQ/WT_rep1_2_Ch6.fastq.gz,unstranded

— **Vérification :** more inputs.csv

— **Output :** OK

Fichier run_pipeline.sh

Ressources

— `https://genotoul-bioinfo.pages.mia.inra.fr/use-nextflow-nfcore-course/nextflow/run_options.html`

— `https://nf-co.re/rnaseq/3.12.0/parameters`

— man sbatch

1. c'est le fait de lire les deux extrémités du fragment d'intérêt au moment du séquençage, produit ainsi 2 fichiers en sortie, un pour chaque sens

Rôle Le but de se fichier est de renseigner la liste des tâches désirées pour notre pipeline. Il s'agit du fichier d'exécution de notre pipeline. Il permet d'automatiser les tâches. C'est celui ci qui sera exécuté (en sbach) et appellera les autres fichiers que nous avons construit.

— De nombreuses options sont disponibles pour la commande nextflow run :

- des options de configuration telles que **'-profile'** → genotoul (limite les ressources à 6G de mémoire, 48 CPU)
- des options de rapport d'exécution (trace) → nous utiliserons cette option (voir sec.1.2.2
- des options de dépendance et d'environnement (conda, docker, singularity) → nous n'utiliserons pas ici
- des options de version des outils utilisées dans le workflow/pipeline (**'-latest'** → nous n'utiliserons pas ici
- des options d'exécution telles que **'-resume'**
  — l'option **'-resume'** est à mettre seulement en deuxième instance (si le travail échoue).
  — Il permet de réutiliser les actions réussies d'une même pipeline (run_pipeline.sh), et d'économiser considérablement des ressources et du temps de calcul. Au lieu de reprendre le workflow depuis le début, il n'effectue que les tâches non terminées. Il permet ainsi la continuité d'un workflow.
  — Pour fonctionner, Nextflow utilise un unique ID pour vérifier si le répertoire existe, le statut de chaque commande et si les fichiers de sortie sont présents. Si les conditions sont satisfaites, alors la tâche n'est pas lancée et conservent les précédentes exécutions. [2]

— **Options batch choisies**

- **nom du job sur cluster :** #SBATCH -J JulieCAMPOS
- **choix de la queue de travail :** #SBATCH -p workq
- **maximum 6G de mémoire :** #SBATCH –mem=6G
- **maximum 24h :** #SBATCH –time=24 : 00 : 00
- **définir le nombre de cpu (max pour profile genotoul 48) :** #SBATCH –cpu=48 [3]

Création du fichier

— **Création du fichier :** touch run_pipeline.sh

— **Vérifier en amont les chemins absolus :**
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/inputs.csv
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/GENOME_REF/ITAG2.3_genomic_Ch6.gtf
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/GENOME_REF/ITAG2.3_genomic_Ch6.fas
ls /home/laurier/work/PROJET_NEXTFLOW/TOMATES/sm_config.cfg

— **Ecriture :** nano run_pipeline.sh

— **Vérification :** more run_pipeline.sh

— **Output :** OK

2. `https://www.nextflow.io/blog/2019/demystifying-nextflow-resume.html`
3. La définition des ressources pouvait aussi être dans le fichier config → process

```
laurier@genologin2 ~/work/PROJET_NEXTFLOW/TOMATES $ more run_pipeline.sh
#!/bin/bash
#SBATCH -J JulieCAMPOS
#SBATCH -p workq
#SBATCH --mem=6G
#SBATCH --time=24:00:00
#SBATCH --cpus-per-task=48

module purge
module load bioinfo/nfcore-Nextflow-v21.04.1
input=/home/laurier/work/PROJET_NEXTFLOW/TOMATES/inputs.csv
gtf=/home/laurier/work/PROJET_NEXTFLOW/TOMATES/GENOME_REF/ITAG2.3_genomic_Ch6.gtf
fasta=/home/laurier/work/PROJET_NEXTFLOW/TOMATES/GENOME_REF/ITAG2.3_genomic_Ch6.fasta
config=/home/laurier/work/PROJET_NEXTFLOW/TOMATES/sm_config.cfg
nextflow run nf-core/rnaseq \
-r 3.0 \
-profile genotoul \
--input $input \
--fasta $fasta \
--gtf $gtf \
--aligner star_rsem \
-c $config
```

FIGURE 1.5 – Script fichier .sh - Tomate

1.2.3 RUN/LANCEMENT DE LA PIPELINE

Run Nous inclurons dans le rapport uniquement les résultats de la pipeline effectué sur les échantillons entier.

- **Commande bash :** sbatch run_pipeline.sh
- **Output :** Submitted batch job 50749365

Vérification

- **Commande bash :** seff 50749365
- **Output :** voir fig.1.6

```
laurier@genologin2 ~/work/PROJET_NEXTFLOW/TOMATES $ seff 50749365
Job ID: 50749365
Cluster: genobull
User/Group: laurier/formation
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 48
CPU Utilized: 00:02:38
CPU Efficiency: 0.56% of 07:51:12 core-walltime
Job Wall-clock time: 00:09:49
Memory Utilized: 1.84 GB
Memory Efficiency: 30.64% of 6.00 GB
```

FIGURE 1.6 – Résultats statut du job 50749365 - Tomate

Explication de la sortie du seff La commande 'seff <numero_job>' nous donne des informations sur un job :

- **Cluster :** nom du cluster où les calculs sont effectués
- **User/group :** le nom de l'utilisateur et le groupe dans lequel il appartient (définit par l'administrateur)
- **State :** Renseigne sur le statut d'un job
- **Ressources utilisées :** [4]
  - **Nodes :** le nombre de noeud utilisé (ici 'work', un seul)
  - **CPU :**
    - le nombre de cœurs maximun utilisable (définit en option ou par défaut)
    - le nombre de cœurs utilisés pour le travail
    - l'efficacité des cœurs
  - **Job time :** Renseigne sur la durée du travail
  - **Memory :** Renseigne sur la mémoire utilisée par le travail, et brute et pourcentage du maximum définit

Log

- **Contenu :** Le fichier 'slurm' renseigne sur toute la pipeline Nexflow exécutée. C'est un fichier important à consulter en cas de problème d'exécution (failed). Il enregistre les actions exécutées.
- **Commande bash :** tail -n 200 slurm-50749365.out
- **Output :** voir fig.2.5

4. Les ressources maximales allouées dépendent du type de profil (nous c'est le profil genotoul)

FIGURE 1.7 – Résultats log du job 50749365 (fichier slurm) - Tomate

1.3 RESULTATS ET INTERPRÉTATION (Q3)

1.3.1 RECUPERATION DES RESULTATS EN LOCAL

— **Commande bash :**
scp laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/TOMATES/pipeline_trace.txt .
scp laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/TOMATES/slurm-50749365.out .
scp laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/TOMATES/results/multiqc/star_rsem/multiqc_report.html .
pwd :

— **Output :** OK

1.3.2 RESULTATS - OUTPUT de la PIPELINE (Q3.1- Expliquer les fichiers de sortie)

La pipeline que nous avons lancée avec Nextflow créer de nombreux fichiers en sortie, ainsi que 2 répertoires :'work' et 'results' :
voir fig.1.8

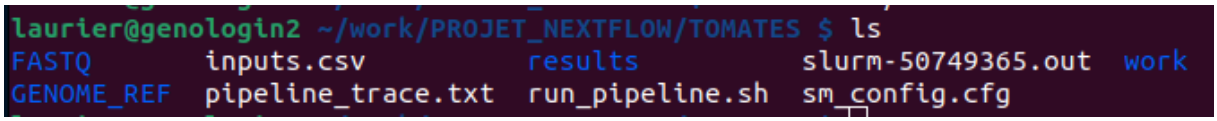


FIGURE 1.8 – Fichiers et répertoires de sortie - Tomate

Répertoire 'work' Le répertoire 'work' contient tous les fichiers intermédiaires, produits à chaque étape de la pipeline <nf core RNAseq> choisie : cf fig. 1.9
Par exemple, dans le sous-répertoire '02', nous trouvons la sortie de la commande Samtools,

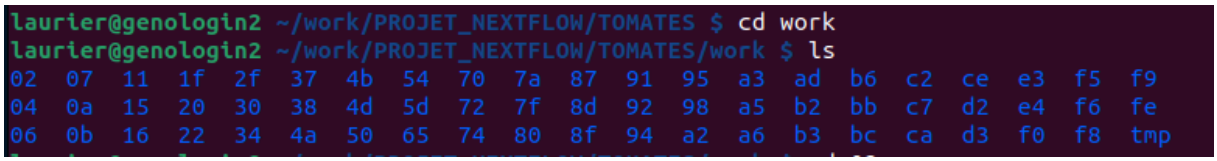


FIGURE 1.9 – Contenu du répertoire de sortie 'work' - Tomate

pour le génome de référence 'ITAQ2.3_genomic_Ch6.fasta'. Après l'indexation, Samtools produit un fichier au format .fai. Si nous regardons d'autres répertoires (pris au hasard), il semblerait qu'un répertoire est créé pour chaque fichier d'entrée (échantillon), pour chaque action : voir fig.1.10
Le document 'workflow_summary.yaml' contient les enregistrements des actions au moment de l'étape du MultiQC.

Répertoire 'results' Le répertoire 'results'contient tous les fichiers de sorties des étapes de notre workflow (nf core RNAseq, voir sec.1.2.1). Un répertoire est créé par étape : voir fig.1.11
Voici le détail du contenu de chaque répertoire produit par la pipeline : voir fig.1.12

FIGURE 1.10 – Contenu du répertoire de sortie 'work' - Tomate

FIGURE 1.11 – Contenu du répertoire de sortie 'results' - Tomate

FIGURE 1.12 – Contenu du répertoire de sortie 'results' - Tomate

— **multiqc :** Le répertoire contient la sortie au format .html ainsi qu'un sous-répertoire contenant tous les documents utilisés dans le document final .html

— **star_rem :** Un répertoire est créé pour chaque étape intermédiaire du mapping avec star_rsem :

- **bigwig, dupradar, featurecounts, preseq, qualimap,rseqc**
- Stringtie
- les statistiques sur le mapping des reads, obtenu avec Samtools stats
- les comptages des reads du package star_rem
- les fichiers bam (obtenu avec star_rem)
- ainsi que leur fichier indexé, .bam.bai (obtenu avec Samtools index)

des statistiques sur l'étape de mapping

— **fastqc :** Sortie du premier contrôle qualité effectué avec FastQC, sur les fichiers .fasta des échantillons

— **genome :** Sortie de l'indexation du génome de référence effectué avec Samtools et star_rsem, ainsi que toutes les statistiques, log au moment de ces étapes (dans répertoire 'index/rsem')

— **trimgalore :** contient les sorties de l'étape de 'timmage' (ote l'adaptateur 3' du séquençage des reads, effectué avec TrimGalore) ainsi qu'un répertoire fastqc, contenant les sorties de la deuxième étape de CQ (effectué avec FastQC)

— **pipeline infos :** Contient les logs récoltés au moment de l'excéution de la pipeline. Il s'agit principalement des fichiers au format .txt et .html.

slurm-numero-du-job voir sec.1.2.1

pipeline-log.txt Ce document contient les détails d'exécution (les commandes 'bach', statistiques d'exécution → ressources utilisées, durée) de chaque action dans la pipeline : fig.1.13

1.3.3 INTERPRETATION CONTROLE - QUALITE -MULTIQC (Q3.2)

Présentation voir annexe A (chapitre.4)
MultiQC permet d'agréger, en un seul document .html, tous les contrôles qualités, statistiques, réalisés au cours de notre pipeline. En tout début du document, figure un résumé de comment et avec quelle pipeline a été généré le rapport. Juste après, nous pouvons trouver la synthèse des statistiques générales sur nos échantillons, tout au long de la pipeline. Par exemple :

— le % d'ARNribosomal trouvé et retiré

— le % de reads alignés

— le % de duplicats

— le nombre de reads alignés, de départ

Voici quelques sorties (graphiques, et statistiques) des contrôles qualités effectués et intégrés dans le rapport MultiQC tout au long de la pipeline :

— **fastqc :** avant (sur les .fastq/RAW)et après l'étape de 'trimmage' (ote l'adaptateur 3')

— **featurecounts :** donne la répartition des reads pour chaque biotype contenu dans la séquence

— **Samtools stat :** vérifie le % de read alignées, et beaucoup d'indication sur l'étape d'alignement (mapping)

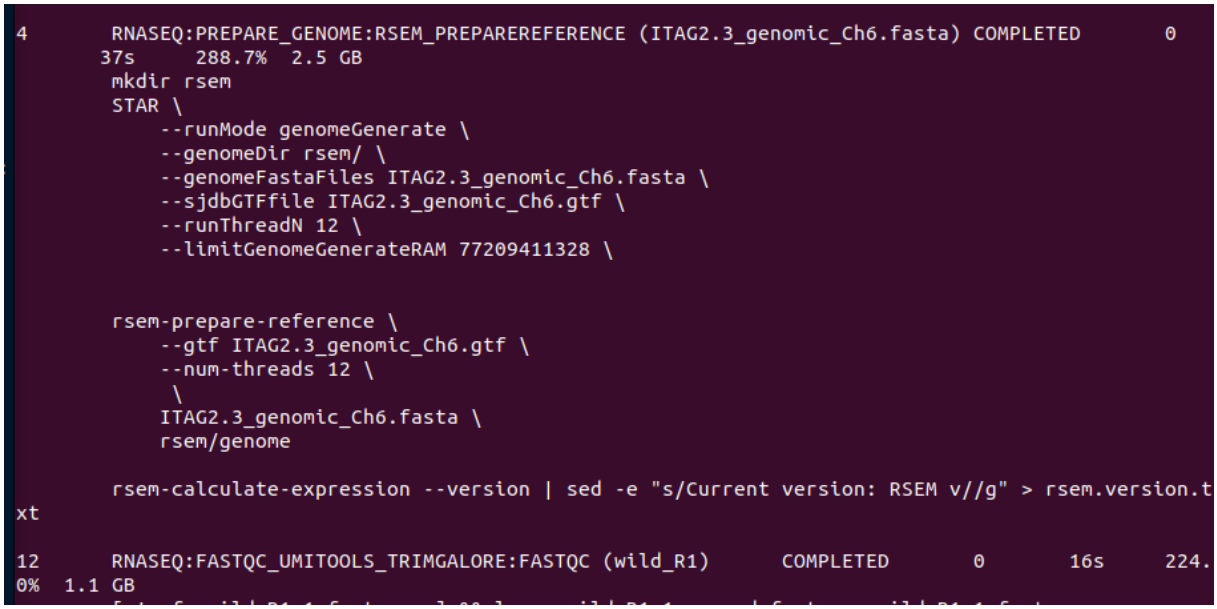


FIGURE 1.13 – Contenu du fichier de sortie 'pipeline_trace' - Tomate

— **Dupradar :** contrôle et génère des statistiques sur les duplicats trouvés(même read à la même position)

— **QualiMap :** permet de classifier les reads alignés (mapped) selon si ce sont des exons, introns, région inter génique.

— **Junction annotation :** donne des renseignements sur les sites d'épissage détectés

— **.BAM :** vérification après l'étape de mapping (sur les fichiers .BAM obtenu avec STAR-rcem

MultiQC indique via un code couleur (vert → les résultats sont satisfaisants, orange→alerte/warning, rouge→ indique un gros problème) l'acceptabilité des résultats des rapports pour chaque étape. Il indique de surcroît le nombre d'échantillons concernés, et pourquoi.
Certains outils utilisés sont redondants. Malgré tout, je trouve important de les conserver. D'une part, parce que 2 vérifications valent mieux qu'une, et d'autre part, il permet de valider/confirmer les mesures effectuées (aucun outil n'est infaillible).
Le rapport MultiQC a le double avantage de rapidement trouver les erreurs potentielles sur nos données (à quelle étape il y a un soucis ?), et de merveilleusement bien synthétiser tous les contrôles qualités générés par chaque étape (fastidieux si nous devons récupérer les résultats un à un, et soit même créer un rapport). La consultation de ce document est une étape clef dans l'analyse des données. Il permet ou non de valider la bonne exécution de l'étape de pre-processing des données d'entrée et de décider sur la poursuite de l'analyse ou non. Cela permet de vérifier aussi la similarité des réplicats par exemple (reproductibilité expérimentale).

Interprétation Voici quelques résultats des rapports intéressant à considérer dans nos échantillons (Mutant et WT) :

— Dans l'échantillon 'mutant-R1', 99% des reads sont alignables, 3.3M de reads avec environs 18% de reads doublons

— seul le type 'Protein coding' est trouvé dans les échantillons

— plus de 80% des reads (sur les 2 échantillons) sont des exons.

— le nombre de reads alignés, de départ

— les résultats des fastqc montrent une bonne qualité des reads [5]

— nous remarquons également que le % de GC est équivalent sur nos 2 échantillons (et est congruent avec le % de GC du génome de la tomate, ce qui indique 2 choses : l'une les échantillons ne sont pas contaminés par autre chose que de la tomate, et que les 2 échantillons proviennent bien de la même espèce)

Toutes étapes ont l'air d'avoir été correctement exécutées et le contrôle qualité des données est bon. Il est donc possible de procéder aux étapes suivantes de l'analyse (DGE par exemple).

5. Ce n'est pas étonnant de trouver que la majorité des reads on un Phred score supérieur à 30, car il s'agit de données publiées, et donc déjà pré-triées selon ce facteur !

CHAPITRE 2

MORUE ATLANTIQUE (Q4)

Nous établirons la pipeline sur un sous-échantillon de 1000 lignes (plus rapide). Puis nous l'appliquerons à la totalité des reads (fichier .fasta entiers). Dans un premier temps, nous analyserons seulement 2 organes ('testis' et 'ovary', voir sec2.2.3). [1]

2.1 SOURCE DE L'ETUDE PRISE EN EXEMPLE

"Gene evolution and gene expression after whole genome duplication in fish : the PhyloFish database."
Pasquier J et al., BMC Genomics, 2016 May 18 ;17 :368

2.2 PREPARATION ENVIRONNEMENT DE TRAVAIL (Q4.1)

2.2.1 CONNEXION AU SERVEUR GENOLOGIN

voir sec.1.1.1

2.2.2 ORGANISATION DU TRAVAIL

— **Aller dans répertoire de travail 'work' :** cd work [2]

— **Créer les répertoires de travail :** dans
mkdir PROJET_NEXTFLOW/MORUE ;
cd PROJET_NEXTFLOW/MORUE ;
mkdir FASTQ ;
mkdir GENOME_REF ;

2.2.3 TÉLÉCHARGEMENT DES DONNÉES

En entrée, la pipeline a besoin des échantillons (format .fastq) et le génome de référence (annotations au format gtf, et la séquence du génome au format fasta/fna).

Fastq

Information sur l'expérience

— **Expérience :**

— **BioProject :** PRJNA256972 (référence NCBI/GEO) :
`https://www.ncbi.nlm.nih.gov/bioproject/PRJNA256972`

— **Echantillons :** voir tab.2.2.3

Information sur les données ARN

— **Extraction :** total RNAseq

— **LIBRARY PREPARATION :** STRANDED total RNA (*"TruSeq stranded total RNA SamplePrep Guide RevC")*

— **Matériel de séquençage :** Illumina HiSeq 2000

1. J'ai d'abord tenté de travailler avec l'organisme C.*elegans*, mais j'ai éprouvé des difficultés avec celui-ci (beaucoup trop de 0 dans les counts, ne permet pas de faire la fin du workflow, est bloqué à l'étape avec Desq2, car requière une étape de transformation en log2 (impossible avec 0). il faut ajouter des 'peuso-counts' ? ou problème avec le génome de référence plus vraissemblablement (j'ai essayé 3 set différents, même constat)
2. les calculs DOIVENT ETRE IMPERATIVEMENT être réalisés dans le répertoire 'work'.

Nom	SRA	Assession	QUAL
Cod testis	SRR2045424	SRX1044005	PAIRED
Cod ovary	SRR2045415	SRX1044005	PAIRED
Cod kidney	SRR2045421	SRX1044011	PAIRED
Cod intestine	SRR2045423	SRX1044013	PAIRED

TABLE 2.1 – Correspondance nom échantillons - numéro GEO-NCBI

— PAIRED_END

— un réplicat par condition

Téléchargement

— Telecharger à partir du local (pas possible à partir de wget)

— **Commande bash :**
scp *.gz laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/MORUE/FAS
pwd :

— **Vérification :** ls ou ls -lrt

— **Output :** OK
SRR2045415.fastq.gz SRR2045424.fastq.gz

Création de sous-échantillons de 1000 lignes [3]

— **Commande bash :** szcat SRR2045424.fastq.gz | head -n 1000 | gzip > SRR2045424_1000.fastq.gz
zcat SRR2045415.fastq.gz | head -n 1000 | gzip > SRR2045415_1000.fastq.gzz

— **Vérification :** ls -lrt

— **Output :** OK
SRR2045415.fastq.gz SRR2045415_1000.fastq.gz
SRR2045424.fastq.gz SRR2045424_1000.fastq.gz
Remarque : il manque les droits pour écriture (x).

— **Ajout des droits d'écritures :**

— **Commande bash :** chmod +x SRR2045424_1000.fastq.gz
chmod +x SRR2045415_1000.fastq.gz

— **Vérification :** ls -lrt
-rwxr-xr-x 1 laurier formation 2130399104 7 oct. 10 :02 SRR2045415.fastq.gz
-rwxr-xr-x 1 laurier formation 2008578343 7 oct. 10 :16 SRR2045424.fastq.gz
-rwxr-xr-x 1 laurier formation 12292 7 oct. 10 :28 SRR2045424_1000.fastq.gz
-rwxr-xr-x 1 laurier formation 12231 7 oct. 10 :28 SRR2045415_1000.fastq.gz

Génome de référence Nous utiliserons le génome de référence suivant : **ENSEMBLE**, version **gadMor3.0**.

— **Organisme :** Gadus.*morhua* (Atlantic cod)

— **Lien :** `https://ftp.ensembl.org/pub/release-110/`

3. Mettons au point la pipeline sur un petit échantillon, pour corriger les erreurs plus rapidement puis, une fois la pipeline sans erreurs, l'appliquer sur toutes les données

— **Téléchargement des données :** (commandes bach)

- fichier .gtf
  wget `https://ftp.ensembl.org/pub/release-110/gtf/gadus_morhua/`
  → Gadus_morhua.gadMor3.0.110.gtf.gz
- fichier .fa
  wget `https://ftp.ensembl.org/pub/release-110/fasta/gadus_morhua/dna/` →
  Gadus_morhua.gadMor3.0.dna.toplevel.fa.gz

— **Vérification :** ls

— **Output :** OK
Gadus_morhua.gadMor3.0.110.gtf.gz Gadus_morhua.gadMor3.0.dna.toplevel.fa.gz

— **Transfert sur le serveur :** [4] (commandes bach)
scp * laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/MORUE/GENO
pwd :

— unzip unzip GCF_902167405.1.zip

— transfert les fichiers importants (gtf et fasta) dans le répertoire 'GENOME_REF'

— **Vérification :** ls -lrt

— **Output :** OK
-rw——- 1 laurier formation 678362735 7 oct. 10 :35 GCF_902167405.1_gadMor3.0_genomic.fna
-rw——- 1 laurier formation 522152862 7 oct. 10 :35 genomic.gtf
drwxr-xr-x 3 laurier formation 4096 7 oct. 10 :36 archives

2.2.4 ORGANIGRAMME

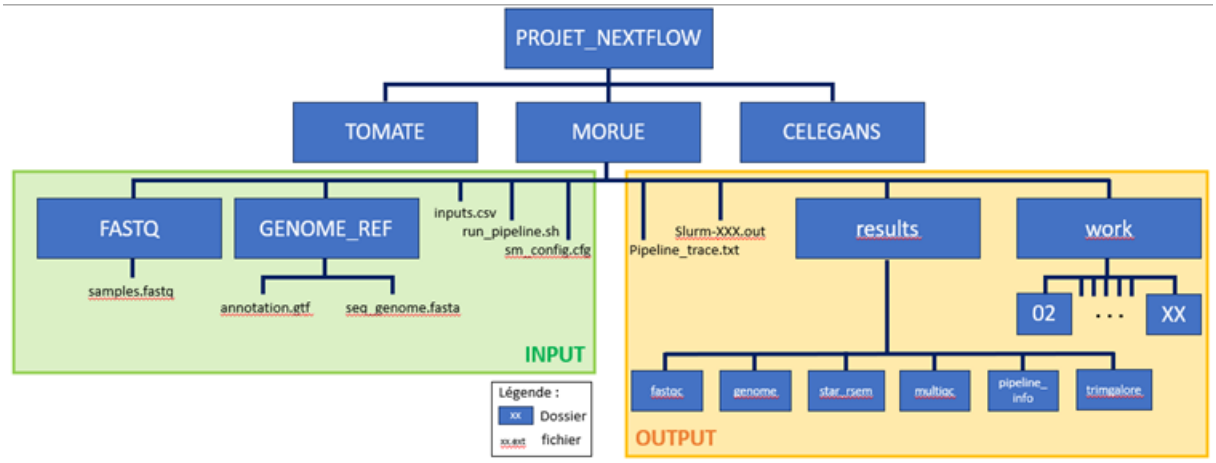


FIGURE 2.1 – Hiérarchie répertoires - Morue

4. possible aussi avec wget)

2.3 CREATION ET LANCEMENT DE LA PIPELINE

2.3.1 CHOIX DE LA PIPELINE

voir sec.1.2.1

2.3.2 PREPARATION DE LA PIPELINE NEXTFLOW

Nous travaillons dans le répertoire 'MORUE'.

Fichier sm_config.cfg voir sec.1.2.2

Fichier inputs.csv

Rôle Le but est de définir le plan d'expérience indiqué sur le site du bioprojet NCBI/GEO.
Information sur les échantillons : voir sec.2.2.3

Création

- **Création du fichier :** touch inputs.csv
- **Vérifier en amont les chemins absolus :**
  ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/FASTQ/SRR2045424.fastq.gz
  ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/FASTQ/SRR2045415.fastq.gz
  ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/FASTQ/SRR2045423.fastq.gz
  ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/FASTQ/SRR2045421.fastq.gz
- **Ecriture :** nano inputs.csv
  group,replicate,fastq_1,fastq_2,strandedness [5]
  testis,1,/home/laurier/work/PROJET_NEXTFLOW/MORUE/FASTQ/SRR2045424.fastq.gz
  „unstranded
  ovary,1,/home/laurier/work/PROJET_NEXTFLOW/MORUE/FASTQ/SRR2045415.fastq.gz,
  ,unstranded
- **Vérification :** more inputs.csv
- **Output :** OK, voir fig.2.2

```
laurier@genologin2 ~/work/PROJET_NEXTFLOW/CELEGANS $ more inputs.csv
sample,replicate,fastq_1,fastq_2,strandedness
wild,1,/home/laurier/work/PROJET_NEXTFLOW/CELEGANS/FASTQ/SRR9602643_1000.fastq.gz,,auto
wild,2,/home/laurier/work/PROJET_NEXTFLOW/CELEGANS/FASTQ/SRR9602645_1000.fastq.gz,,auto
mutant,1,/home/laurier/work/PROJET_NEXTFLOW/CELEGANS/FASTQ/SRR9602649_1000fastq.gz,,auto
mutant,2,/home/laurier/work/PROJET_NEXTFLOW/CELEGANS/FASTQ/SRR9602653_1000.fastq.gz,,auto
```

FIGURE 2.2 – Script fichier shcéma expérience (inputs.csv) - Morue

Fichier run_pipeline.sh

5. Les options possibles pour 'strandedness' sont :'forward','reverse' et 'unstranded'. J'ai fait un premier essai avec 'forward', étant donné que les échantillons sont indiqués 'stranded'. Mais le MultiQC indiquait 50% sens et 50% antisens environ. J'ai décidé de refaire tourner avec 'unstranded', et laisser fq, Salmon détecter automatiquement le sens du brin.

Ressources voir sec.1.2.2

Rôle voir sec.1.2.2

Création du fichier

— **Création du fichier :** touch run_pipeline.sh

— **Vérifier en amont les chemins absolus :**
ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/inputs.csv
ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/GENOME_REF/Gadus_morhua.gadMor3.0.11
ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/GENOME_REF/Gadus_morhua.gadMor3.0.dn
ls /home/laurier/work/PROJET_NEXTFLOW/MORUE/sm_config.cfg

— **Ecriture :** nano run_pipeline.sh voir fig.2.3

— **Vérification :** more run_pipeline.sh

— **Output :** OK

```
#!/bin/bash
#SBATCH -J JulieCAMPOS
#SBATCH -p workq
#SBATCH --mem=6G
#SBATCH --time=24:00:00
#SBATCH --cpus-per-task=48

module purge
module load bioinfo/nfcore-Nextflow-v21.04.1
input=/home/laurier/work/PROJET_NEXTFLOW/MORUE/inputs.csv
gtf=/home/laurier/work/PROJET_NEXTFLOW/MORUE/GENOME_REF/Gadus_morhua.gadMor3.0.110.gtf
fasta=/home/laurier/work/PROJET_NEXTFLOW/MORUE/GENOME_REF/
Gadus_morhua.gadMor3.0.dna.toplevel.fa
config=/home/laurier/work/PROJET_NEXTFLOW/MORUE/sm_config.cfg
nextflow run nf-core/rnaseq \
-r 3.0 \
-profile genotoul \
--input $input \
--fasta $fasta \
--gtf $gtf \
--aligner star_rsem \
-c $config \
-resume
```

FIGURE 2.3 – Script fichier .sh - Morue

2.3.3 RUN/LANCEMENT DE LA PIPELINE

Nous inclurons dans le rapport uniquement les résultats de la pipeline effectué sur les échantillons entier.

Run

— **Commande bash :** sbatch run_pipeline.sh

— **Output :** Submitted batch job 50755492

Vérification

— **Commande bash :** seff 50755492

— **Output :** voir fig.2.4

```
Job ID: 50755492
Cluster: genobull
User/Group: laurier/formation
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 48
CPU Utilized: 00:02:57
CPU Efficiency: 0.06% of 3-11:21:36 core-walltime
Job Wall-clock time: 01:44:12
Memory Utilized: 1.81 GB
Memory Efficiency: 30.25% of 6.00 GB
```

FIGURE 2.4 – Résultats statut du job 50755492 - Morue

Explication de la sortie du seff 1.2.3

Log 1.2.3 **Commande bash :** tail -n 200 slurm-50755492.out

```
[4b/7d4e18] process > RNASEQ:RSEQC:RSEQC_READDIST... [100%] 2 of 2 ✔
[8b/f8ba25] process > RNASEQ:RSEQC:RSEQC_READDUPL... [100%] 2 of 2 ✔
[90/38c6d4] process > RNASEQ:MULTIQC_CUSTOM_STRAN... [100%] 1 of 1 ✔
[db/857251] process > RNASEQ:GET_SOFTWARE_VERSIONS   [100%] 1 of 1 ✔
[84/fe9ad7] process > RNASEQ:MULTIQC (1)             [100%] 1 of 1 ✔
-[nf-core/rnaseq] 2/2 samples passed STAR 5% mapped threshold:
    78.64%: testis_R1
    84.71%: ovary_R1
-
-[nf-core/rnaseq] Pipeline completed successfully-
Completed at: 07-oct.-2023 18:28:53
Duration    : 1h 44m 5s
CPU hours   : 30.6
Succeeded   : 71
```

FIGURE 2.5 – Résultats log du job 50755492 (fichier slurm) - Morue

2.4 RESULTATS ET INTERPRÉTATION

2.4.1 RECUPERATION DES RESULTATS EN LOCAL

— **Commande bash :**
scp laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/MORUE/pipeline_trace.txt .
scp laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/MORUE/slurm-50755492.out .
scp laurier@genologin.toulouse.inrae.fr :/home/laurier/work/PROJET_NEXTFLOW/MORUE/results/multiqc/star_rsem/multiqc_report.html .
pwd :

— **Output :** OK

2.4.2 INTERPRETATION - CONTROLE QUALITE AVEC MULTIQC

voir annexe B (chapitre.5)

Présentation voir section.1.3.3

Interprétation Voici quelques résultats résumés sur MultiQC intéressants à considérer dans nos échantillons (*'ovary'*/ovaires et *'testis'*/testicules) :

— Bien que proches,nous constatons quelques différences mineurs de % d'alignement entre nos 2 échantillons (environ 47M de reads pour ovaire et 38M de reads pour testicule)

— est indiqué aussi la vérification des sens des reads (forward ou reverse), selon indiqué dans notre schéma expérimental (*inputs.csv*) [6]

— la majorité des reads sont du biotype *'protein coding'*. Un autre biotype est suffisamment abondant pour être visible sur le barplot (bien qu'extrêmement minoritaire : lncRNA ou miscRNA ?

— nous remarquons que plus de 55% des reads sont des duplicats ! Ce qui est beaucoup ! (Normalement toléré : moins de 20%)

— nos 2 échantillons sont relativement bien répartis entre exonx/introns et régions intergenique.

— les résultats des fastqc montrent une bonne qualité des reads, toute égale à un PHRED score de 30 (données publiées)

— le % de GC est différent de celui rencontré sur les échantillons de tomates (45% pour les tomates contre environ 55% chez la morue Atlantique). Les 2 échantillons sont assez proche, mais suffisament éloignée pour que MultiQC nous indique par une couleur orange, un **WARNING**

Le nombre élévé de duplicats m'interpelle, et avant de poursuivre plus en avant l'analyse de ces échantillons, je souhaiterais approfondir cet aspect, et vérifier plus en détails pourquoi.

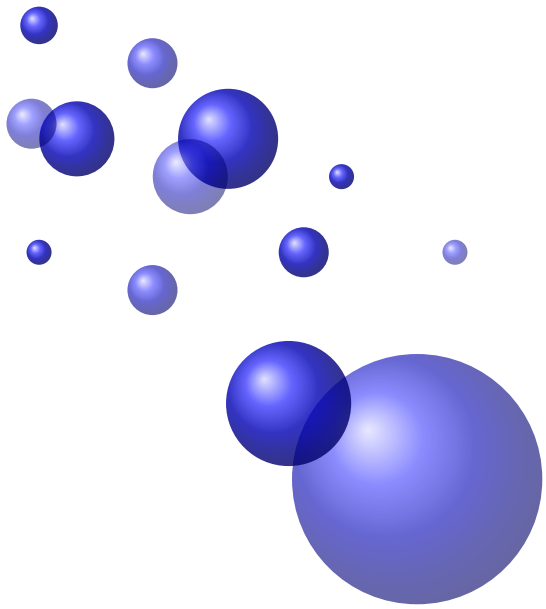
6. J'ai tenté 2 approches, une avec option *'forward'*, et une avec *'unstranded'*. Le rapport fournit a probablement été réalisé avec *'forward'*, car un **WARNING** signale que si j'ai renseigné 'forward', il s'avère que 49% des reads sont en fait 49% 'sense' et 47.5% 'antisens'. Le reste est indéterminé. Donc, l'option 'unstranded' aurait été plus adapté à notre cas.

CHAPITRE 3

CONCLUSION

L'immense avantage de nextflow, couplé avec 'nf score' permet une très haute reproductibilité, un incroyable gain de temps tout en restant simple et rapide (en comparaison avec une pipeline élaborée manuellement).

Petite remarque personnelle sur LateX :
Je découvre LateX. J'ai été traumatisée des rapports sur 'WORD' et m'étais jurée de ne jamais plus recommencer. Plusieurs personnes m'avaient déjà vantés ses biens faits. J'ai été très heureuse d'avoir trouvé l'occasion de prendre le temps de découvrir et pour la première fois de ma vie peut être, ça a été un réel plaisir d'écrire un rapport !
J'adhère complètement ! les possibilités sont incroyables et créer aussi facilement un rapport vraiment propre et beau est un régal. Petit bémol sur le non contrôle de la position des images. D'un côté, je trouve dommage de ne pas pouvoir gérer cet aspect, et d'un autre côté...extrêmement reposant de ne pas avoir à devoir le faire !
Donc, merci :)



CHAPITRE 4

ANNEXE A - MULTIQC - TOMATE

MultiQC

A modular tool to aggregate results from bioinformatics analyses across many samples into a single report.

This report has been generated by the nf-core/rnaseq analysis pipeline. For information about how to interpret these results, please see the documentation.

Report generated on 2023-10-06, 14:18 based on data in: `/work/laurier/PROJET_NEXTFLOW/TOMATES/work/bc/be579a745e5eaee703e257e5d1b518`

Welcome! Not sure where to start? Watch a tutorial video *(6:06)* don't show again ×

General Statistics

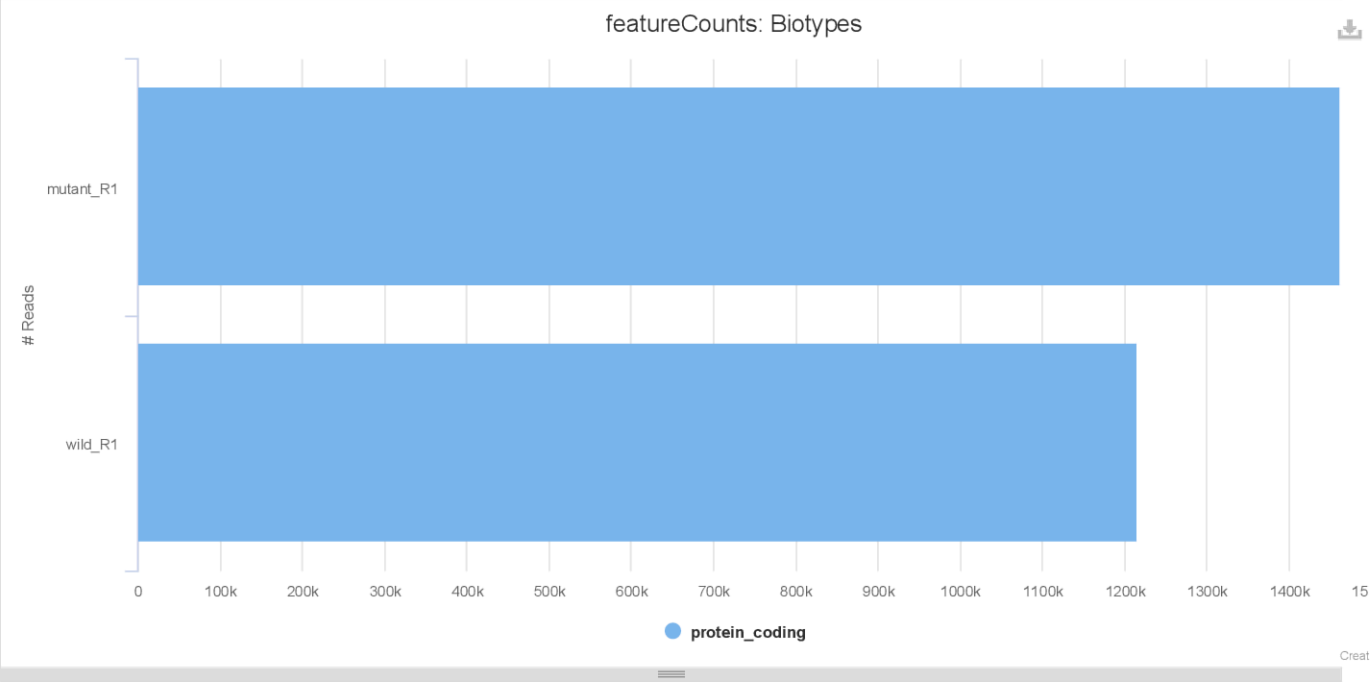
Copy table | Configure Columns | Plot | Showing 6/6 rows and 21/27 columns.

Sample Name	M Reads Mapped	% rRNA	dupInt	% Dups	5'-3' bias	M Aligned	% Alignable	% Proper Pairs	Error rate	M Non-P
mutant_R1	3.3	0.00%	0.00%	17.3%	1.43	1.6	99.2%	78.3%	0.16%	0.1
mutant_R1_1										
mutant_R1_2										
wild_R1	2.7	0.00%	0.00%	18.3%	1.43	1.3	99.3%	76.9%	0.16%	0.1
wild_R1_1										
wild_R1_2										

Biotype Counts

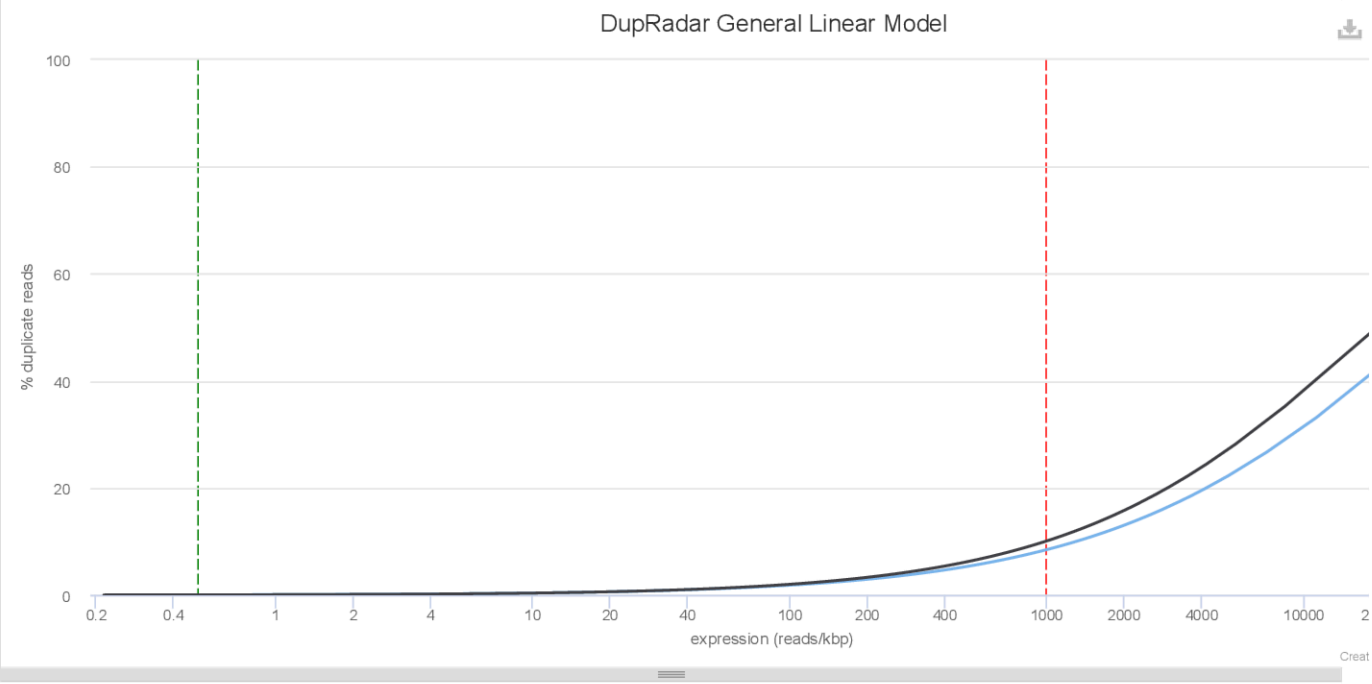
shows reads overlapping genomic features of different biotypes, counted by featureCounts.

Number of Reads | Percentages



DupRadar

provides duplication rate quality control for RNA-Seq datasets. Highly expressed genes can be expected to have a lot of duplicate reads, but high numbers of duplicates at low read counts can indicate low library complexity with technical duplication. This plot shows the general linear models - a summary of the gene duplication distributions.



Picard

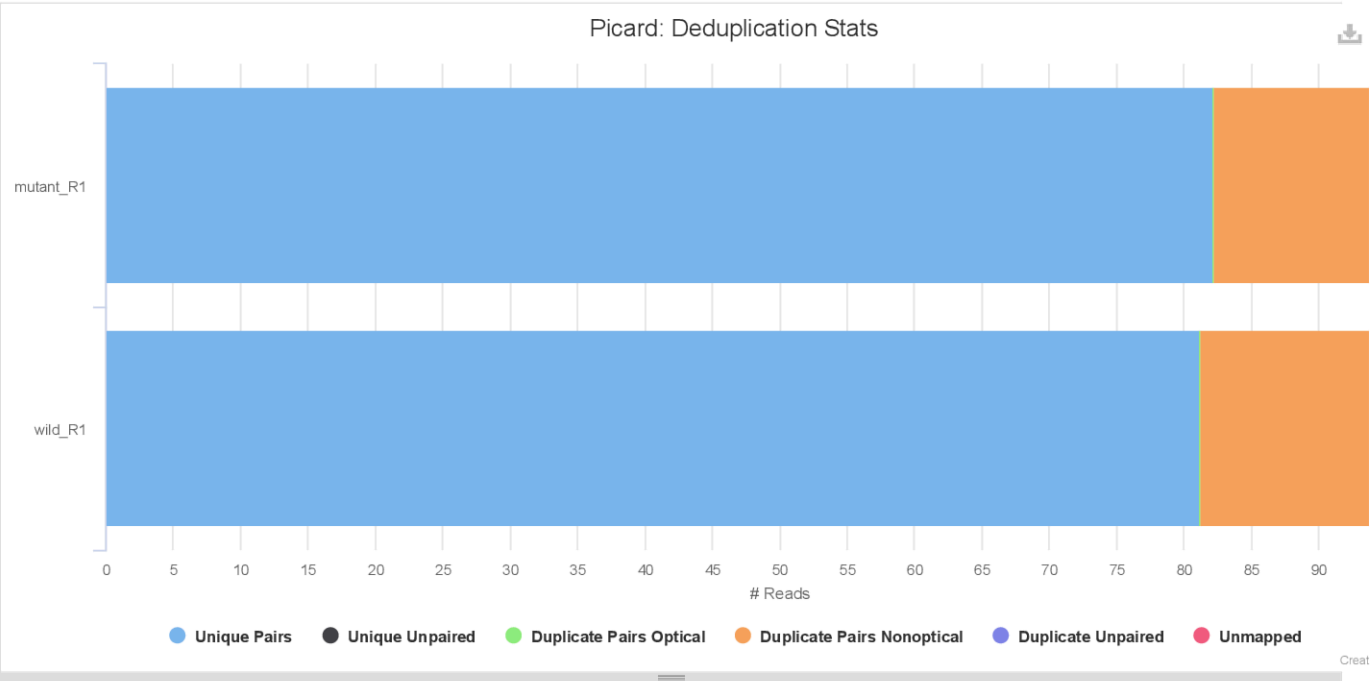
Picard is a set of Java command line tools for manipulating high-throughput sequencing data.

Mark Duplicates

Help

Number of reads, categorised by duplication state. **Pair counts are doubled** - see help text for details.

Number of Reads | Percentages

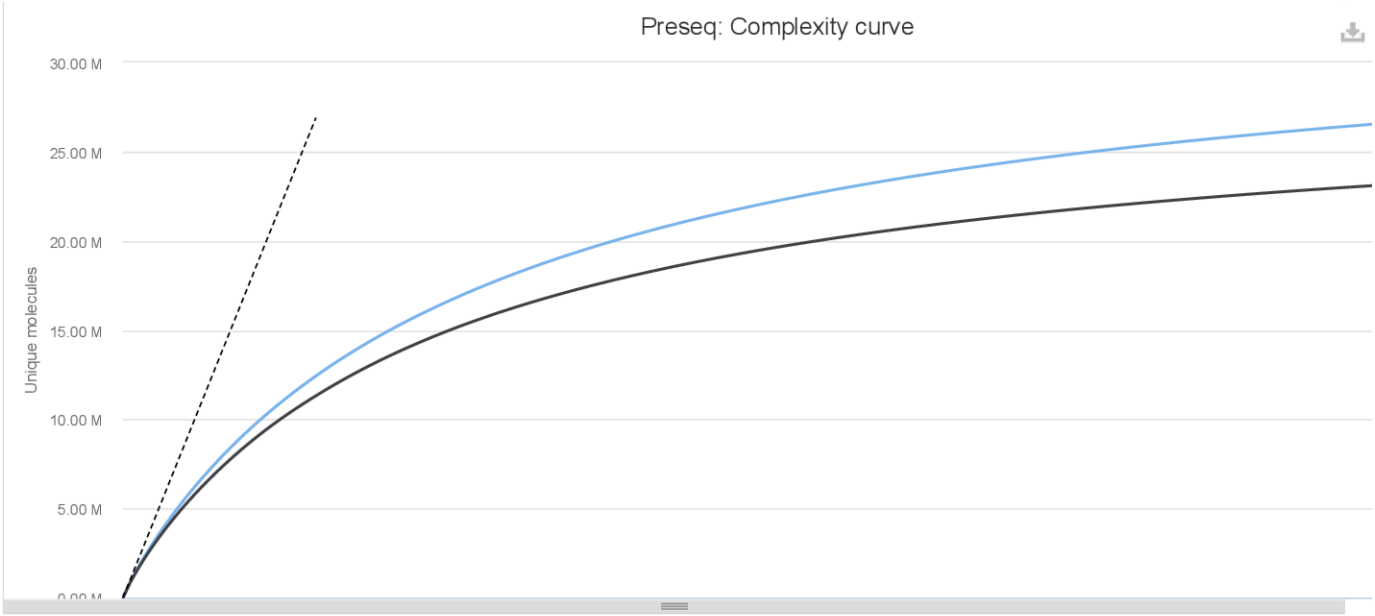


Preseq

Preseq estimates the complexity of a library, showing how many additional unique reads are sequenced for increasing total read count. A shallow curve indicates complexity saturation. The dashed line shows a perfectly complex library where total reads = unique reads.

Complexity curve

Note that the x axis is trimmed at the point where all the datasets show 80% of their maximum y-value, to avoid ridiculous scales.



QualiMap

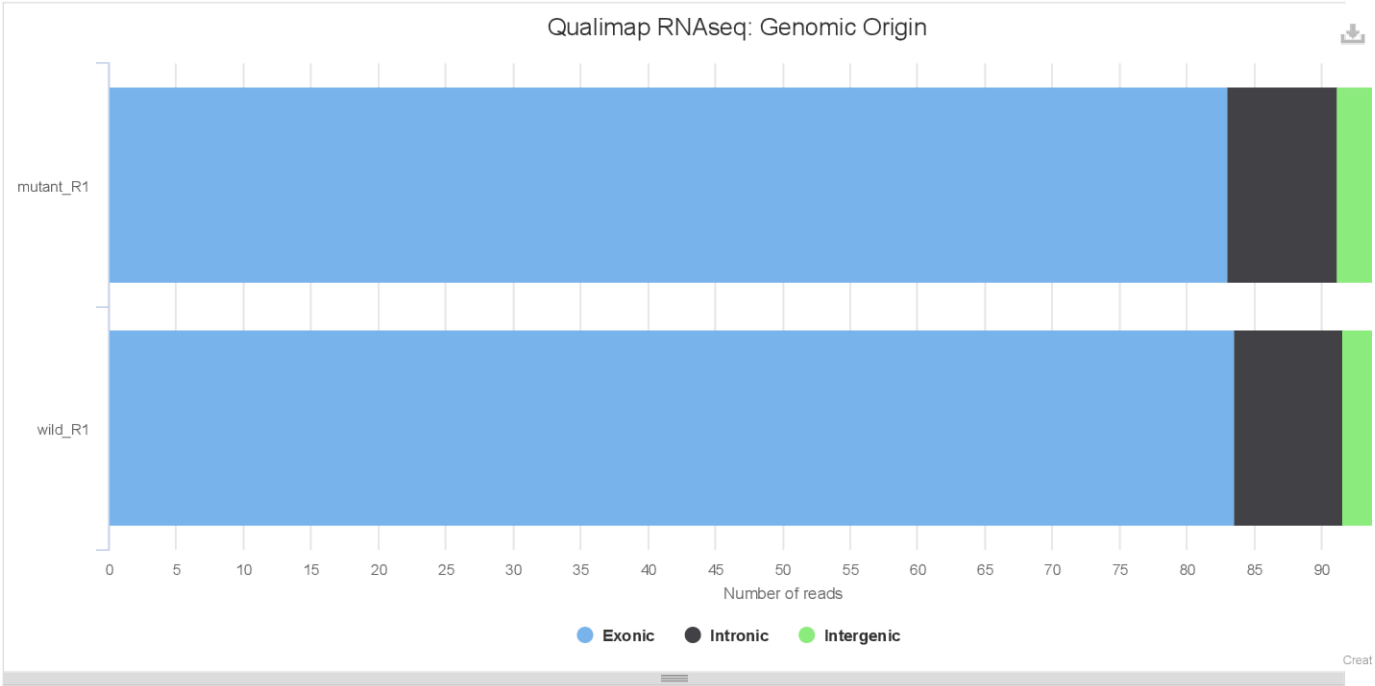
QualiMap is a platform-independent application to facilitate the quality control of alignment sequencing data and its derivatives like feature counts.

Genomic origin of reads

Help

Classification of mapped reads as originating in exonic, intronic or intergenic regions. These can be displayed as either the number or percentage of mapped reads.

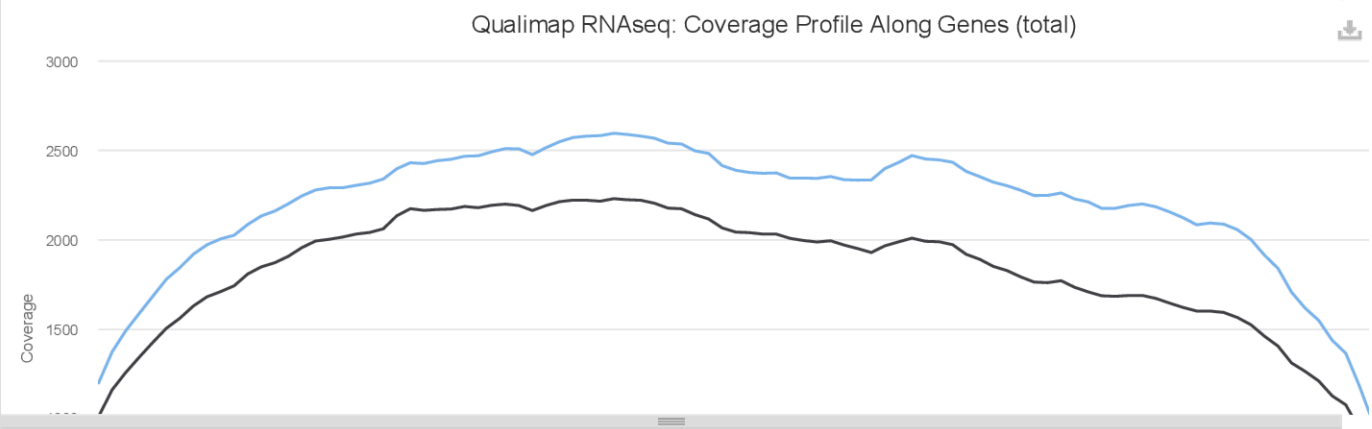
Counts | Percentages



Gene Coverage Profile

Help

Mean distribution of coverage depth across the length of all mapped transcripts.



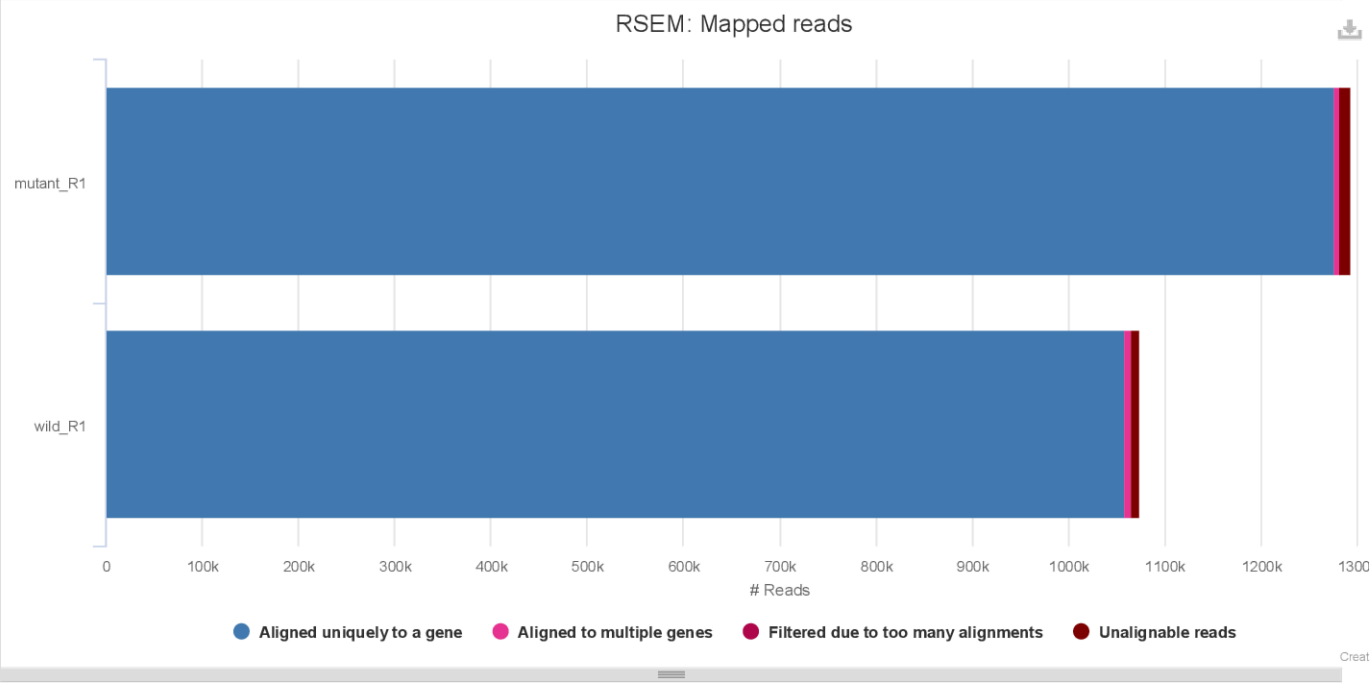
Rsem

Rsem RSEM (RNA-Seq by Expectation-Maximization) is a software package forestimating gene and isoform expression levels from RNA-Seq data.

Mapped Reads

A breakdown of how all reads were aligned for each sample.

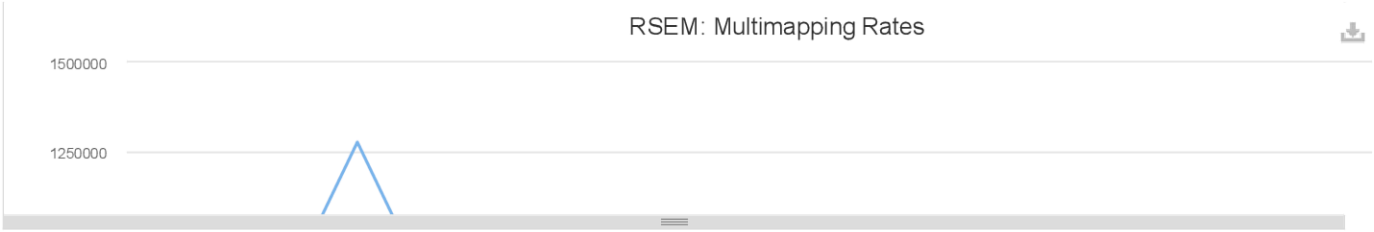
Number of Reads | Percentages



Multimapping rates

Help

A frequency histogram showing how many reads were aligned to n reference regions.



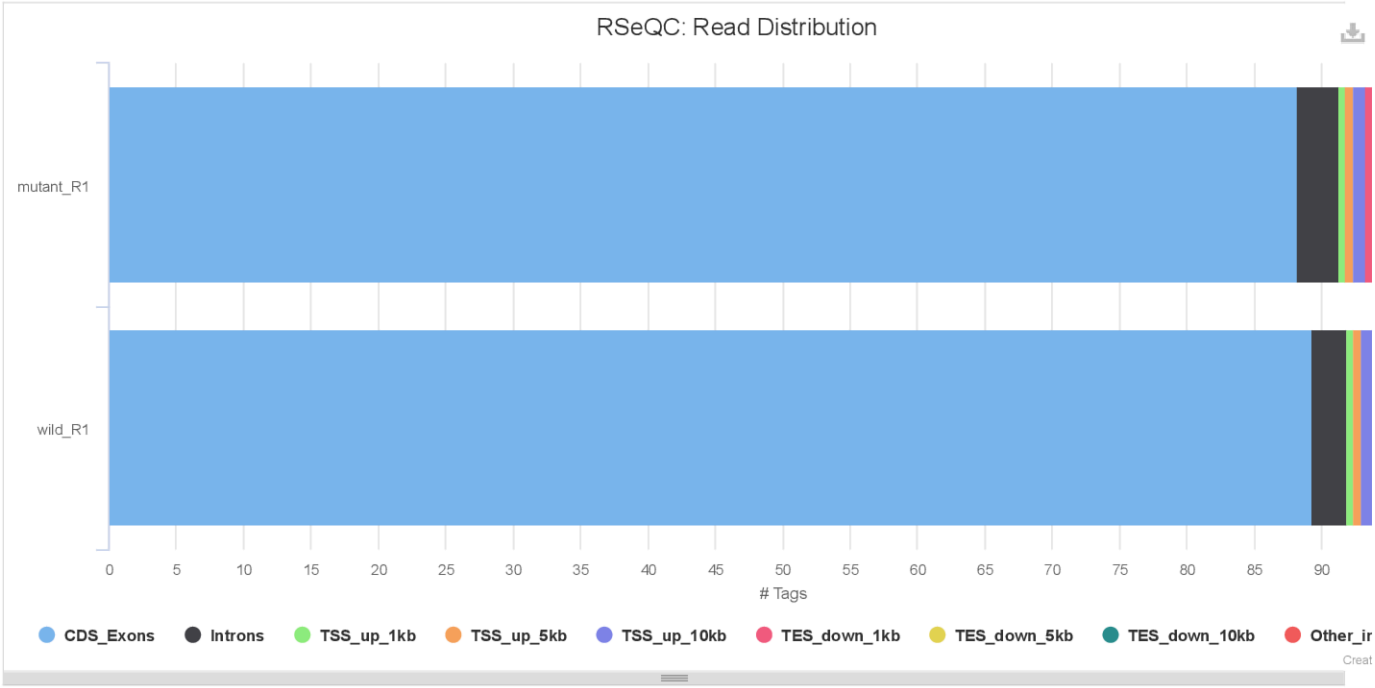
RSeQC

RSeQC package provides a number of useful modules that can comprehensively evaluate high throughput RNA-seq data.

Read Distribution

Read Distribution calculates how mapped reads are distributed over genome features.

Number of Tags | Percentages



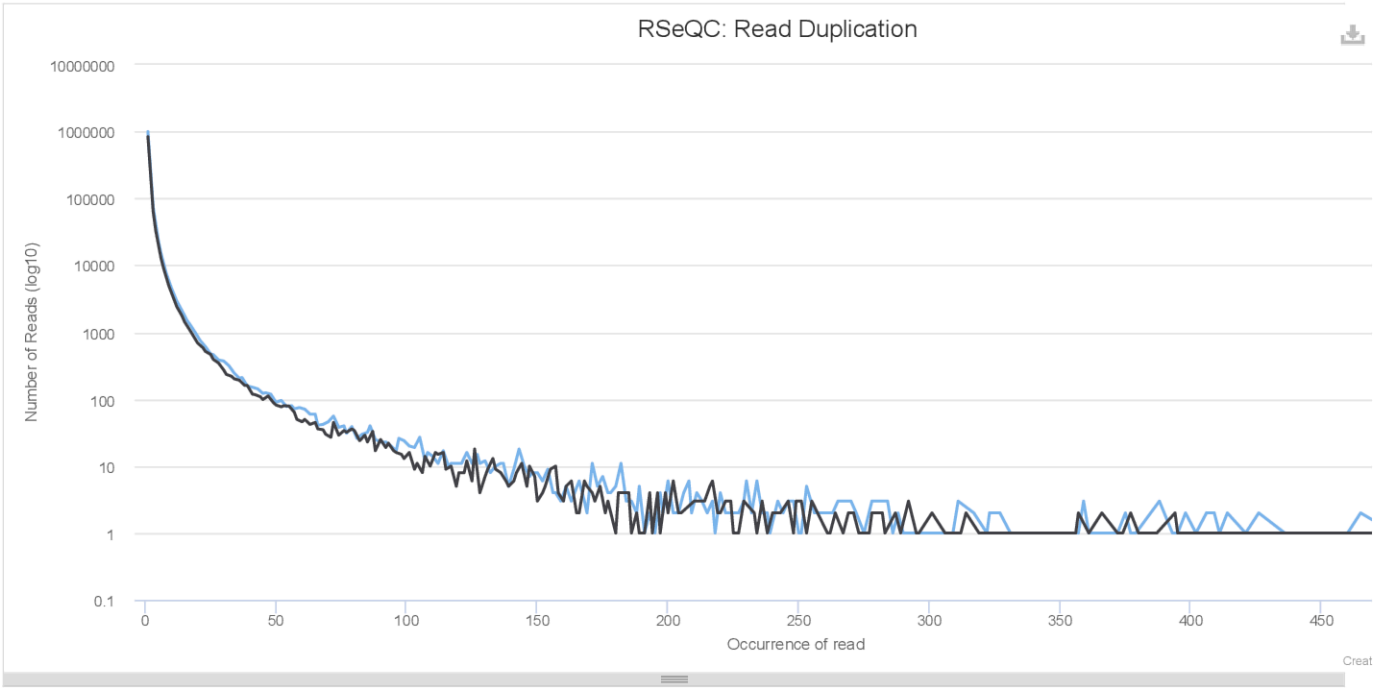
Inner Distance

Inner Distance calculates the inner distance (or insert size) between two paired RNA reads. Note that this can be negative if fragments overlap.

Counts | Percentages

Read Duplication

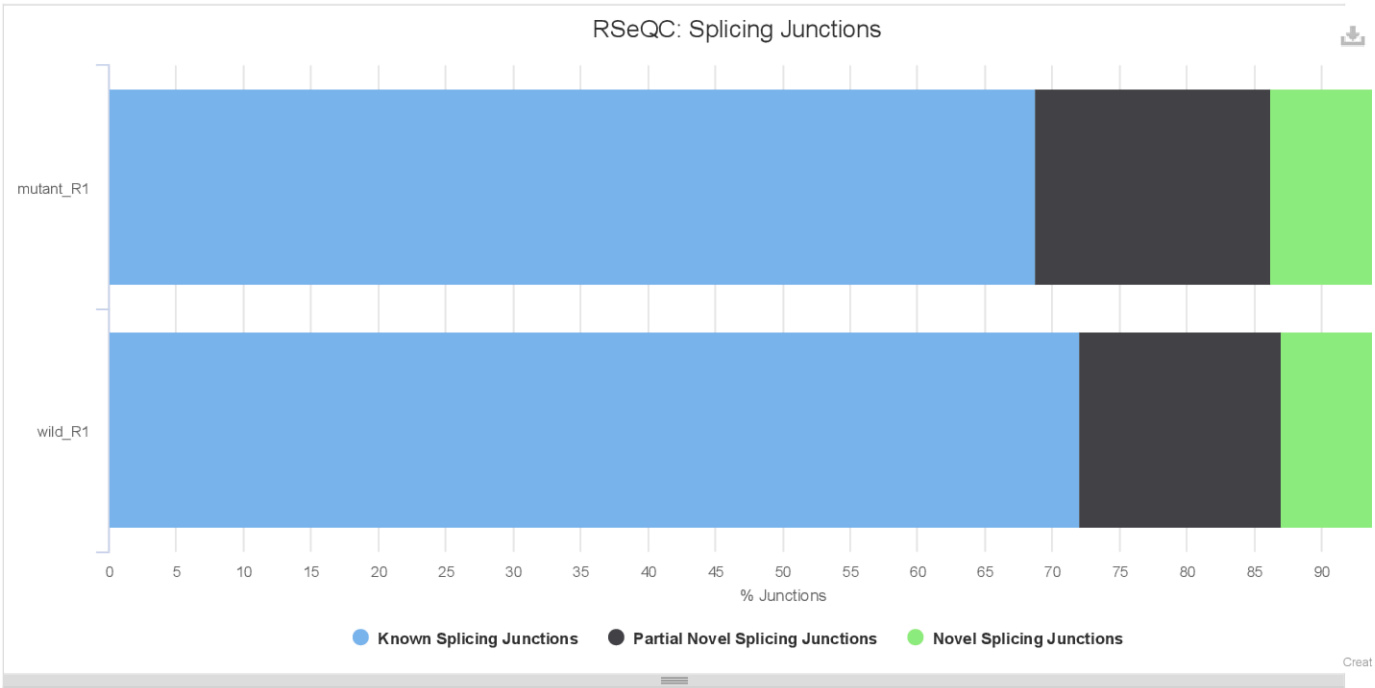
read_duplication.py calculates how many alignment positions have a certain number of exact duplicates. Note - plot truncated at 500 occurrences and binned.



Junction Annotation

Junction annotation compares detected splice junctions to a reference gene model. An RNA read can be spliced 2 or more times, each time is called a splicing event.

Counts | Percentages | Junctions | Events

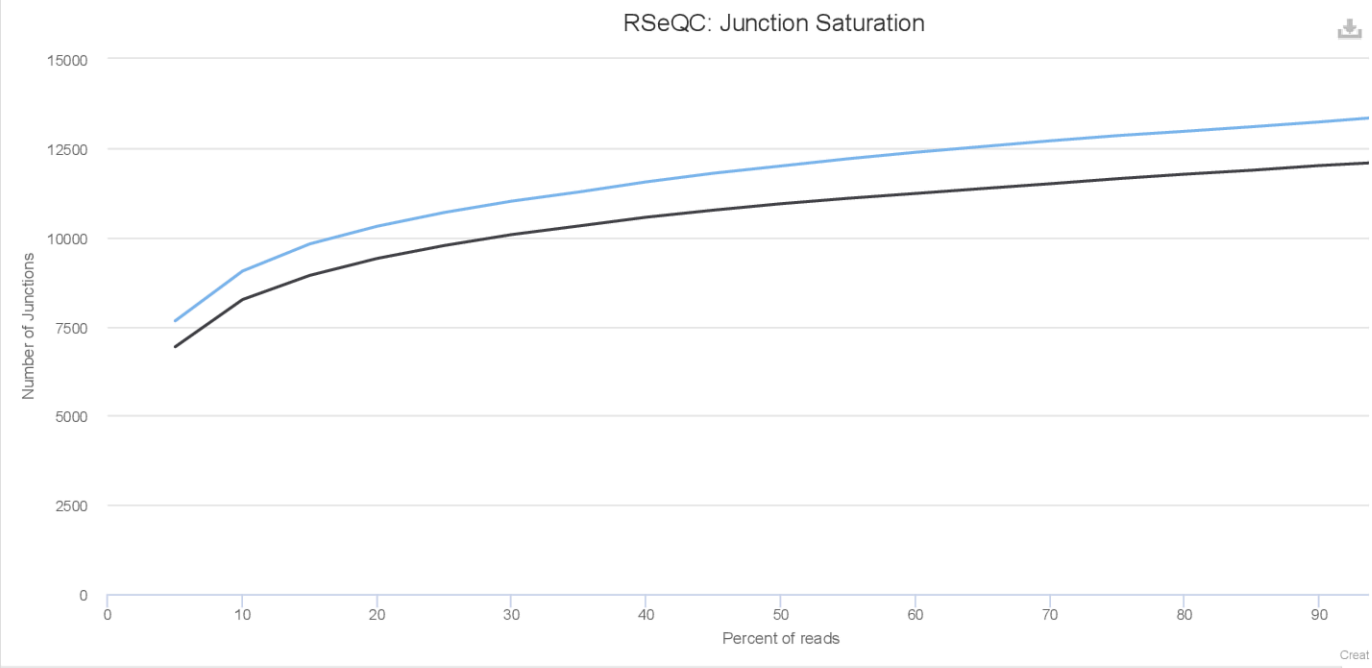


Junction Saturation

Junction Saturation counts the number of known splicing junctions that are observed in each dataset. If sequencing depth is sufficient, all (annotated) splice junctions should be rediscovered, resulting in a curve that reaches a plateau. Missing low abundance splice junctions can affect downstream analysis.

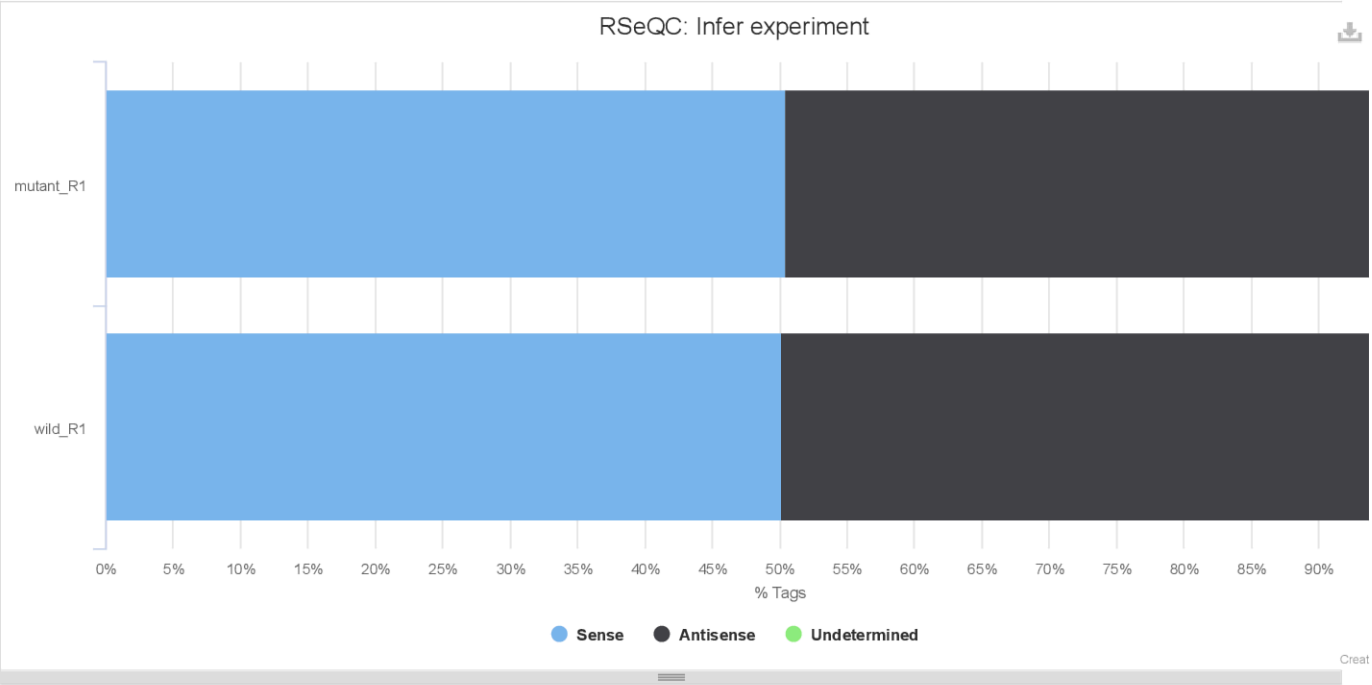
Click a line to see the data side by side (as in the original RSeQC plot).

All Junctions | Known Junctions | Novel Junctions



Infer experiment

Infer experiment counts the percentage of reads and read pairs that match the strandedness of overlapping transcripts. It can be used to infer whether RNA-seq library preps are stranded (sense or antisense).

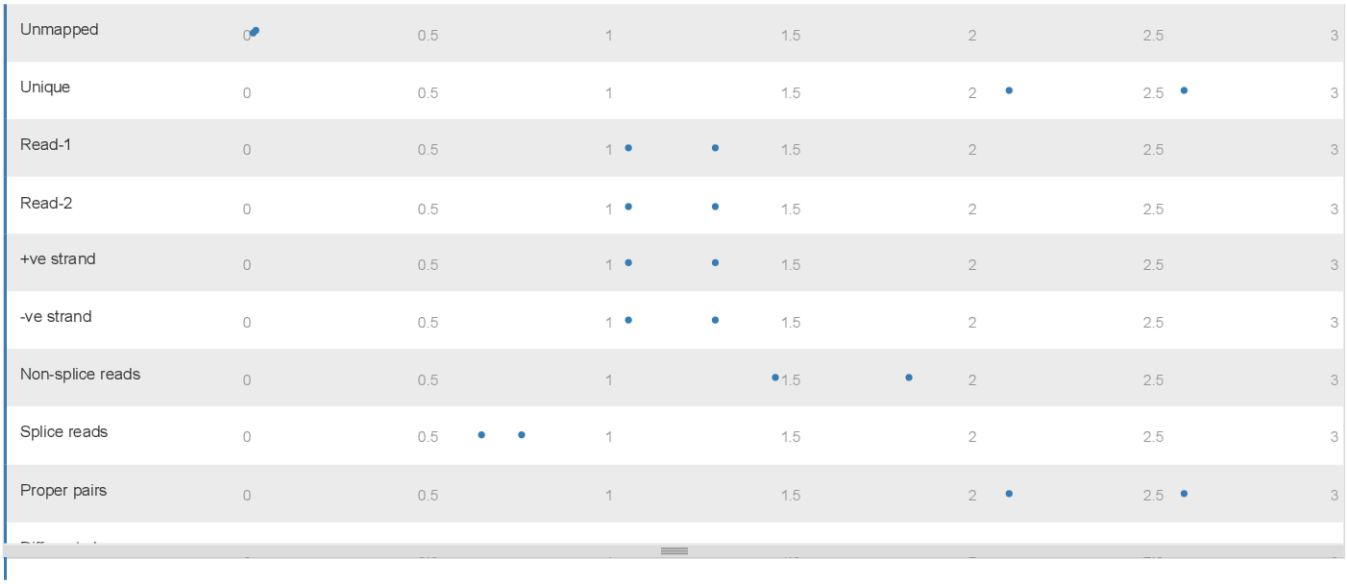


Bam Stat

All numbers reported in millions.

Hover over a data point for more information

Total records	0	0.5	1	1.5	2	2.5	3
QC failed	0	0.5	1	1.5	2	2.5	3
Duplicates	0	0.5	1	1.5	2	2.5	3
Non primary hit	0	0.5	1	1.5	2	2.5	3



Samtools

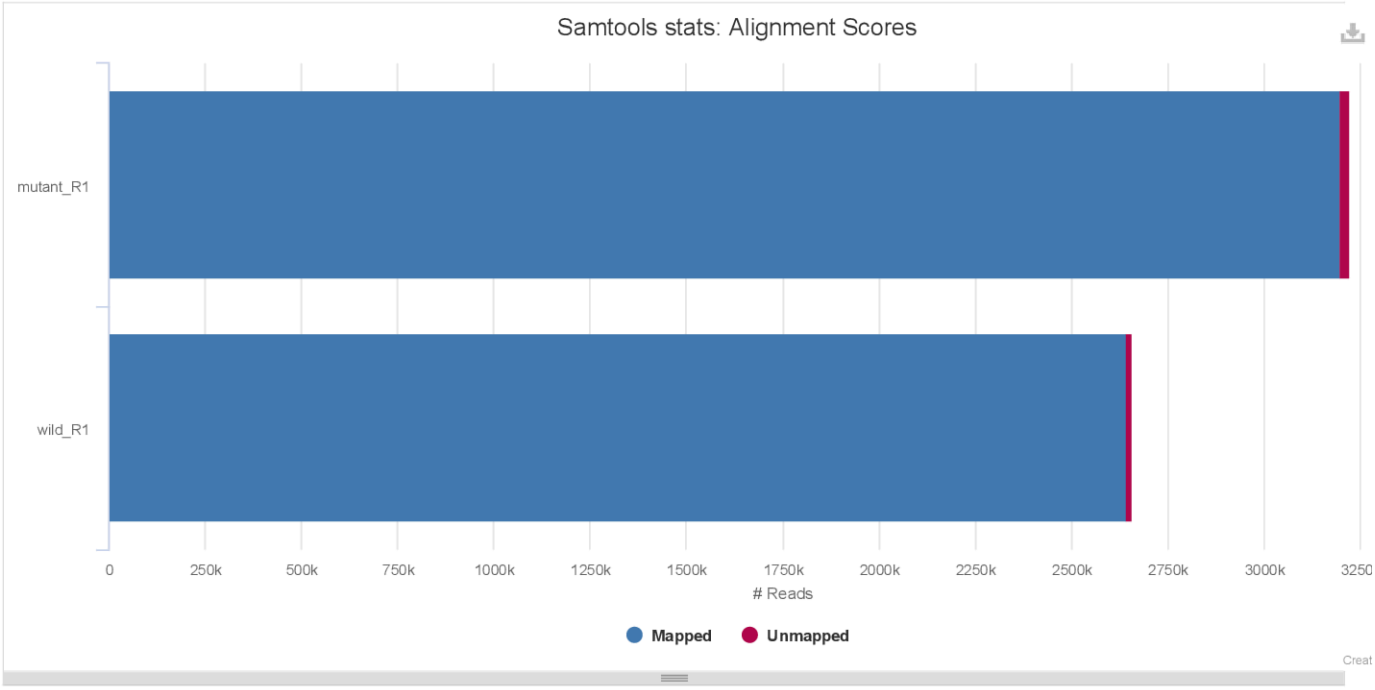
Samtools is a suite of programs for interacting with high-throughput sequencing data.

Percent Mapped

Help

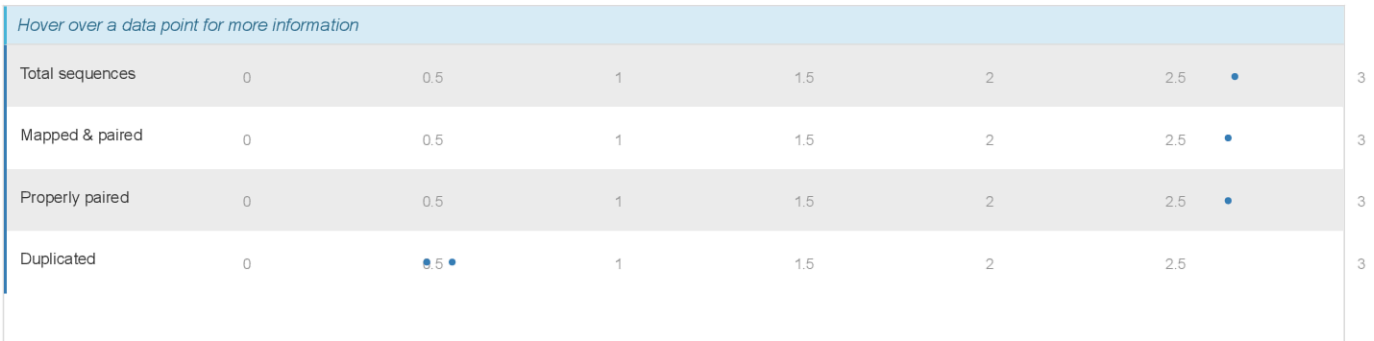
Alignment metrics from `samtools stats`; mapped vs. unmapped reads.

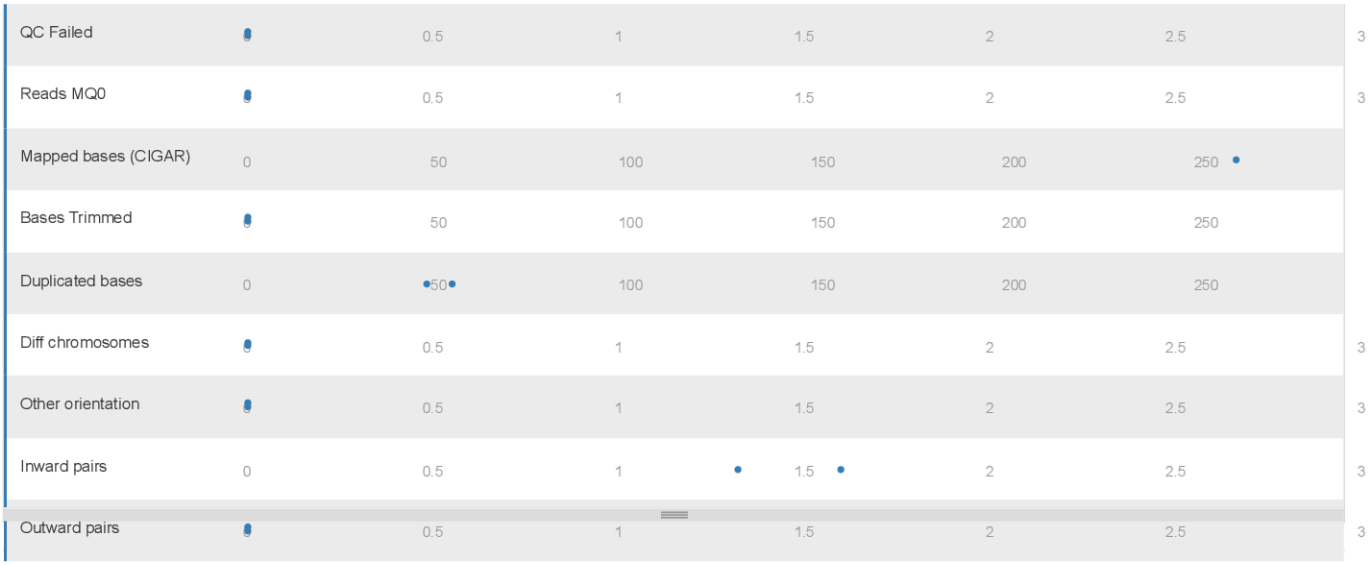
Number of Reads | Percentages



Alignment metrics

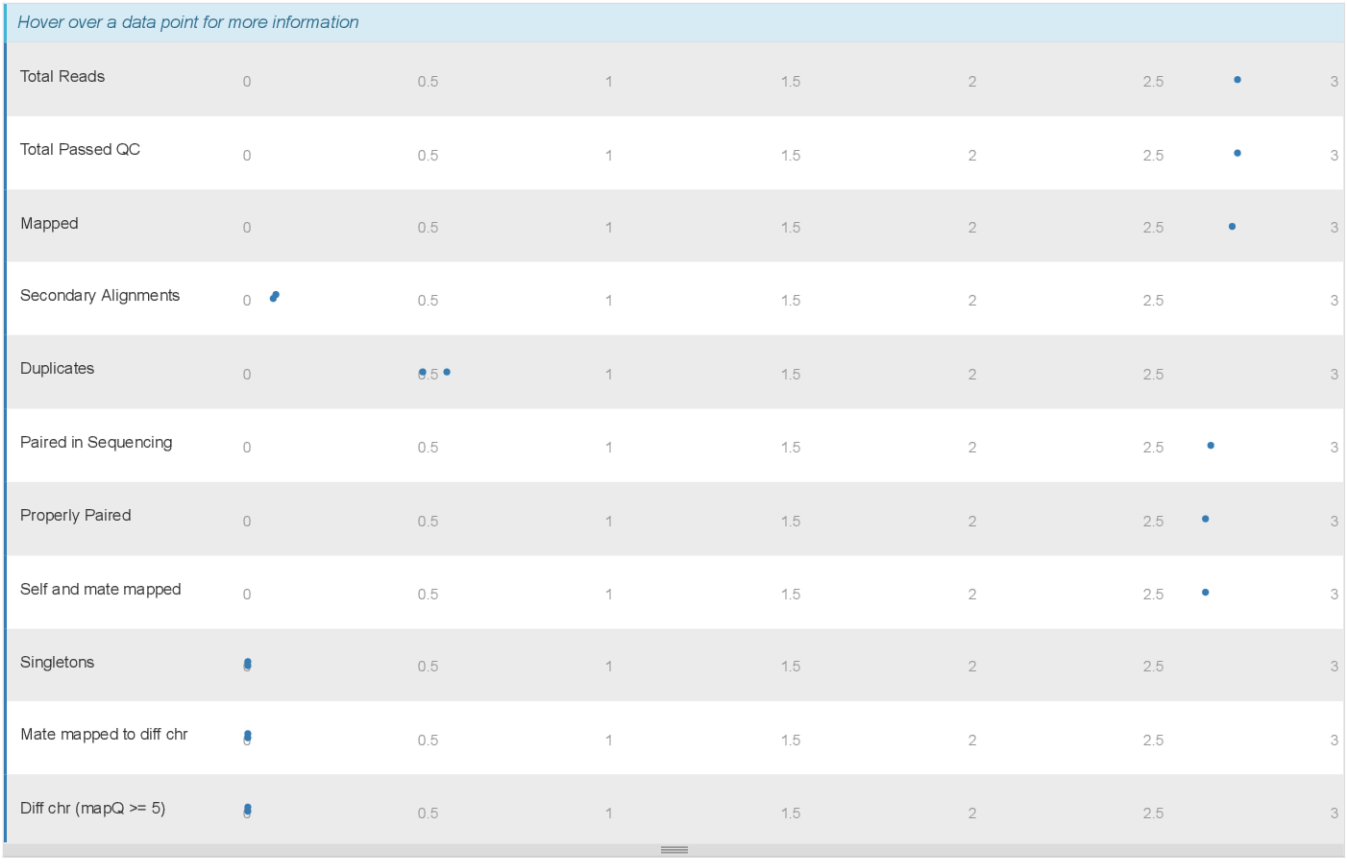
This module parses the output from `samtools stats`. All numbers in millions.





Samtools Flagstat

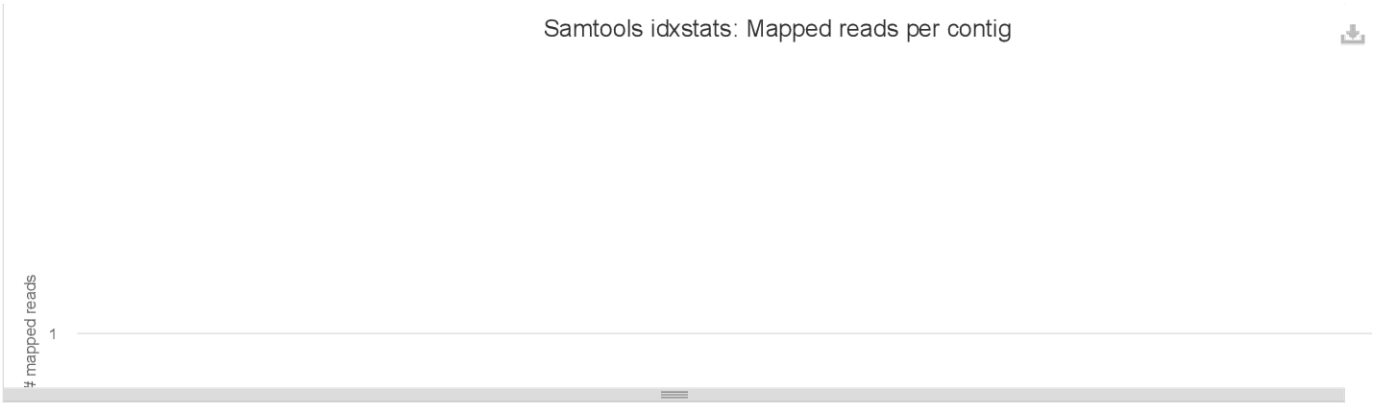
This module parses the output from `samtools flagstat`. All numbers in millions.



Mapped reads per contig

The `samtools idxstats` tool counts the number of mapped reads per chromosome / contig. Chromosomes with < 0.1% of the total aligned reads are omitted from this plot.

Counts | Log10 | Normalised Counts | Observed over Expected Counts | Raw Counts



FastQC (raw)

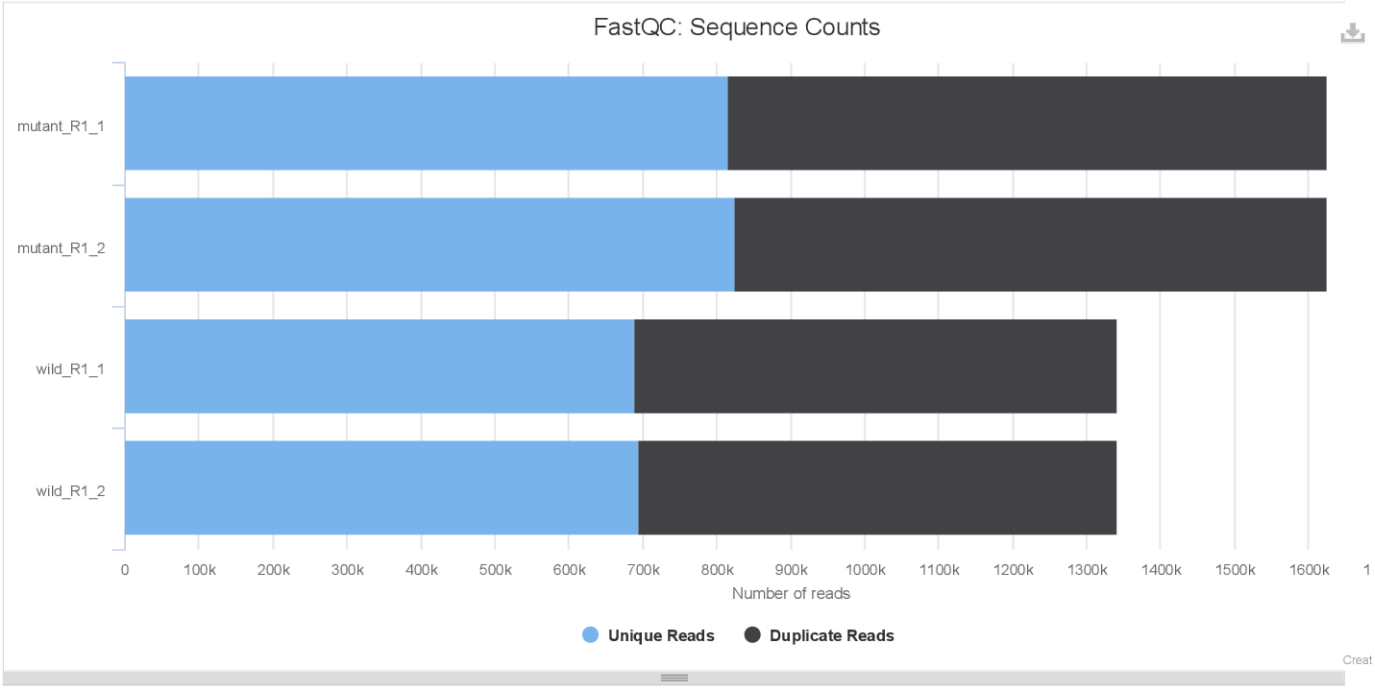
FastQC (raw) This section of the report shows FastQC results before adapter trimming.

Sequence Counts

Help

Sequence counts for each sample. Duplicate read counts are an estimate only.

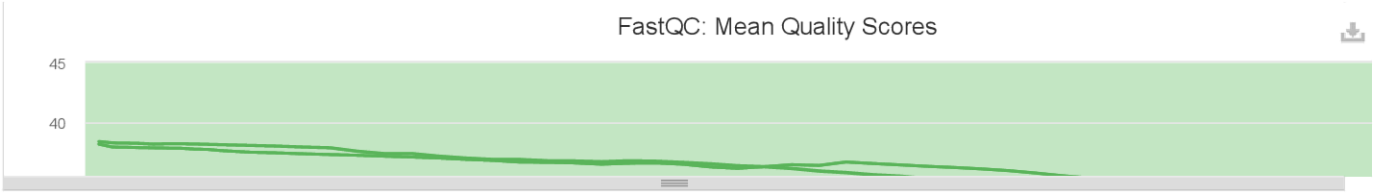
Number of reads | Percentages



Sequence Quality Histograms 4

Help

The mean quality value across each base position in the read.

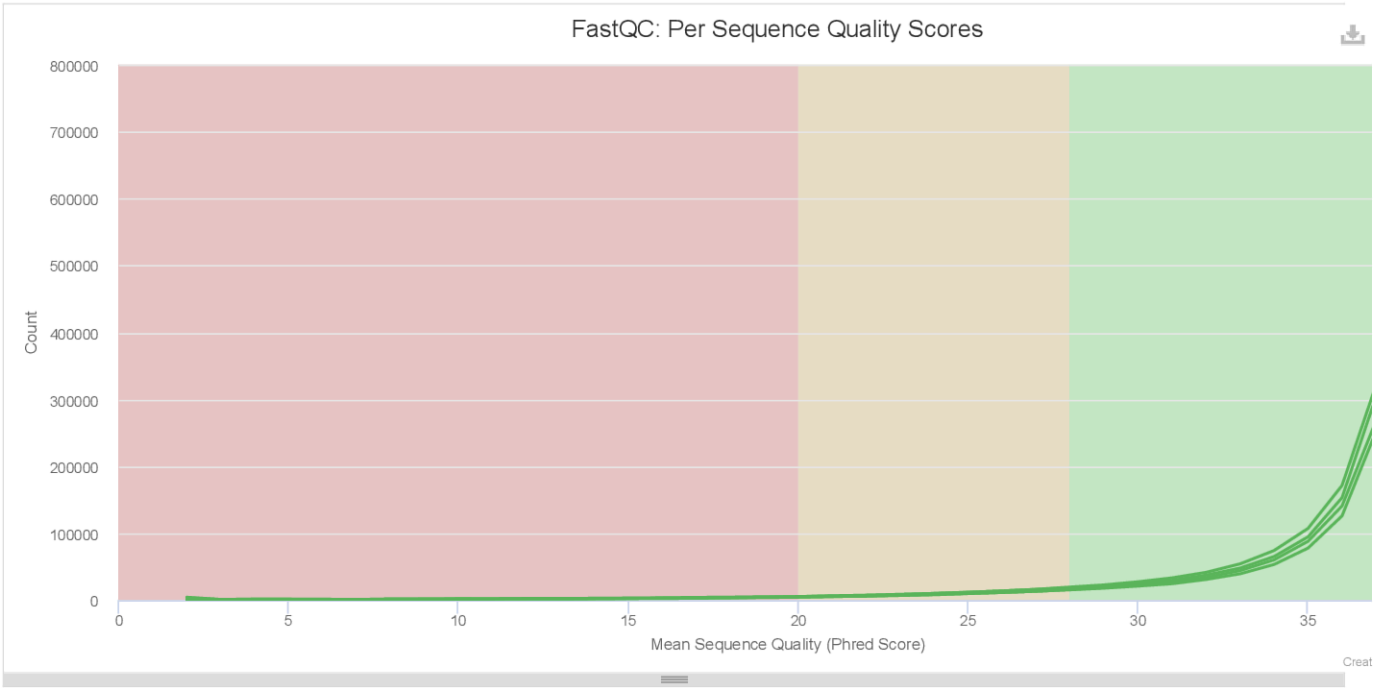


Per Sequence Quality Scores

4

Help

The number of reads with average quality scores. Shows if a subset of reads has poor quality.



Per Base Sequence Content

4

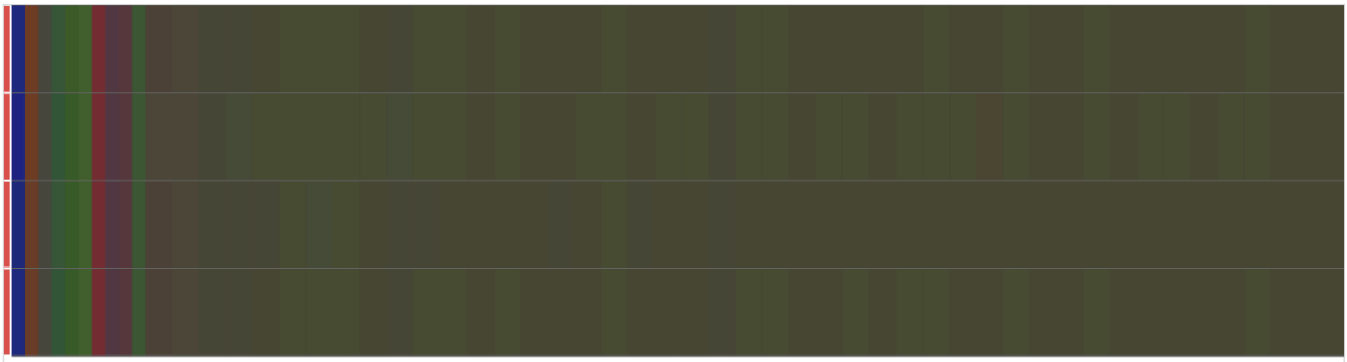
Help

The proportion of each base position for which each of the four normal DNA bases has been called.

Click a sample row to see a line plot for that dataset.

Rollover for sample name

Position: - %T: - %C: - %A: - %G: - Export Plot



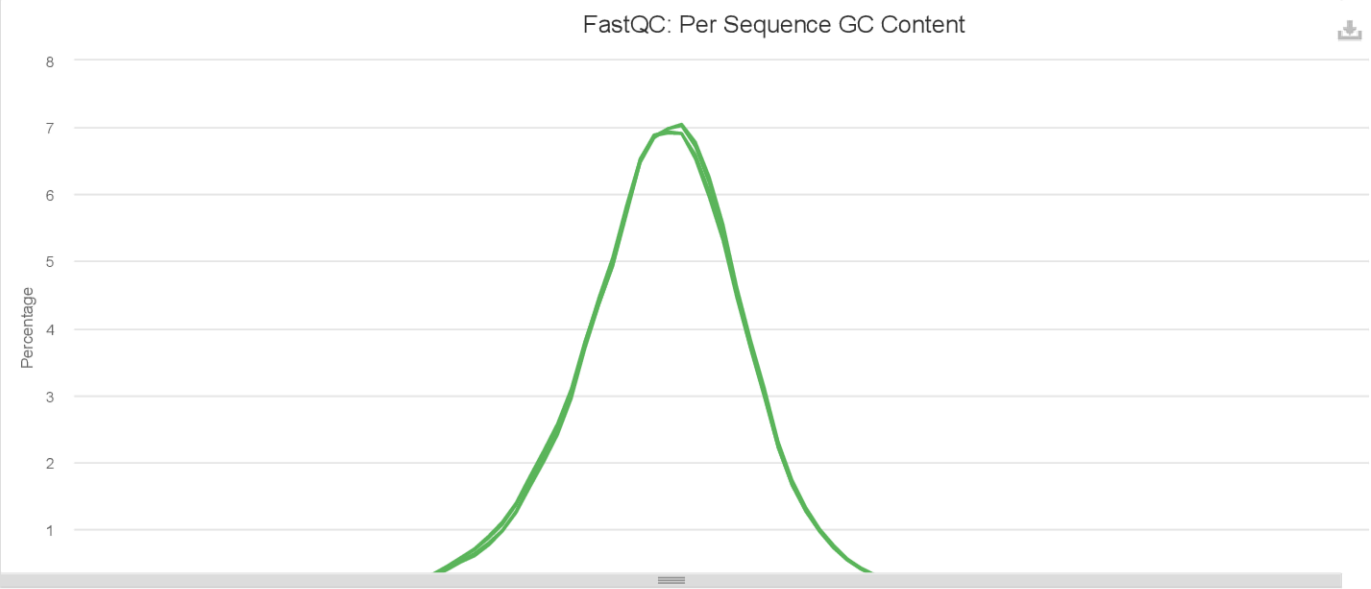
Per Sequence GC Content

4

Help

The average GC content of reads. Normal random library typically have a roughly normal distribution of GC content.

Percentages | Counts

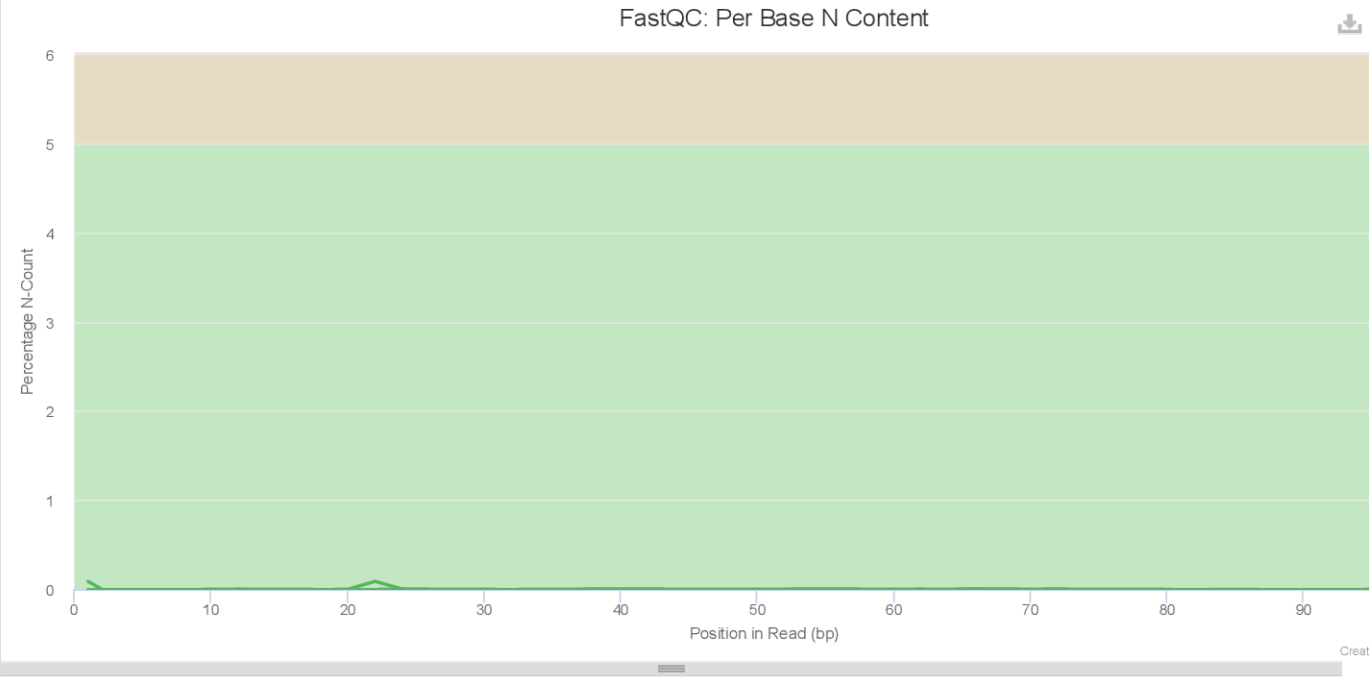


Per Base N Content

4

Help

The percentage of base calls at each position for which an N was called.



Sequence Length Distribution

4

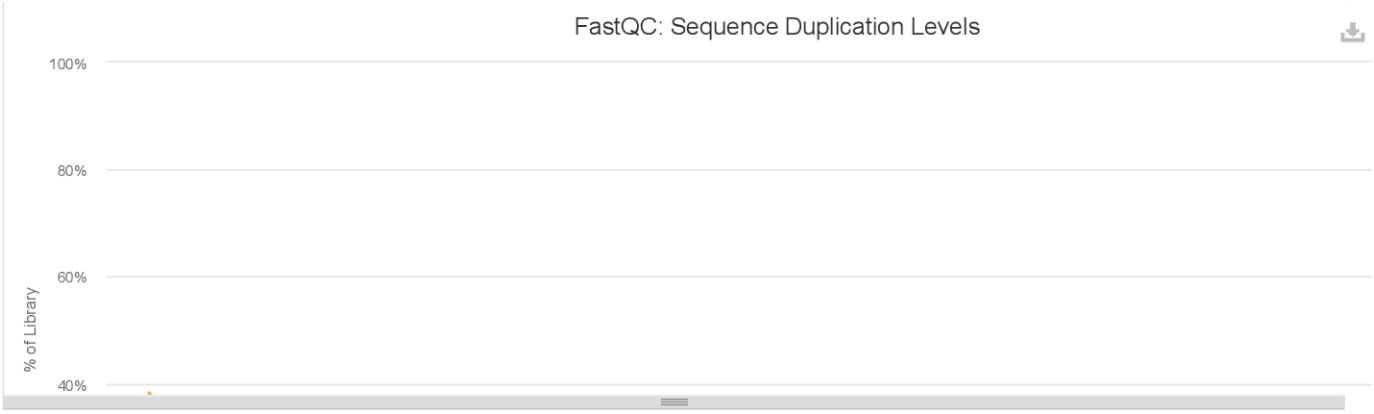
All samples have sequences of a single length (101bp).

Sequence Duplication Levels

4

Help

The relative level of duplication found for every sequence.



Overrepresented sequences 4

Help

The total amount of overrepresented sequences found in each library.

4 samples had less than 1% of reads made up of overrepresented sequences

Adapter Content 4

Help

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

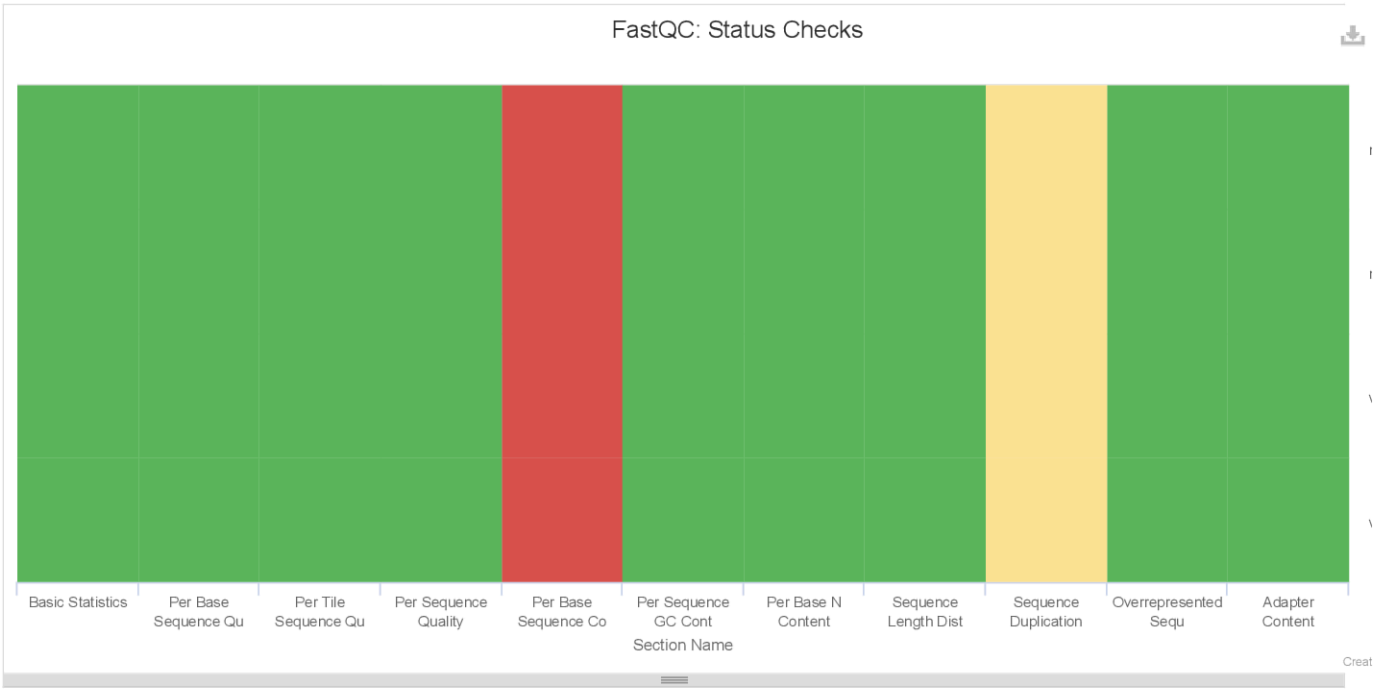
No samples found with any adapter contamination > 0.1%

Status Checks

Help

Status for each FastQC section showing whether results seem entirely normal (green), slightly abnormal (orange) or very unusual (red).

Sort by highlight



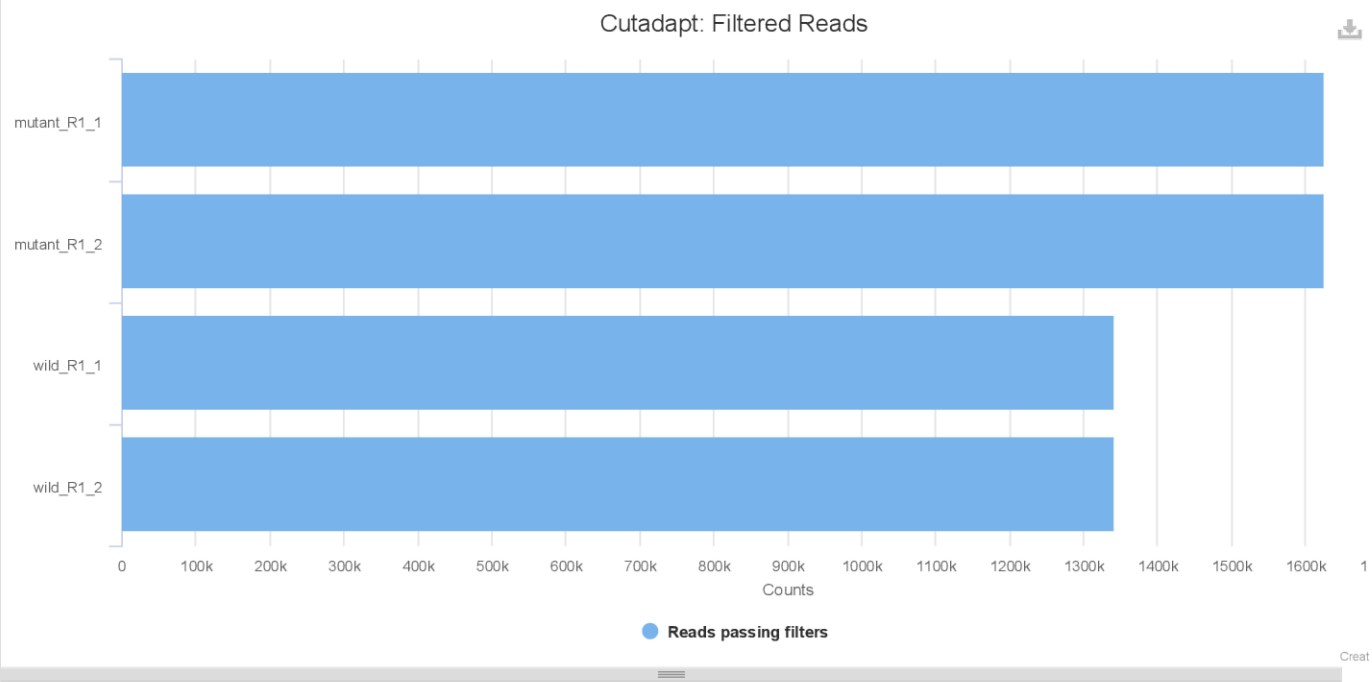
Cutadapt

Cutadapt is a tool to find and remove adapter sequences, primers, poly-Atails and other types of unwanted sequence from your high-throughput sequencing reads.

Filtered Reads

This plot shows the number of reads (SE) / pairs (PE) removed by Cutadapt.

Counts | Percentages

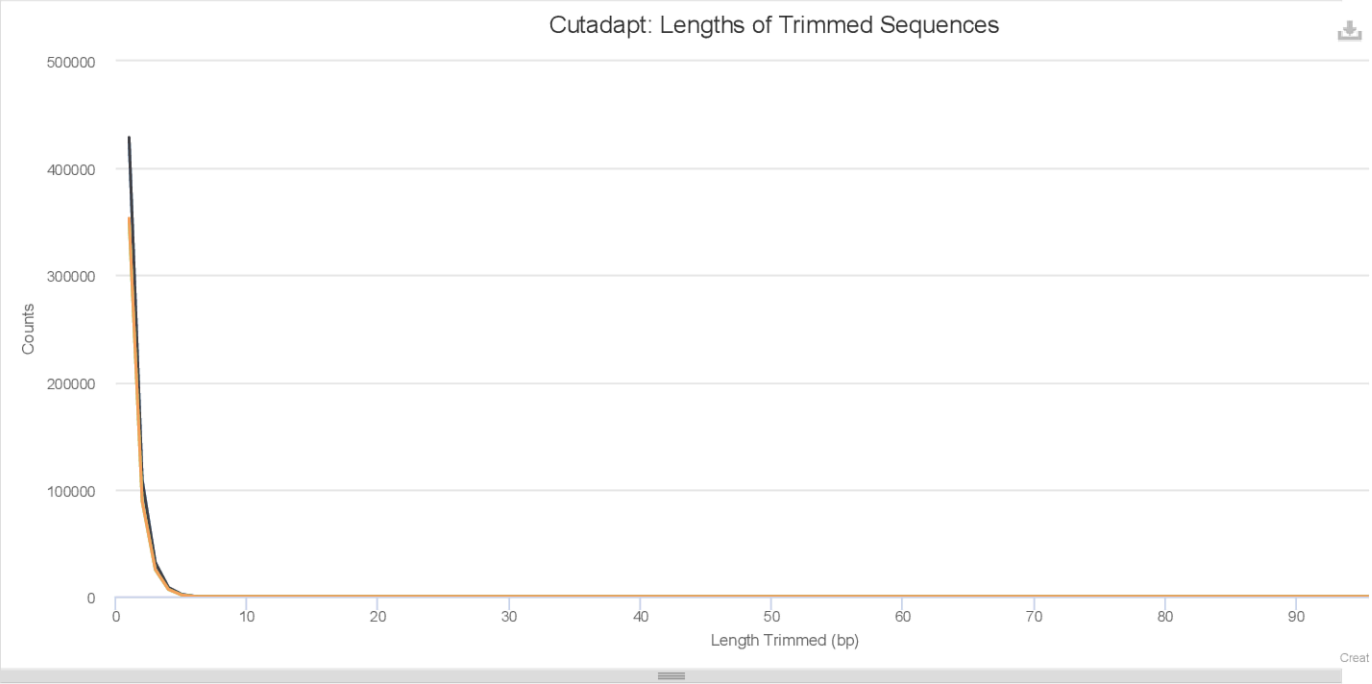


Trimmed Sequence Lengths

Help

This plot shows the number of reads with certain lengths of adapter trimmed.

Counts | Obs/Exp



FastQC (trimmed)

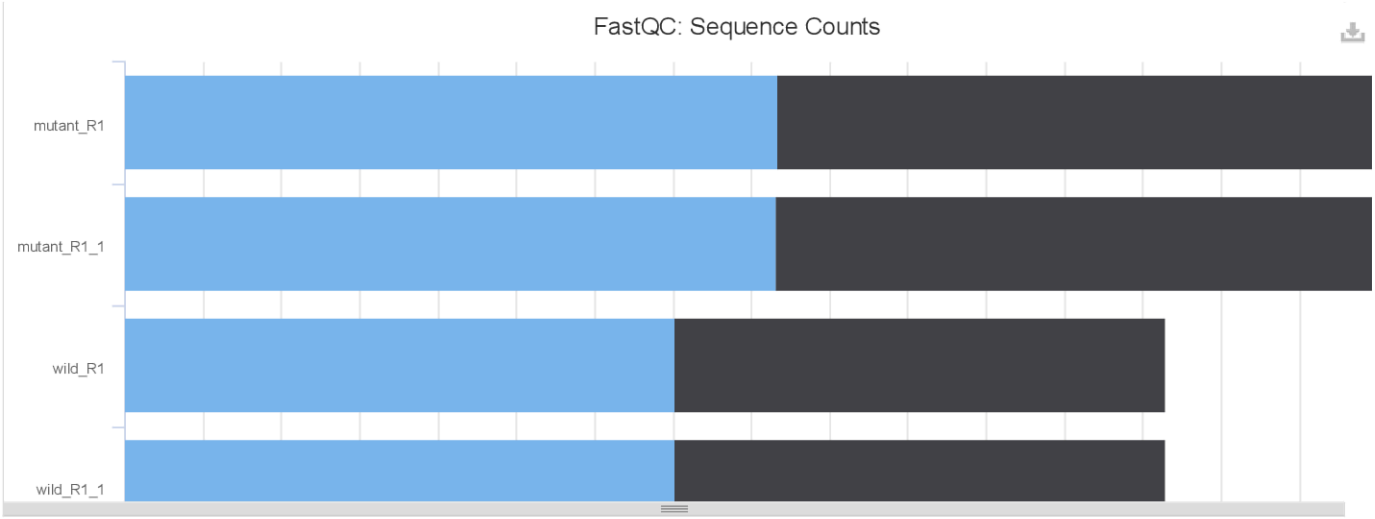
FastQC (trimmed) This section of the report shows FastQC results after adapter trimming.

Sequence Counts

Help

Sequence counts for each sample. Duplicate read counts are an estimate only.

Number of reads | Percentages

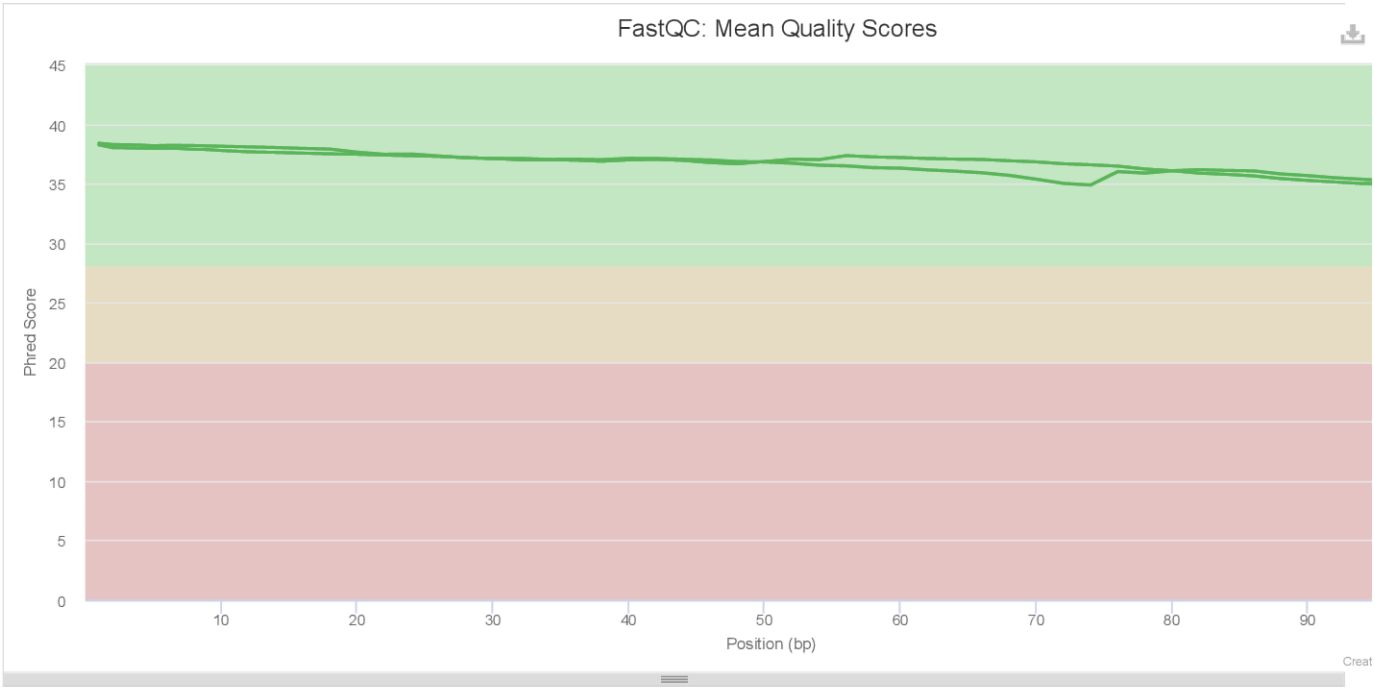


Sequence Quality Histograms

4

Help

The mean quality value across each base position in the read.



Per Sequence Quality Scores

4

Help

The number of reads with average quality scores. Shows if a subset of reads has poor quality.

Per Base Sequence Content

4

Help

The proportion of each base position for which each of the four normal DNA bases has been called.

Click a sample row to see a line plot for that dataset.

Rollover for sample name

Position: - %T: - %C: - %A: - %G: - Export Plot

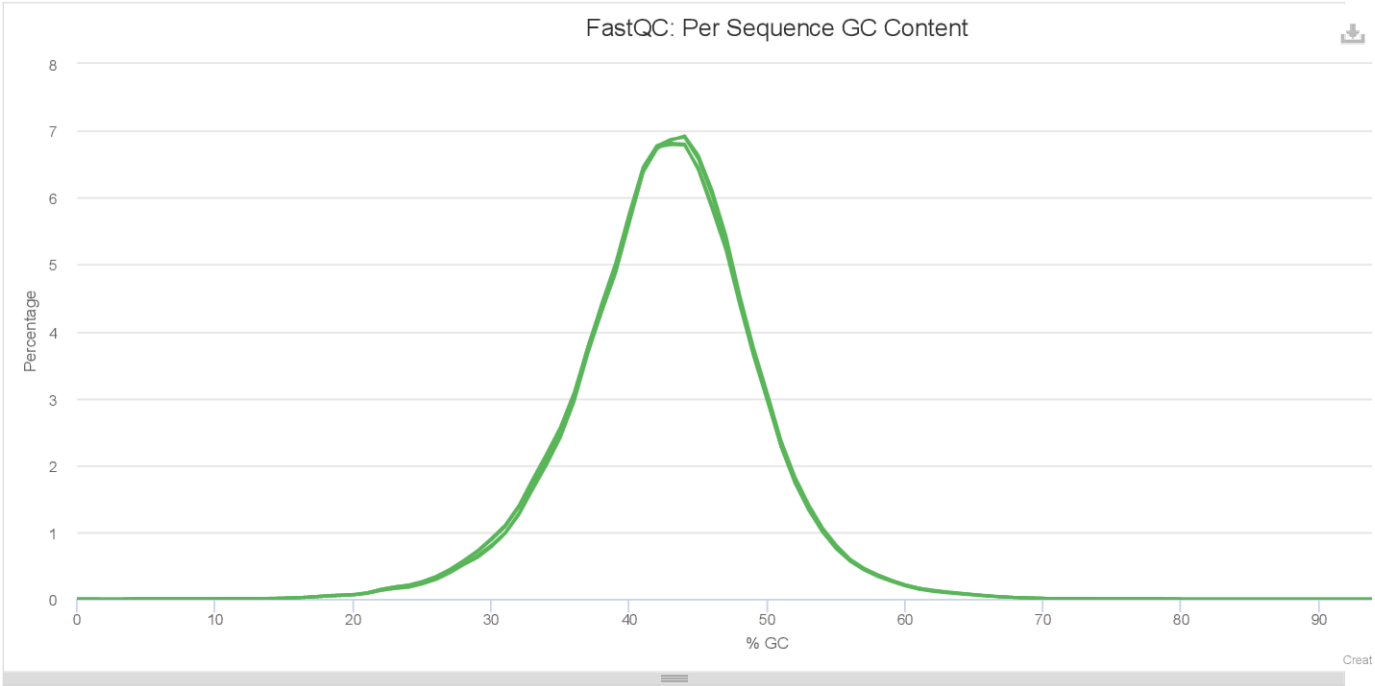
Per Sequence GC Content

4

Help

The average GC content of reads. Normal random library typically have a roughly normal distribution of GC content.

Percentages | Counts

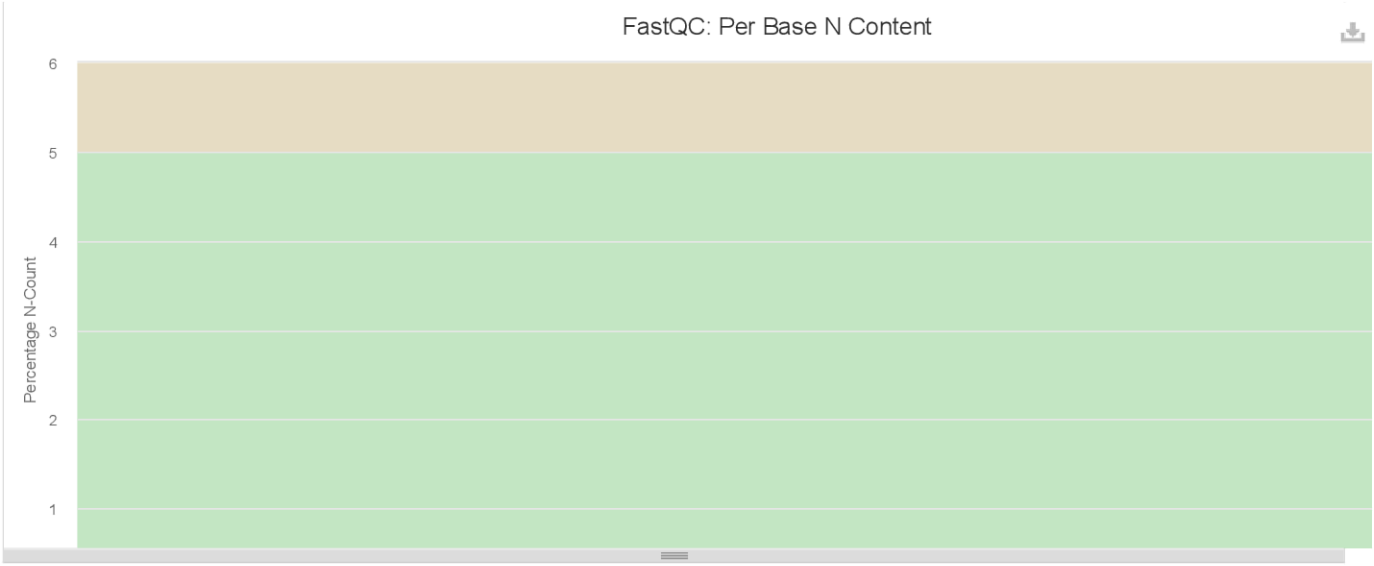


Per Base N Content

4

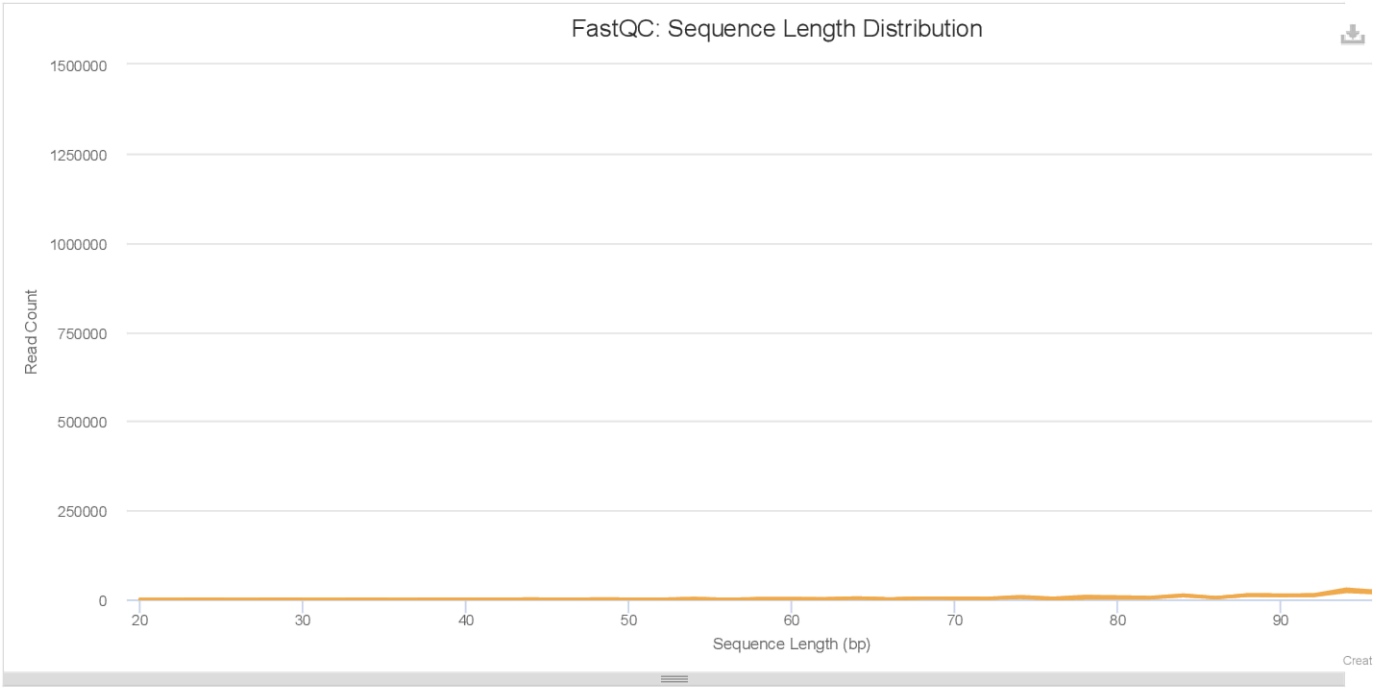
Help

The percentage of base calls at each position for which an N was called.



Sequence Length Distribution 4

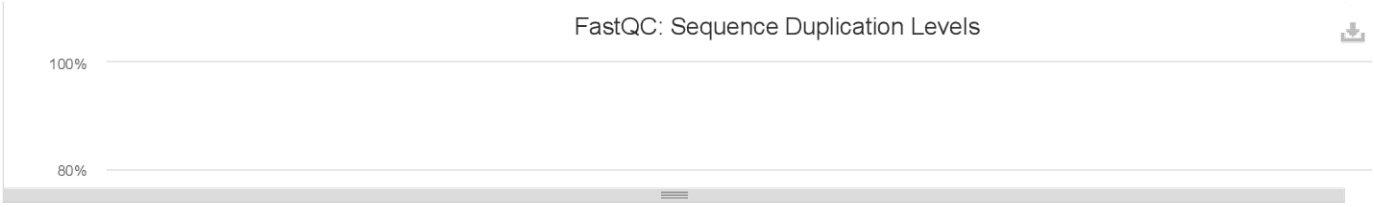
The distribution of fragment sizes (read lengths) found. See the FastQC help



Sequence Duplication Levels 4

Help

The relative level of duplication found for every sequence.



Overrepresented sequences 4

Help

The total amount of overrepresented sequences found in each library.

4 samples had less than 1% of reads made up of overrepresented sequences

Adapter Content 4

Help

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

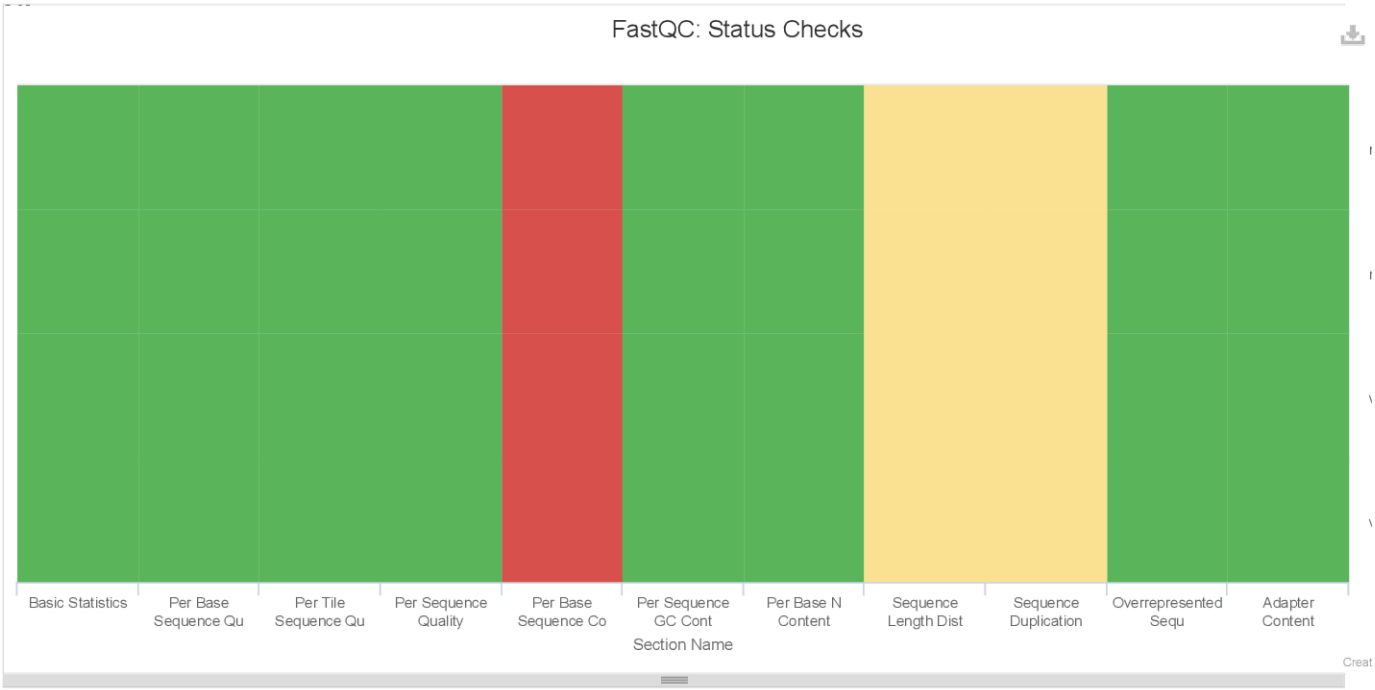
No samples found with any adapter contamination > 0.1%

Status Checks

Help

Status for each FastQC section showing whether results seem entirely normal (green), slightly abnormal (orange) or very unusual (red).

Sort by highlight



nf-core/rnaseq Software Versions

are collected at run time from the software output.

bedtools
2.29.2
deseq2
1.28.0
dupradar
1.18.0
fastqc
0.11.9
nextflow
21.04.1
nf-core/rnaseq
3.0
picard
2.23.9

preseq
2.0.3
qualimap
2.2.2-dev
rsem
1.3.1
rseqc
3.0.1
samtools
1.10
stringtie
2.1.4
subread
2.0.1
trimgalore
0.6.6
ucsc
377

---

nf-core/rnaseq Workflow Summary

- this information is collected when the pipeline is started.

Core Nextflow options

revision
3.0
runName
thirsty_goodall
containerEngine
singularity
launchDir
/work/laurier/PROJET_NEXTFLOW/TOMATES
workDir
/work/laurier/PROJET_NEXTFLOW/TOMATES/work
projectDir
/home/laurier/.nextflow/assets/nf-core/rnaseq
userName
laurier
profile
genotoul
configFiles
/home/laurier/.nextflow/assets/nf-core/rnaseq/nextflow.config, /home/laurier/work/PROJET_NEXTFLOW/TOMATES/sm_config.cfg

Input/output options

input
/home/laurier/work/PROJET_NEXTFLOW/TOMATES/inputs.csv

Reference genome options

fasta
/home/laurier/work/PROJET_NEXTFLOW/TOMATES/GENOME_REF/ITAG2.3_genomic_Ch6.fasta
gtf
/home/laurier/work/PROJET_NEXTFLOW/TOMATES/GENOME_REF/ITAG2.3_genomic_Ch6.gtf
save_reference
true
igenomes_ignore
true

Alignment options

aligner
star_rsem

Institutional config options

config_profile_description
The Genotoul cluster profile
config_profile_contact
support.bioinfo.genotoul@inra.fr
config_profile_url
http://bioinfo.genotoul.fr/

Max job request options

max_cpus

```
48
```
max_memory
```
120 GB
```
max_time
```
4d
```

CHAPITRE 5

ANNEXE B - MULTIQC - MORUE

MultiQC

A modular tool to aggregate results from bioinformatics analyses across many samples into a single report.

This report has been generated by the nf-core/rnaseq analysis pipeline. For information about how to interpret these results, please see the documentation.

Report generated on 2023-10-07, 18:28 based on data in: `/work/laurier/PROJET_NEXTFLOW/MORUE/work/84/fe9ad7e5229c6cf9d4dad3065a580b`

Welcome! Not sure where to start? Watch a tutorial video *(6:06)* don't show again ×

General Statistics

Copy table | Configure Columns | Plot Showing 2/2 rows and 19/26 columns.

Sample Name	M Reads Mapped	% rRNA	dupInt	% Dups	5'-3' bias	M Aligned	% Alignable	Error rate	M Non-Primary	M Reads
ovary_R1	46.9	0.19%	0.06%	67.2%	1.23	40.1	85.7%	1.15%	6.8	40.1
testis_R1	38.3	0.37%	0.01%	56.4%	1.27	34.0	74.5%	0.68%	4.3	34.0

WARNING: Fail Strand Check

List of samples that failed the strandedness check between that provided in the samplesheet and calculated by the RSeQC infer_experiment.py tool.

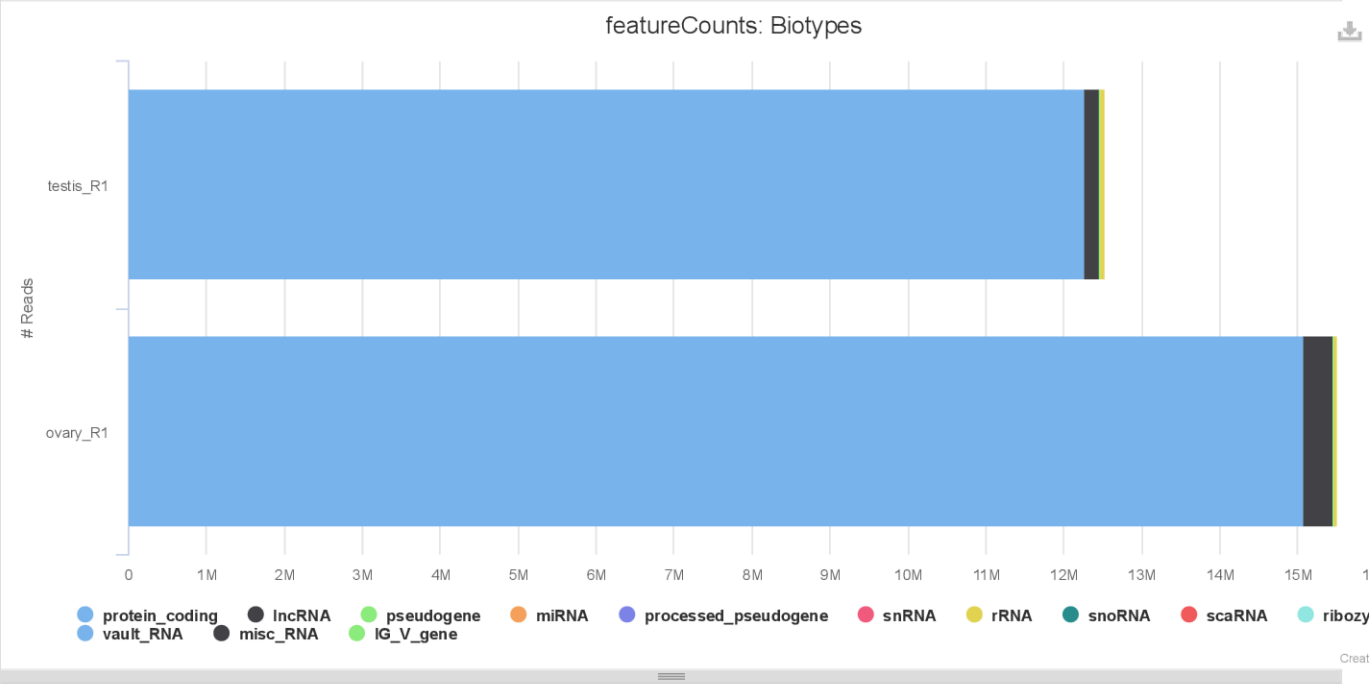
Copy table | Configure Columns | Plot Showing 2/2 rows and 5/5 columns.

Sample	Provided strandedness	Inferred strandedness	Sense (%)	Antisense (%)	Undetermined (%)
testis_R1	forward	unstranded	49.69	47.54	2.77
ovary_R1	forward	unstranded	48.98	48.08	2.94

Biotype Counts

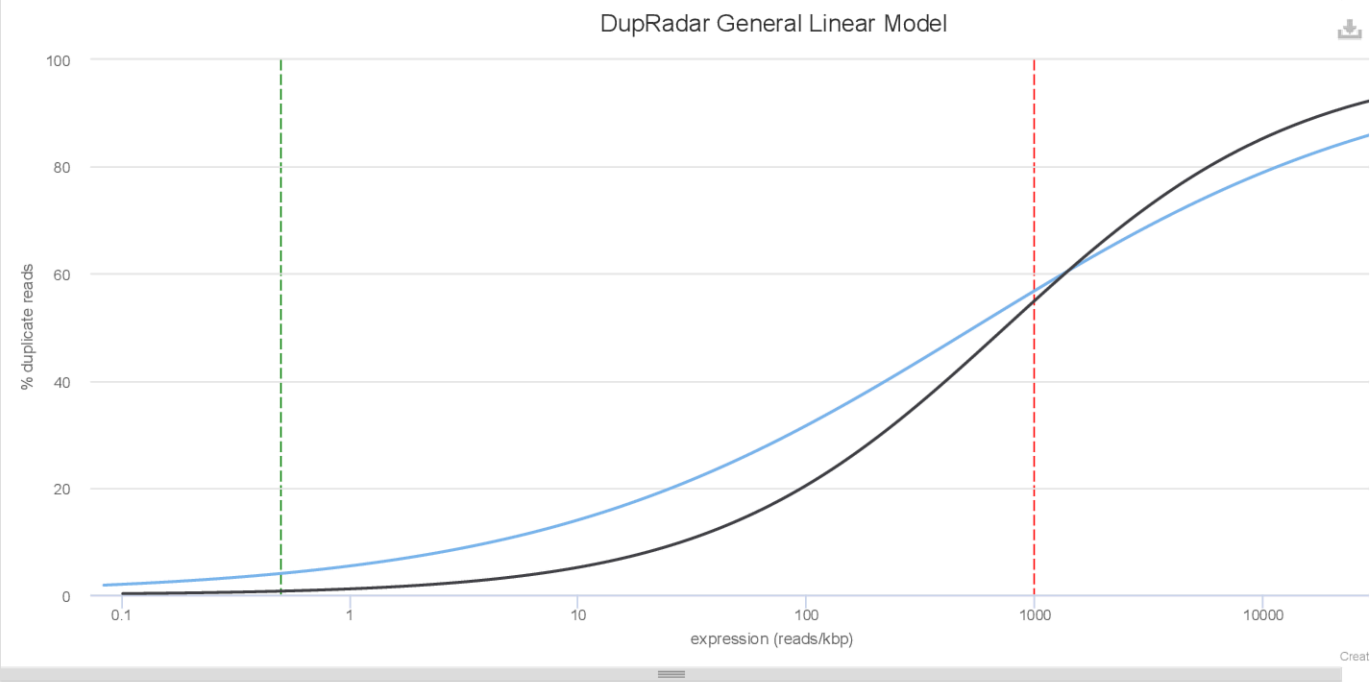
shows reads overlapping genomic features of different biotypes, counted by featureCounts.

Number of Reads | Percentages



DupRadar

provides duplication rate quality control for RNA-Seq datasets. Highly expressed genes can be expected to have a lot of duplicate reads, but high numbers of duplicates at low read counts can indicate low library complexity with technical duplication. This plot shows the general linear models - a summary of the gene duplication distributions.



Picard

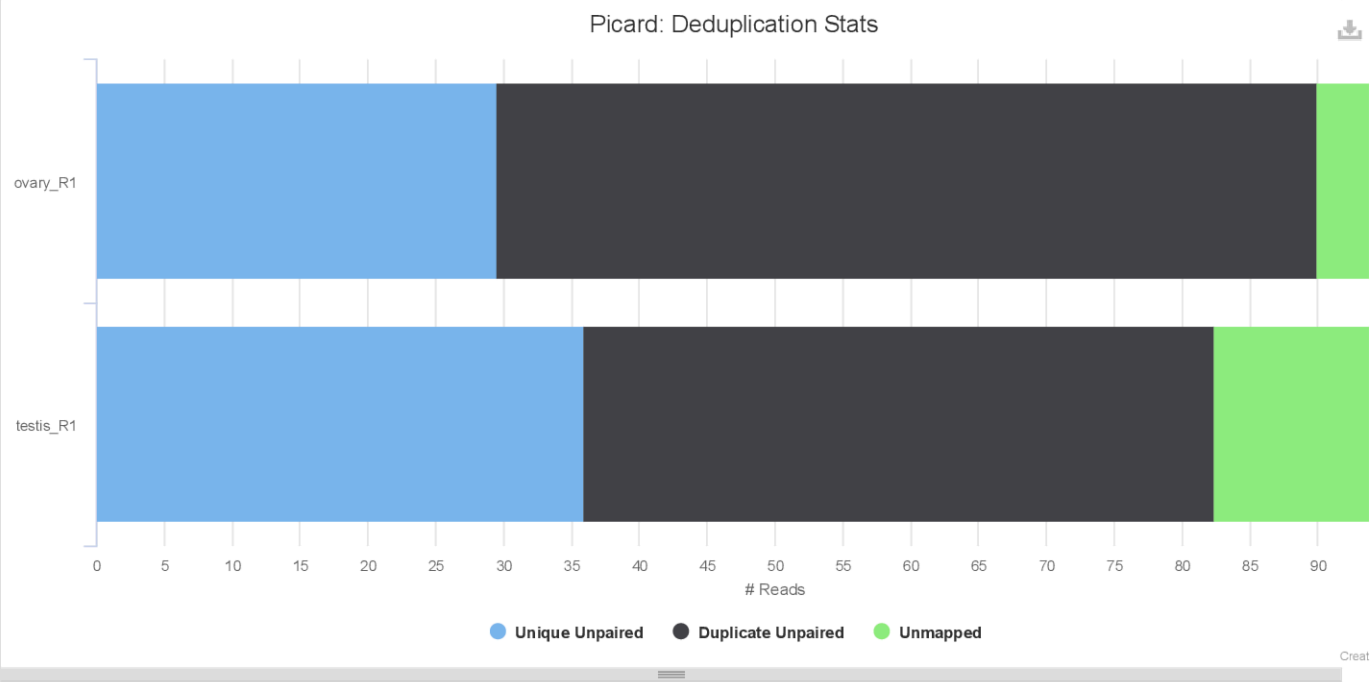
Picard is a set of Java command line tools for manipulating high-throughput sequencing data.

Mark Duplicates

Help

Number of reads, categorised by duplication state. **Pair counts are doubled** - see help text for details.

Number of Reads | Percentages

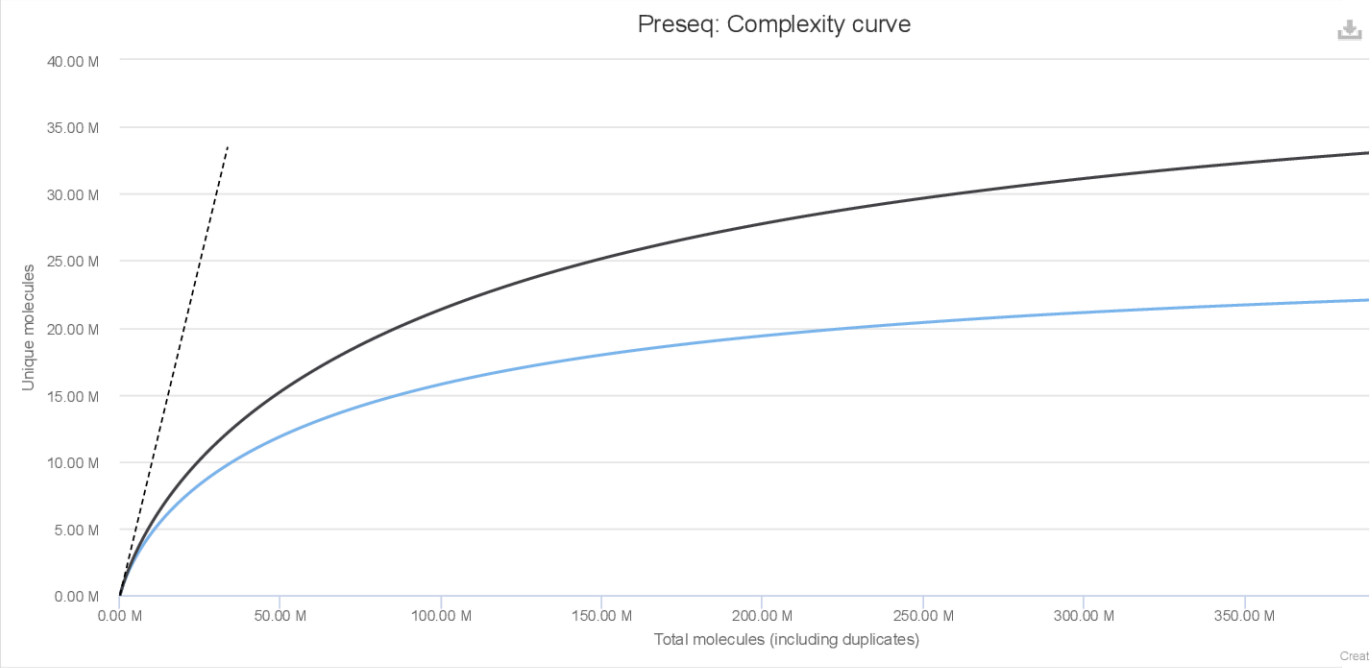


Preseq

Preseq estimates the complexity of a library, showing how many additional unique reads are sequenced for increasing total read count. A shallow curve indicates complexity saturation. The dashed line shows a perfectly complex library where total reads = unique reads.

Complexity curve

Note that the x axis is trimmed at the point where all the datasets show 80% of their maximum y-value, to avoid ridiculous scales.



QualiMap

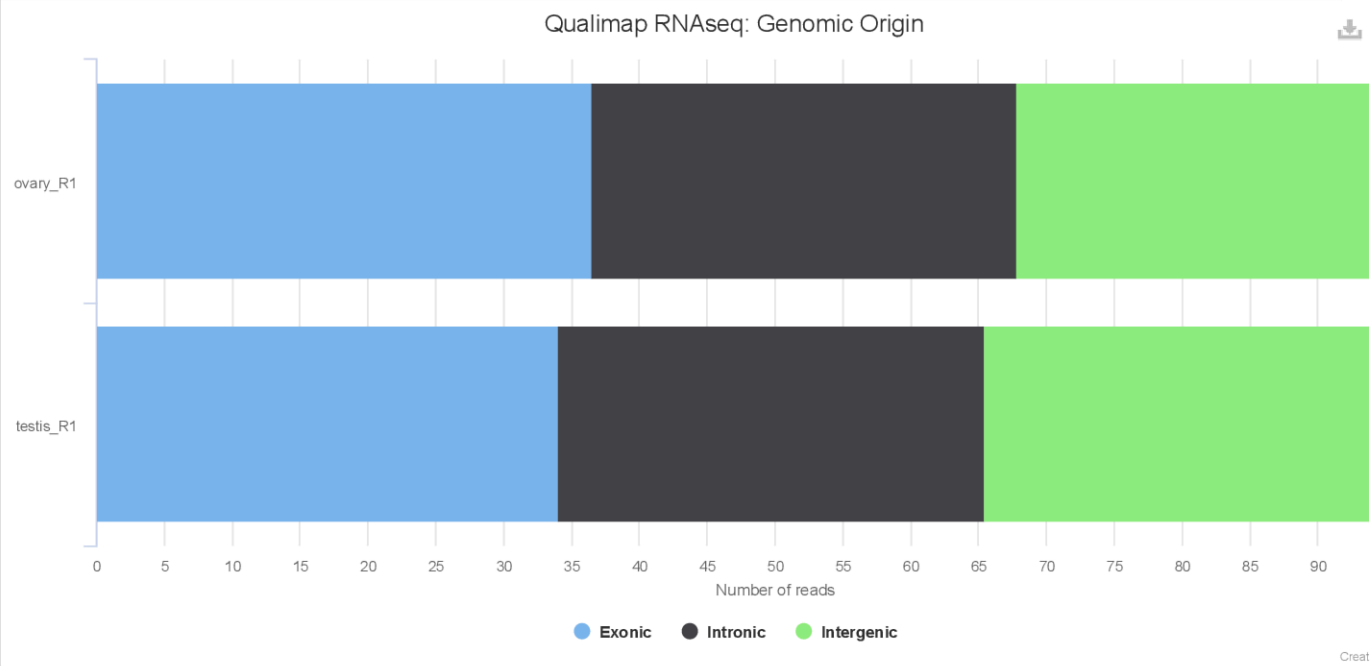
QualiMap is a platform-independent application to facilitate the quality control of alignment sequencing data and its derivatives like feature counts.

Genomic origin of reads

Help

Classification of mapped reads as originating in exonic, intronic or intergenic regions. These can be displayed as either the number or percentage of mapped reads.

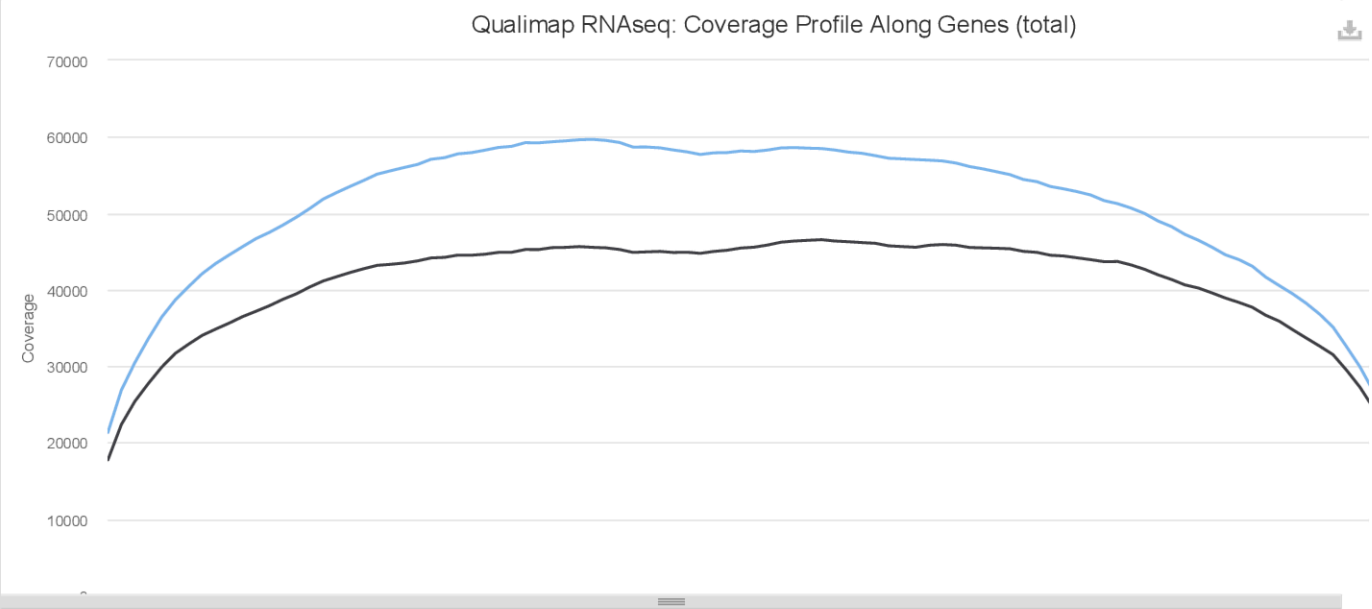
Counts | Percentages



Gene Coverage Profile

Help

Mean distribution of coverage depth across the length of all mapped transcripts.



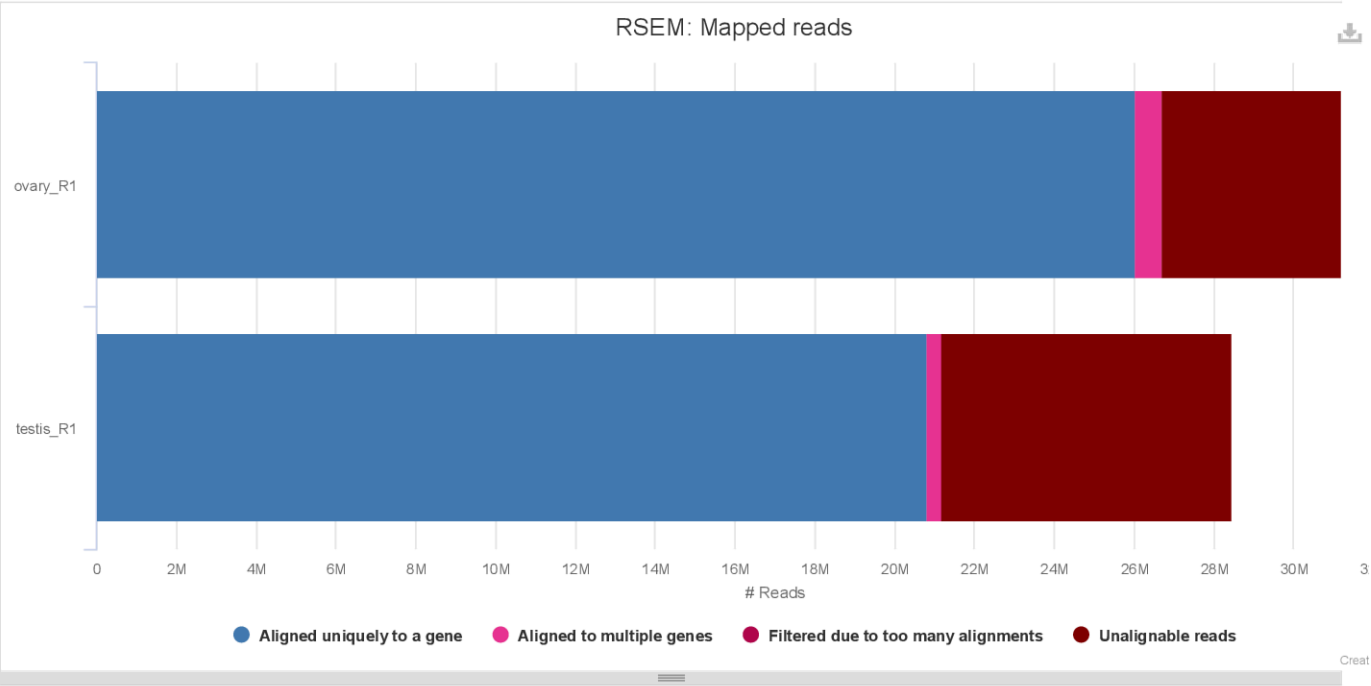
Rsem

Rsem RSEM (RNA-Seq by Expectation-Maximization) is a software package forestimating gene and isoform expression levels from RNA-Seq data.

Mapped Reads

A breakdown of how all reads were aligned for each sample.

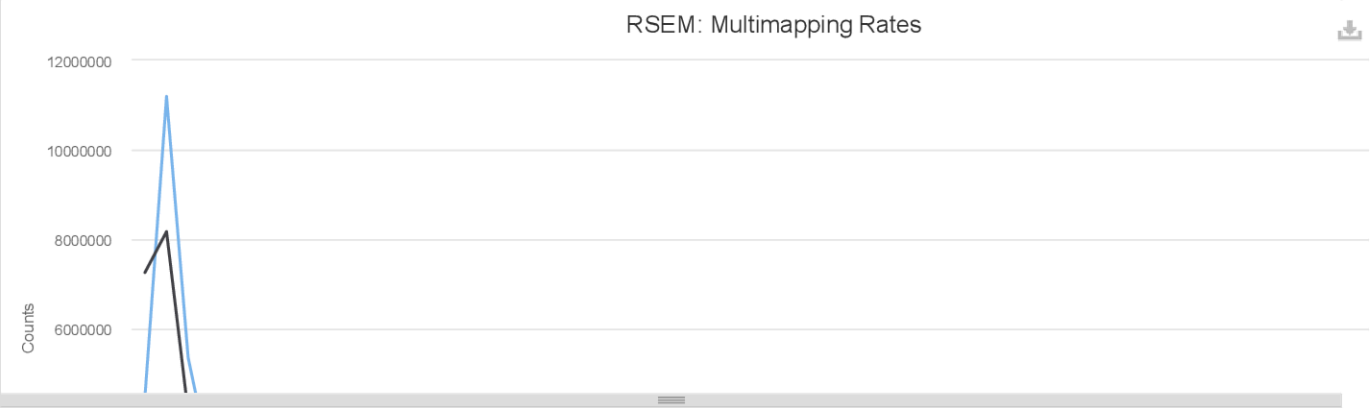
Number of Reads | Percentages



Multimapping rates

Help

A frequency histogram showing how many reads were aligned to n reference regions.



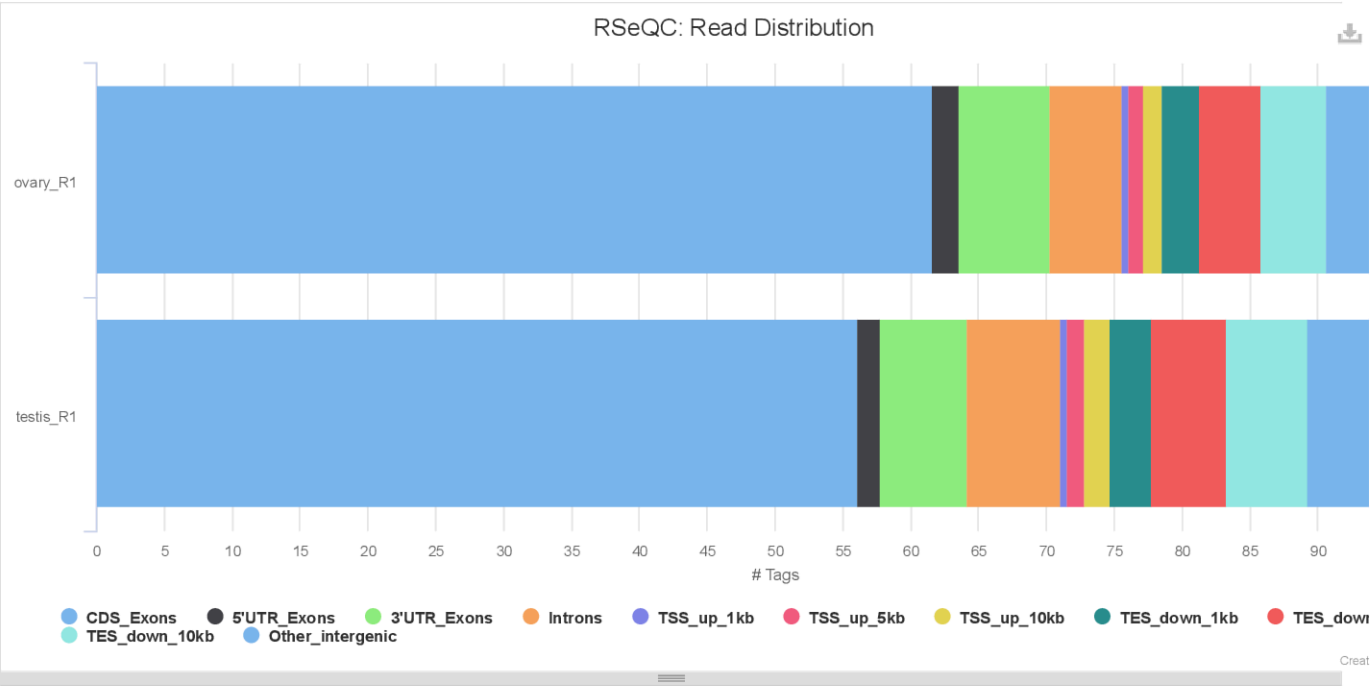
RSeQC

RSeQC package provides a number of useful modules that can comprehensively evaluate high throughput RNA-seq data.

Read Distribution

Read Distribution calculates how mapped reads are distributed over genome features.

Number of Tags | Percentages



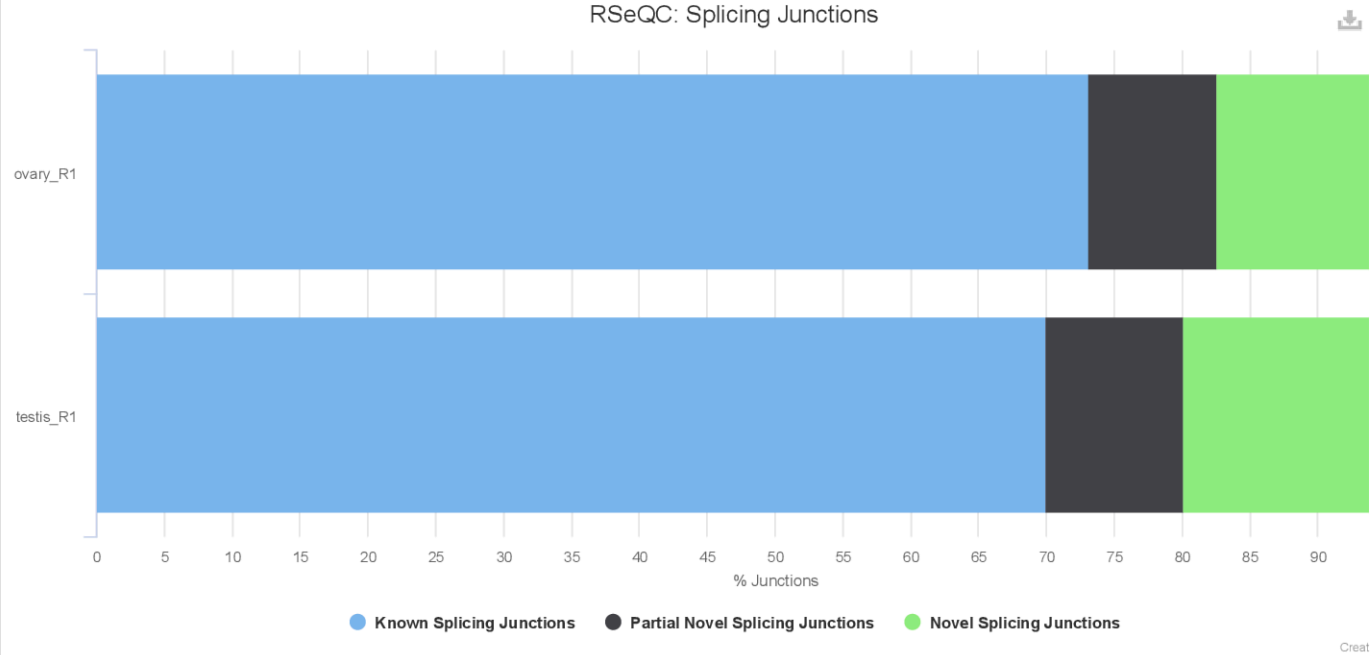
Read Duplication

read_duplication.py calculates how many alignment positions have a certain number of exact duplicates. Note - plot truncated at 500 occurrences and binned.

Junction Annotation

Junction annotation compares detected splice junctions to a reference gene model. An RNA read can be spliced 2 or more times, each time is called a splicing event.

Counts | Percentages | Junctions | Events



Junction Saturation

Junction Saturation counts the number of known splicing junctions that are observed in each dataset. If sequencing depth is sufficient, all (annotated) splice junctions should be rediscovered, resulting in a curve that reaches a plateau. Missing low abundance splice junctions can affect downstream analysis.

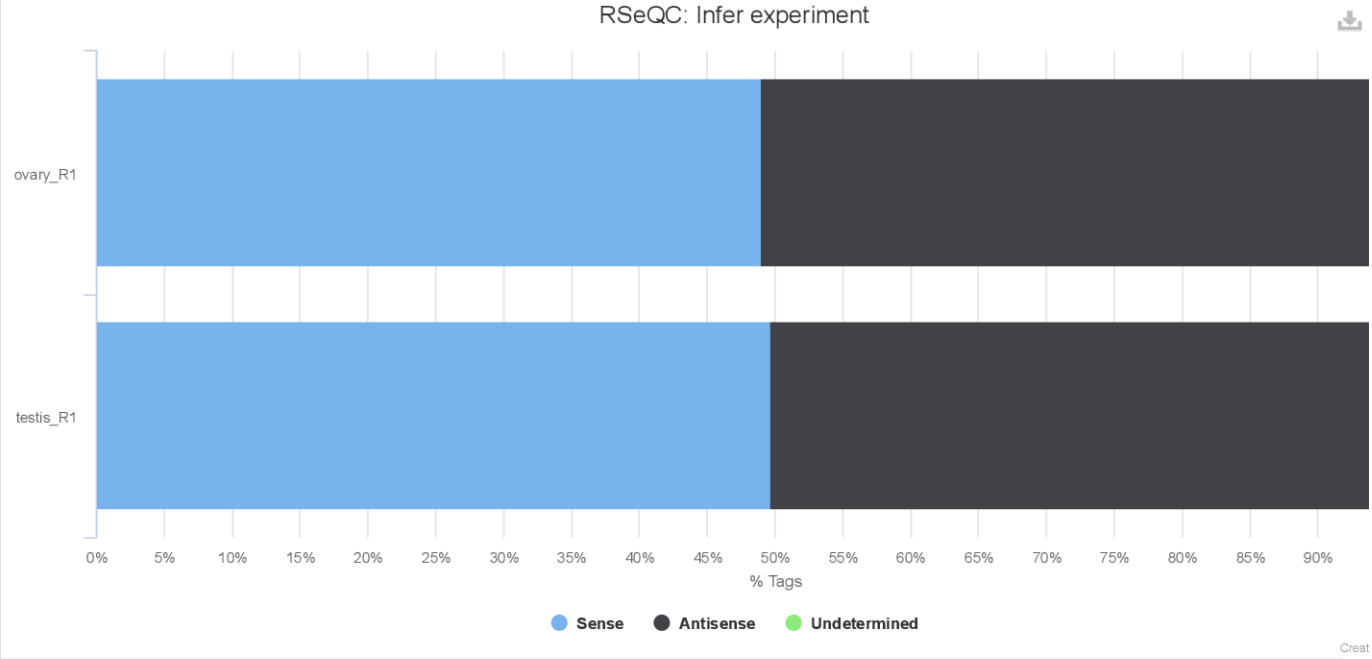
Click a line to see the data side by side (as in the original RSeQC plot).

All Junctions | Known Junctions | Novel Junctions

RSeQC: Junction Saturation

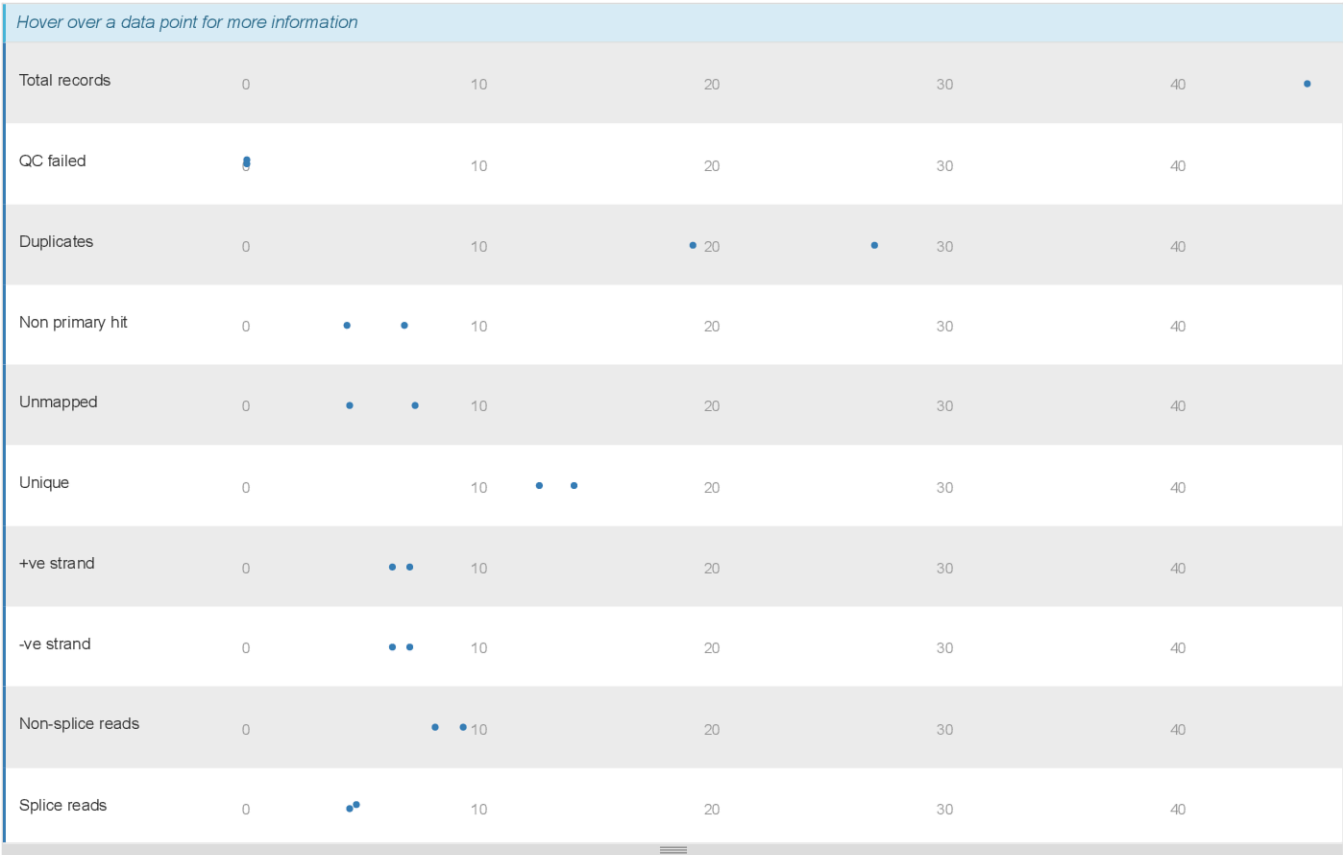
Infer experiment

Infer experiment counts the percentage of reads and read pairs that match the strandedness of overlapping transcripts. It can be used to infer whether RNA-seq library preps are stranded (sense or antisense).



Bam Stat

All numbers reported in millions.



Samtools

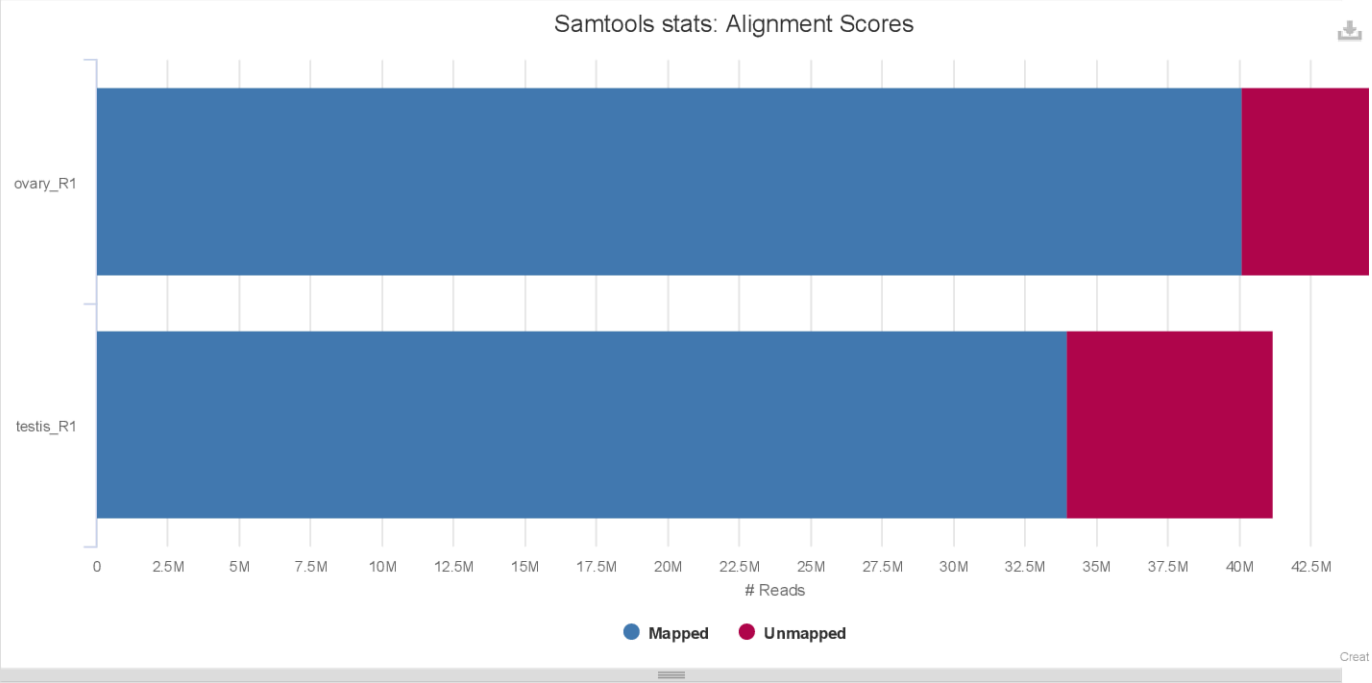
Samtools is a suite of programs for interacting with high-throughput sequencing data.

Percent Mapped

Help

Alignment metrics from `samtools stats`; mapped vs. unmapped reads.

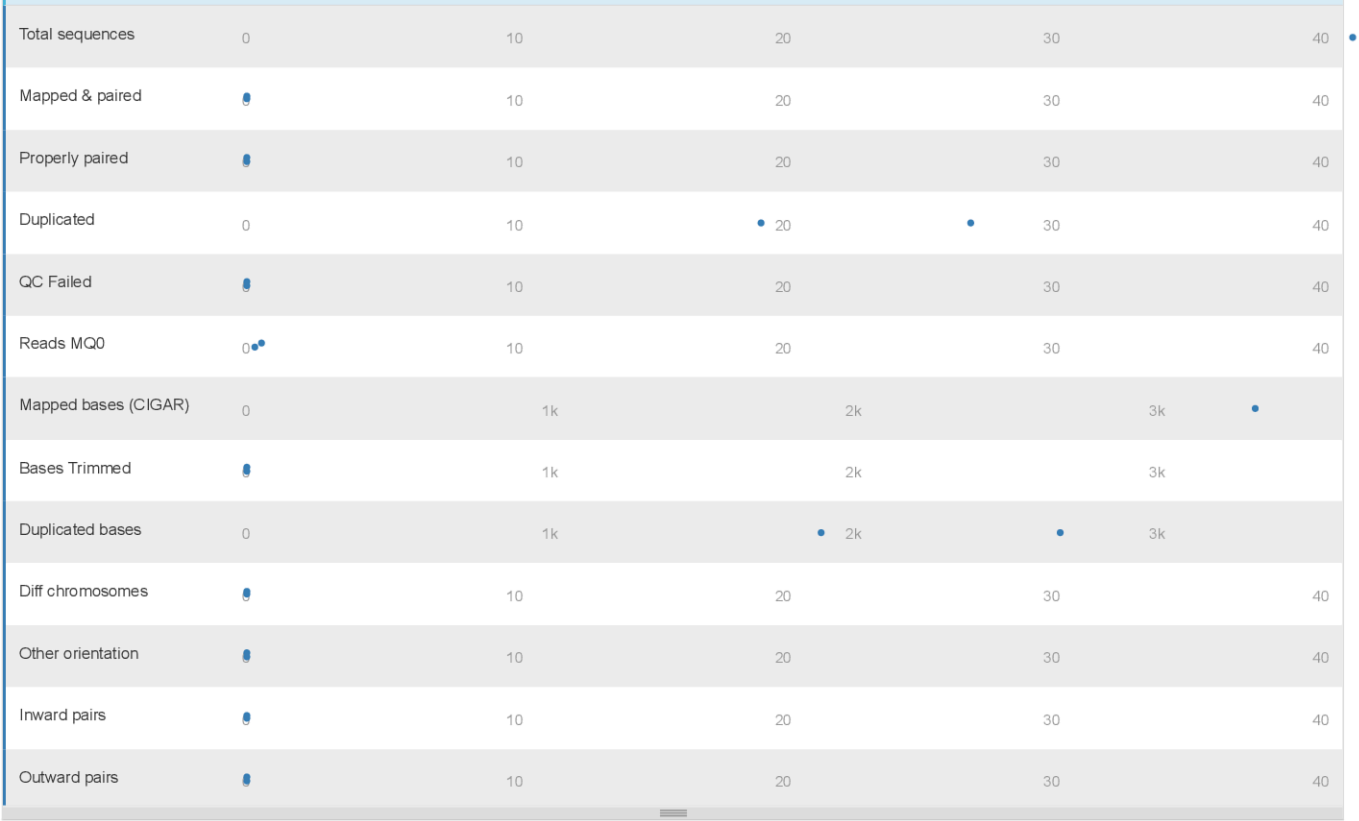
Number of Reads | Percentages



Alignment metrics

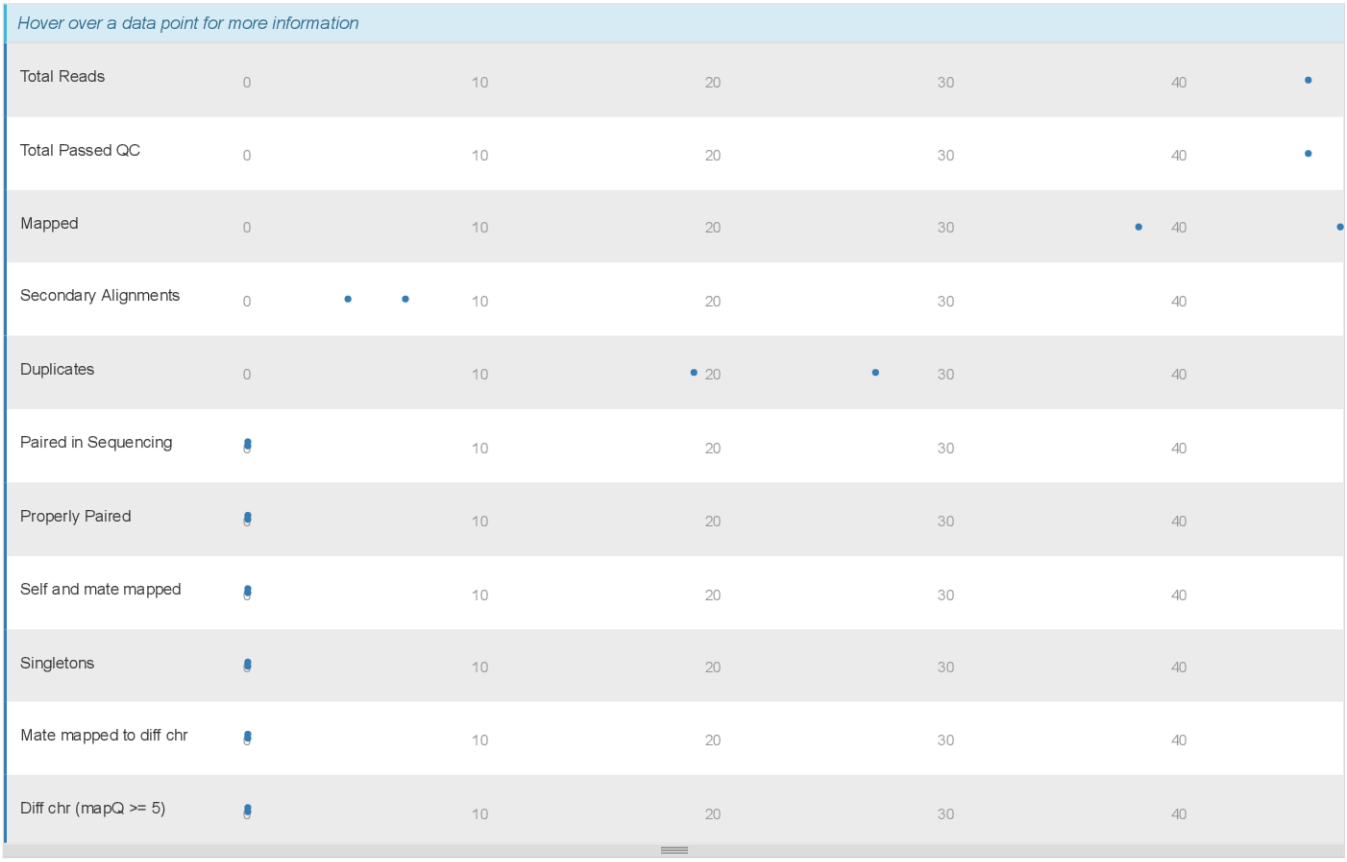
This module parses the output from `samtools stats`. All numbers in millions.

Hover over a data point for more information



Samtools Flagstat

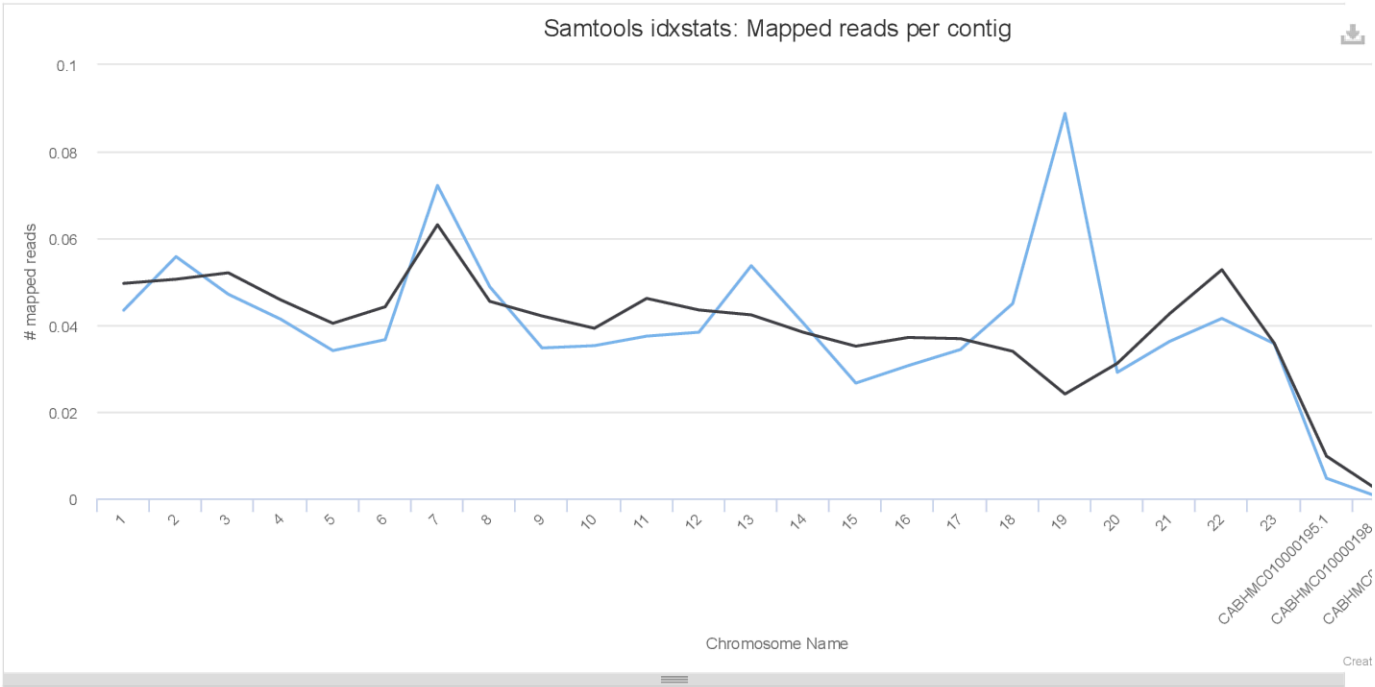
This module parses the output from `samtools flagstat`. All numbers in millions.



Mapped reads per contig

The `samtools idxstats` tool counts the number of mapped reads per chromosome / contig. Chromosomes with < 0.1% of the total aligned reads are omitted from this plot.

Counts | Log10 | Normalised Counts | Observed over Expected Counts | Raw Counts



FastQC (raw)

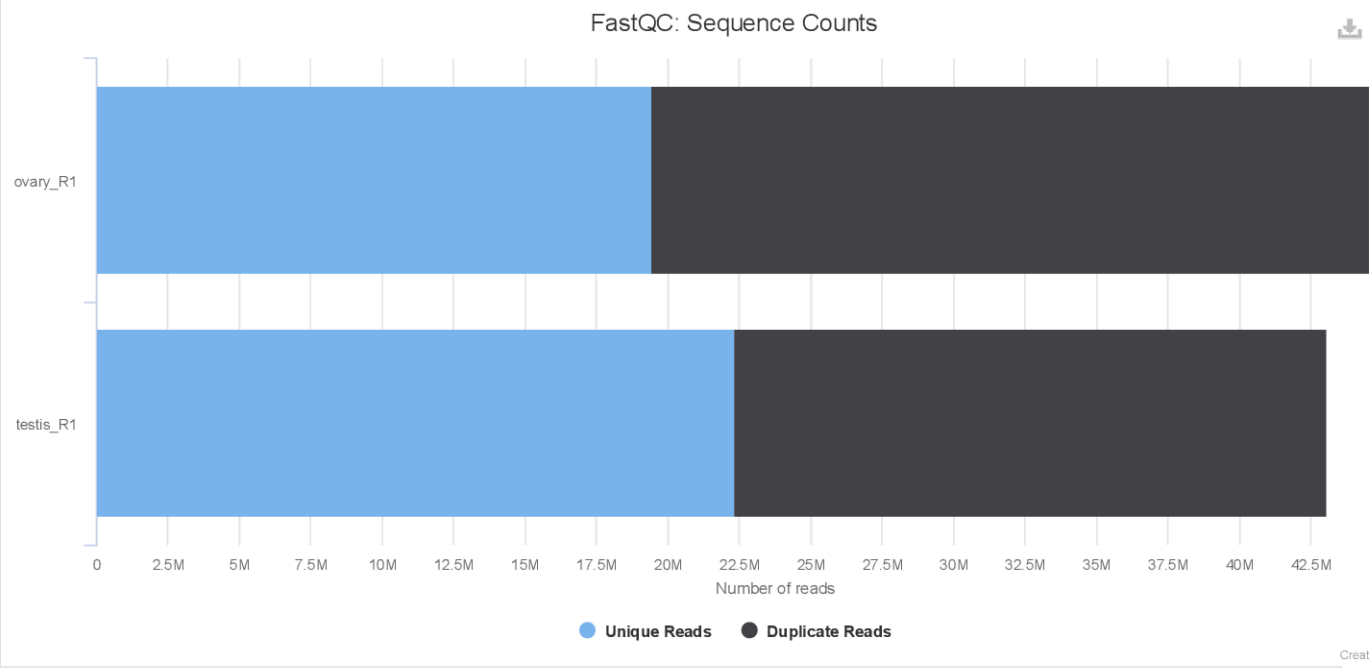
FastQC (raw) This section of the report shows FastQC results before adapter trimming.

Sequence Counts

Help

Sequence counts for each sample. Duplicate read counts are an estimate only.

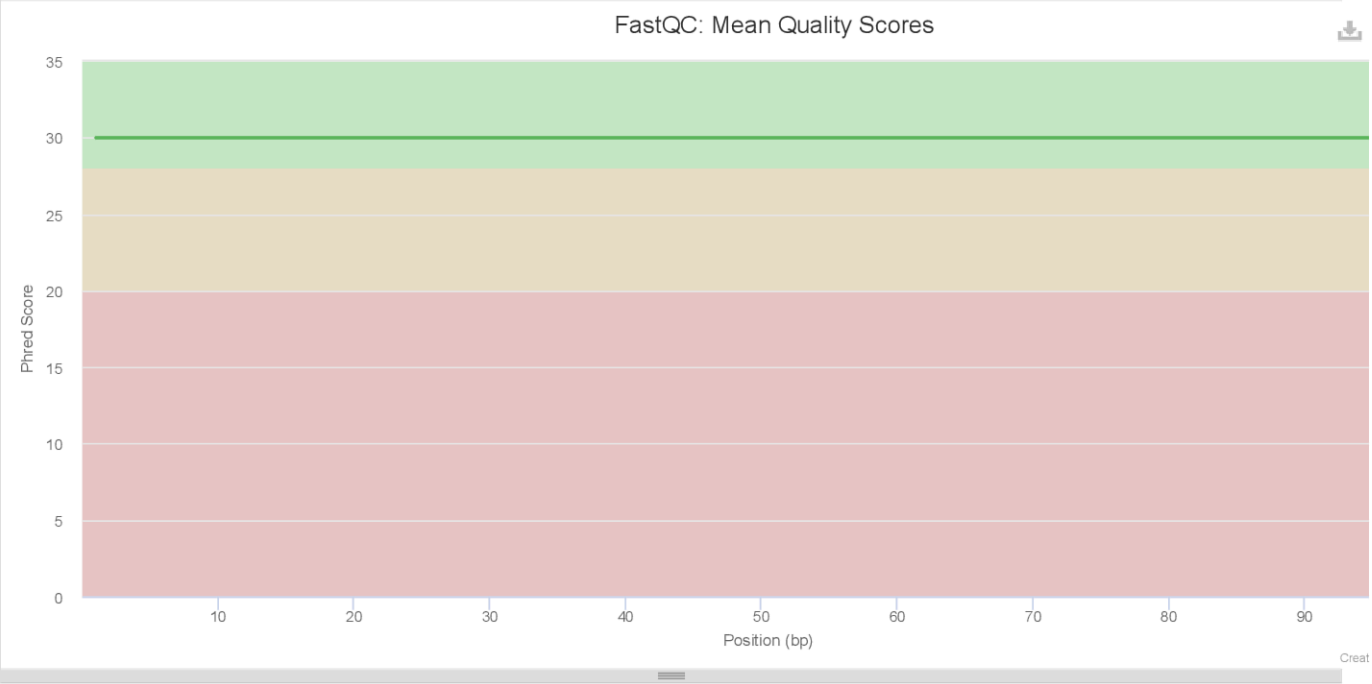
Number of reads | Percentages



Sequence Quality Histograms 2

Help

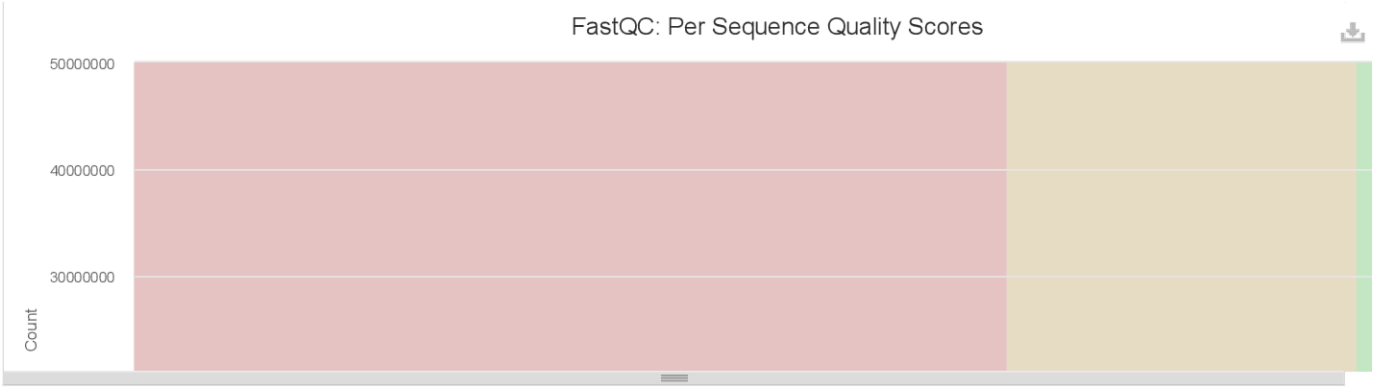
The mean quality value across each base position in the read.



Per Sequence Quality Scores 2

Help

The number of reads with average quality scores. Shows if a subset of reads has poor quality.



Per Base Sequence Content

1 1

Help

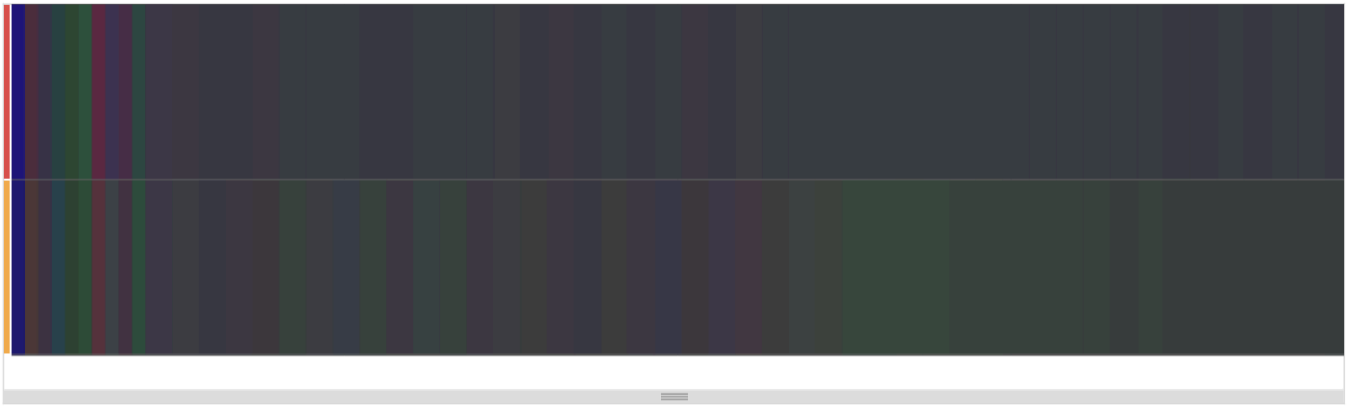
The proportion of each base position for which each of the four normal DNA bases has been called.

Click a sample row to see a line plot for that dataset.

Rollover for sample name

Position: - %T: - %C: - %A: - %G: -

Export Plot



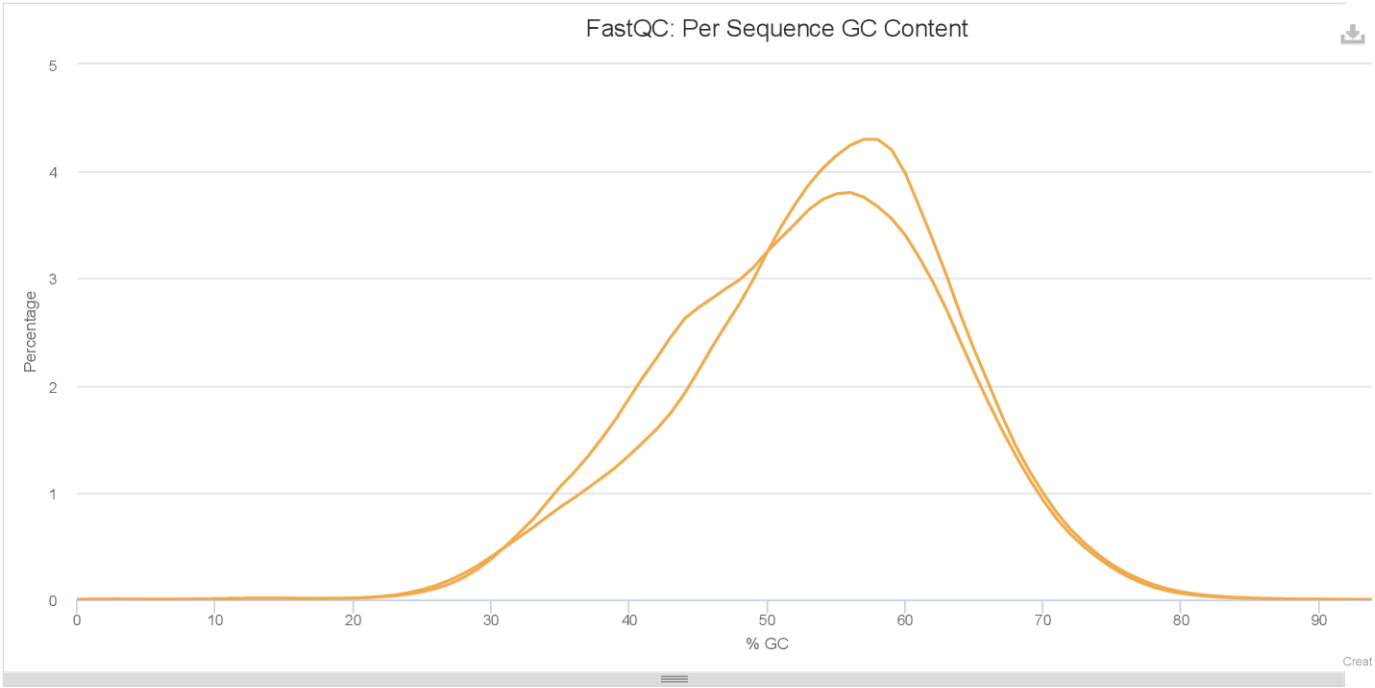
Per Sequence GC Content

2

Help

The average GC content of reads. Normal random library typically have a roughly normal distribution of GC content.

Percentages | Counts

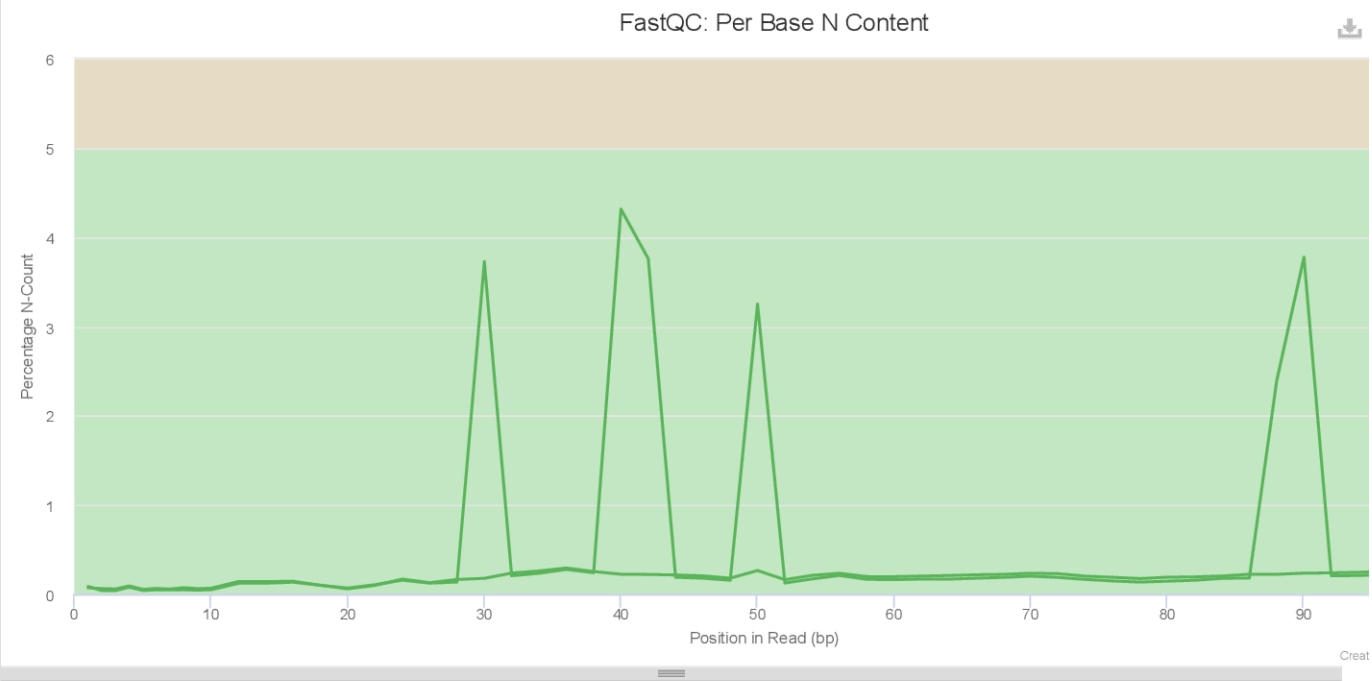


Per Base N Content

2

Help

The percentage of base calls at each position for which an N was called.



Sequence Length Distribution

2

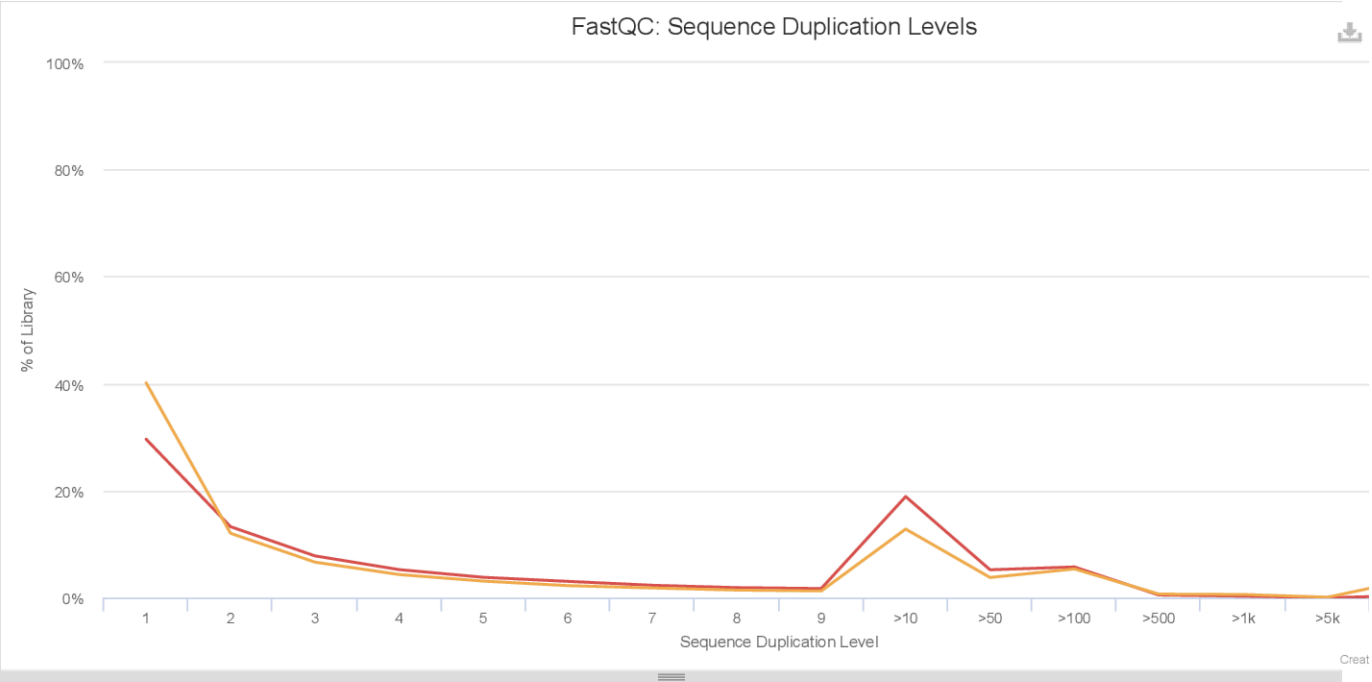
All samples have sequences of a single length (100bp).

Sequence Duplication Levels

1 1

Help

The relative level of duplication found for every sequence.

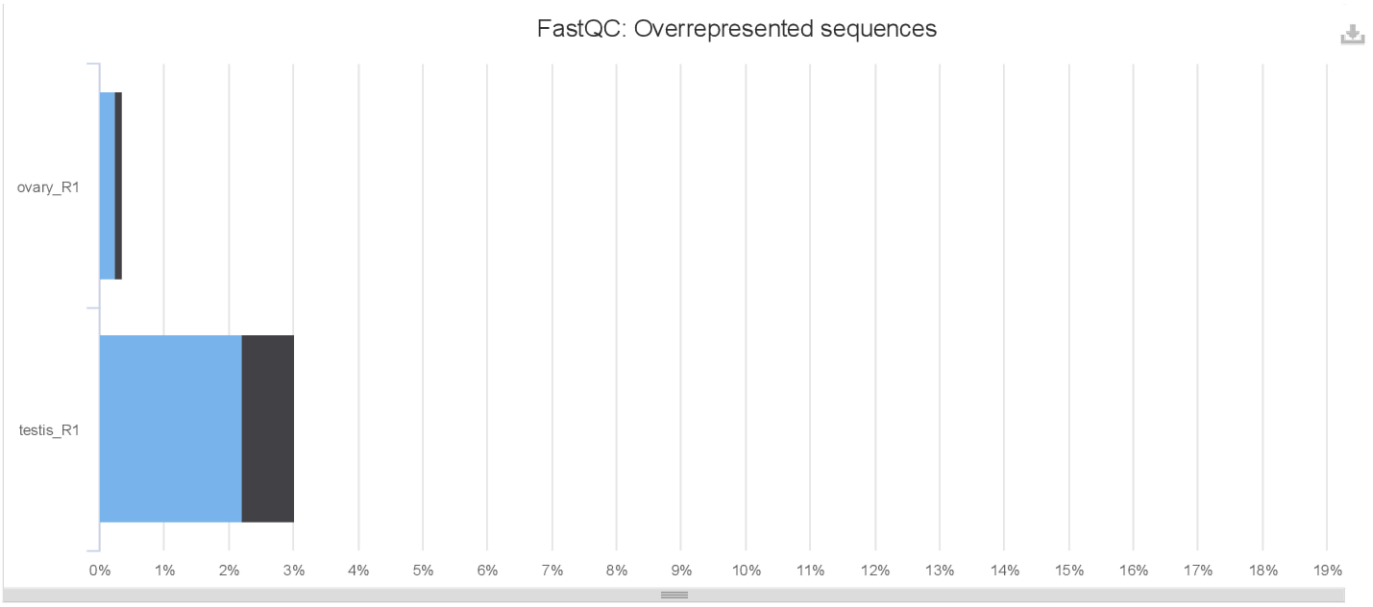


Overrepresented sequences

1 1

Help

The total amount of overrepresented sequences found in each library.

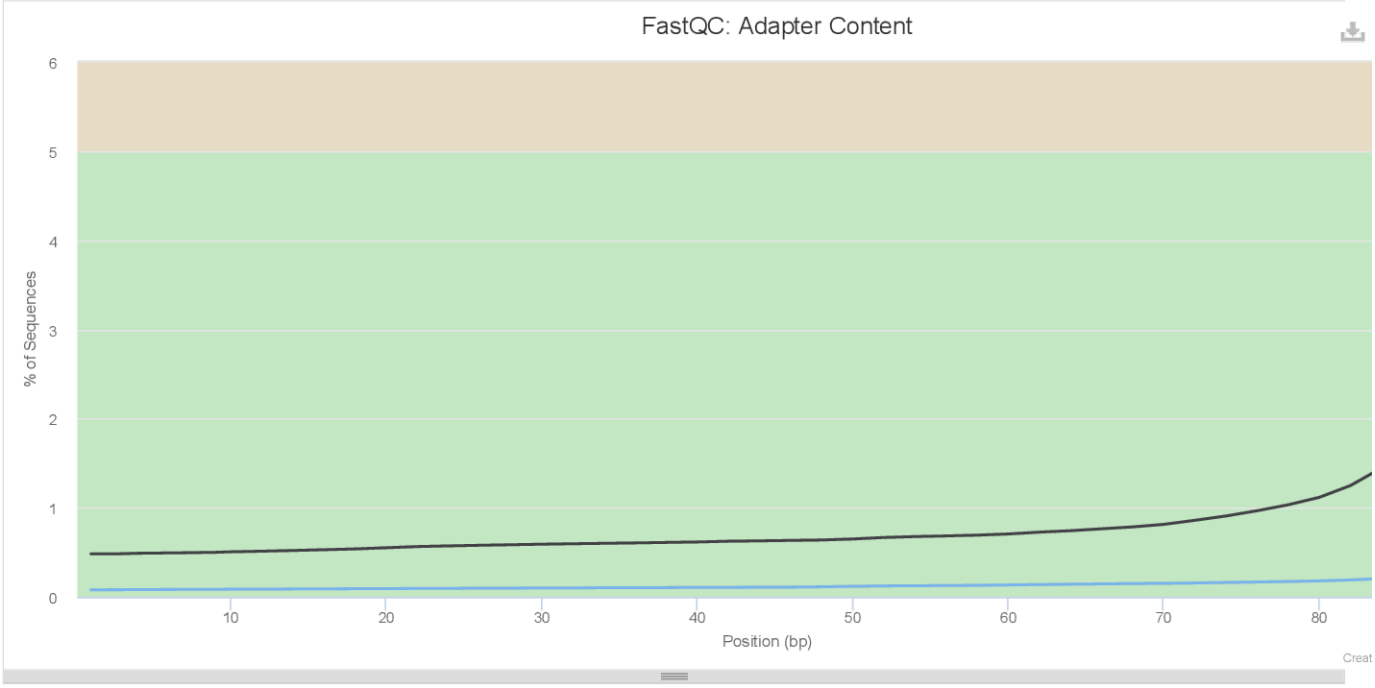


Adapter Content

2

Help

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

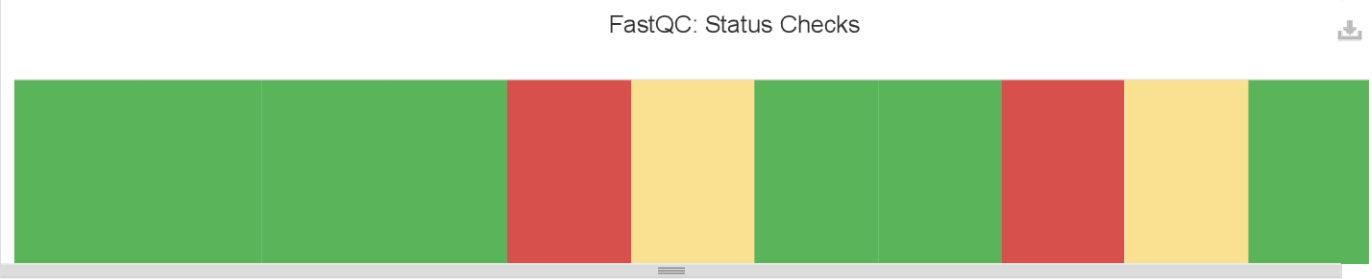


Status Checks

Help

Status for each FastQC section showing whether results seem entirely normal (green), slightly abnormal (orange) or very unusual (red).

Sort by highlight



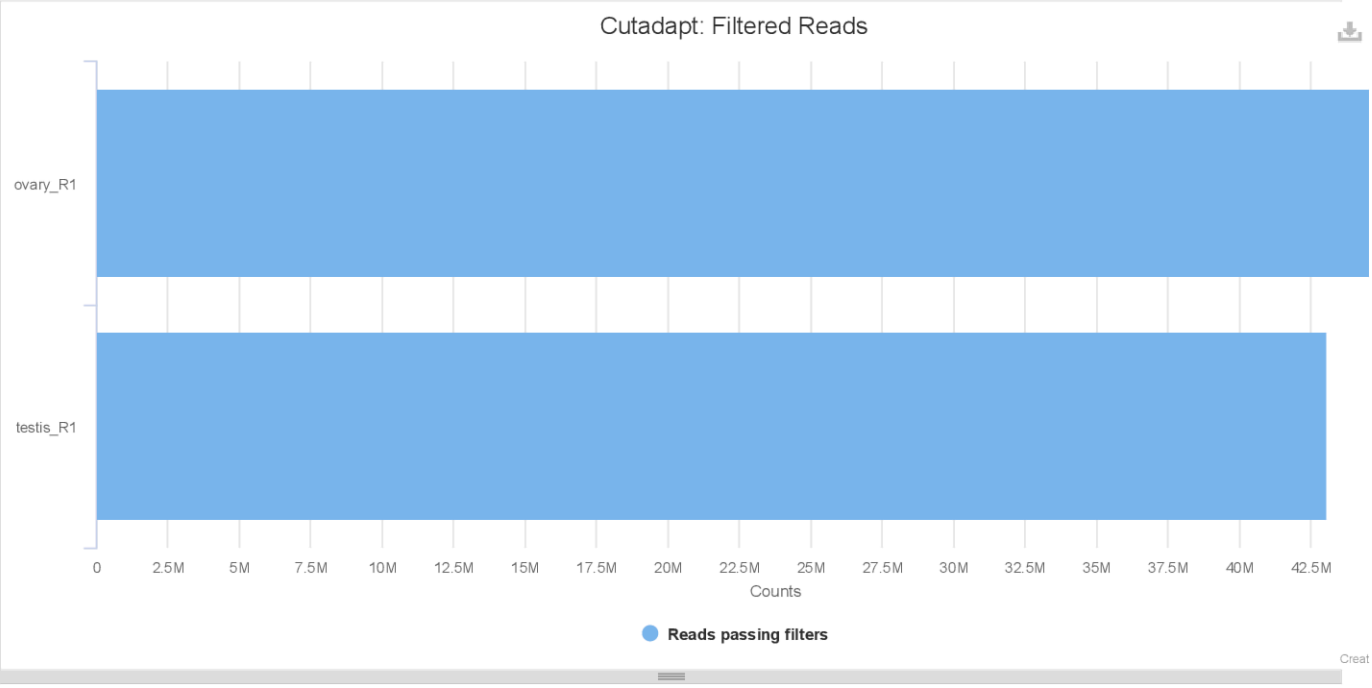
Cutadapt

Cutadapt is a tool to find and remove adapter sequences, primers, poly-Atails and other types of unwanted sequence from your high-throughput sequencing reads.

Filtered Reads

This plot shows the number of reads (SE) / pairs (PE) removed by Cutadapt.

Counts | Percentages

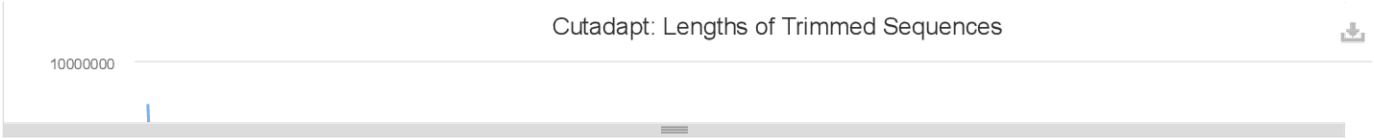


Trimmed Sequence Lengths

Help

This plot shows the number of reads with certain lengths of adapter trimmed.

Counts | Obs/Exp



FastQC (trimmed)

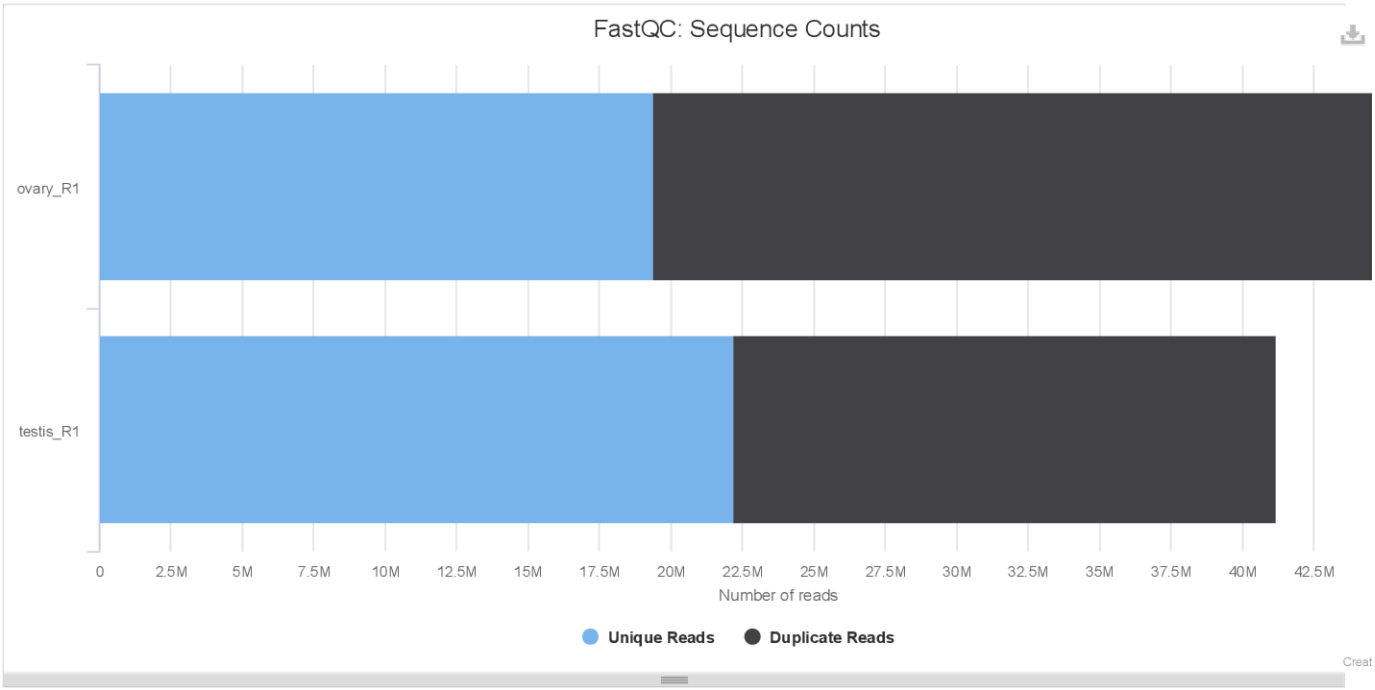
FastQC (trimmed) This section of the report shows FastQC results after adapter trimming.

Sequence Counts

Help

Sequence counts for each sample. Duplicate read counts are an estimate only.

Number of reads | Percentages

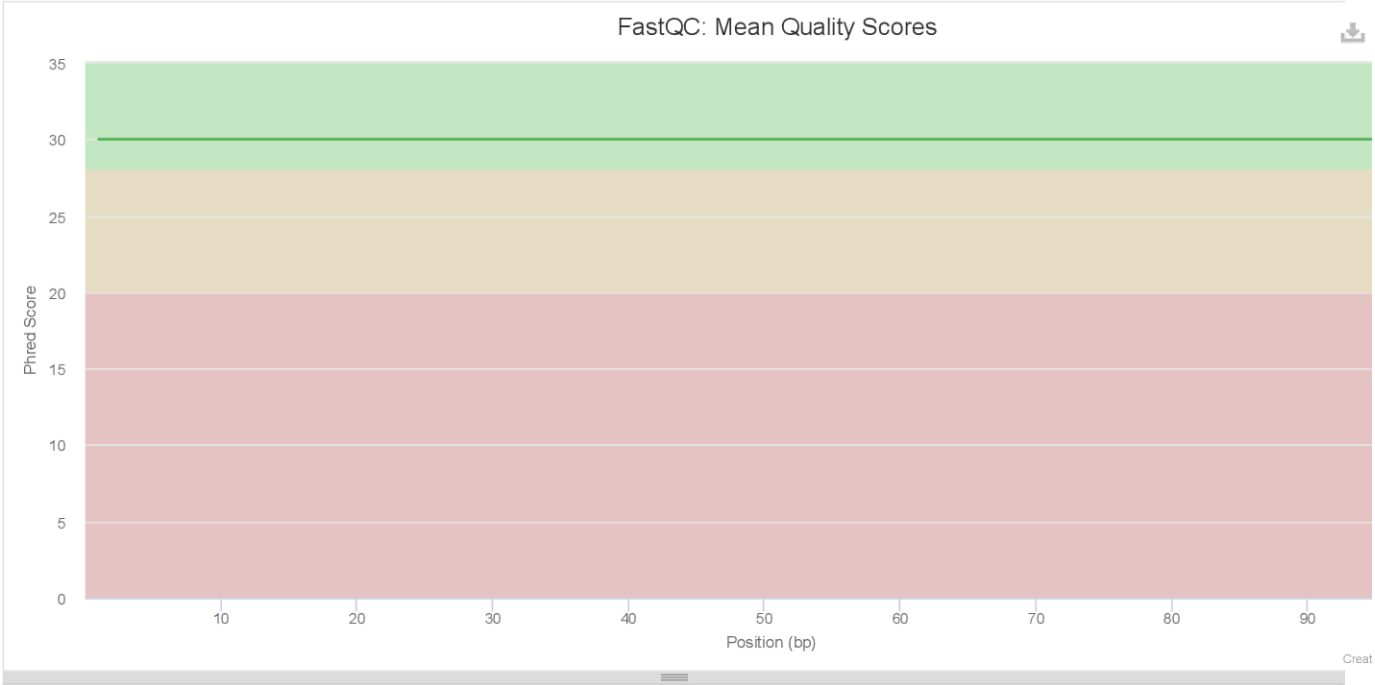


Sequence Quality Histograms

2

Help

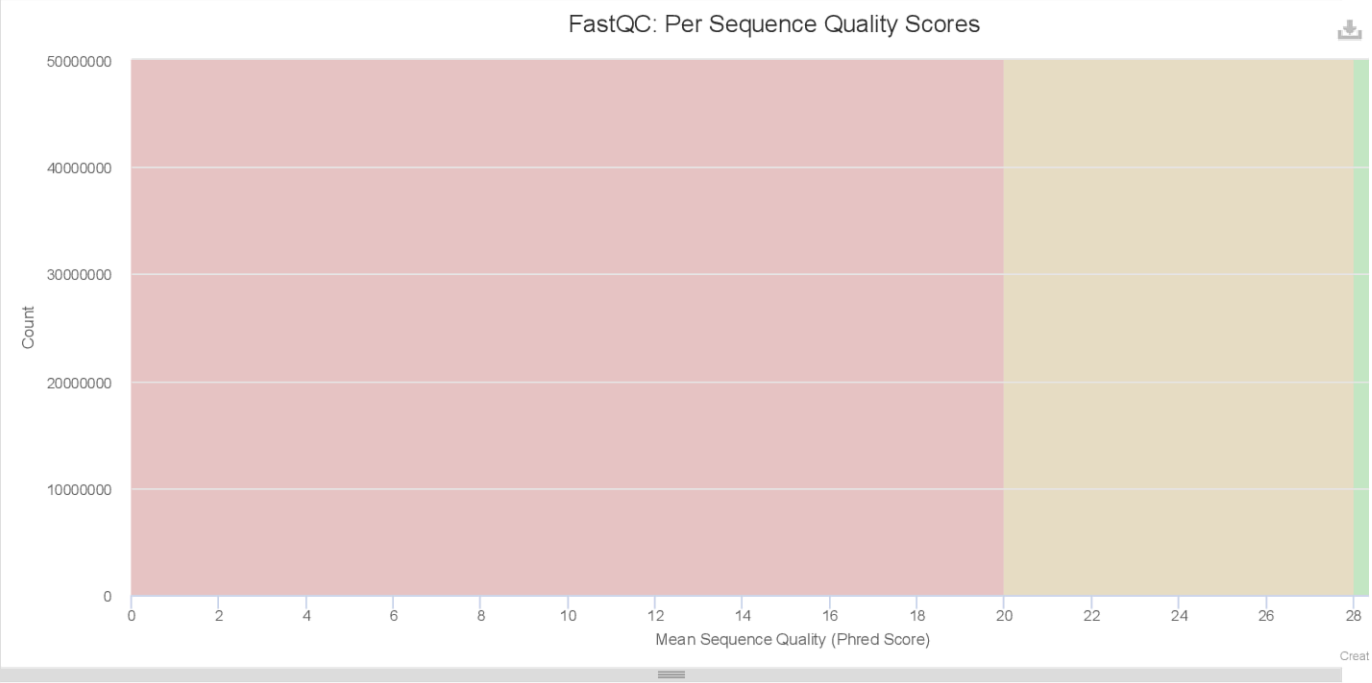
The mean quality value across each base position in the read.



Per Sequence Quality Scores 2

Help

The number of reads with average quality scores. Shows if a subset of reads has poor quality.



Per Base Sequence Content 2

Help

The proportion of each base position for which each of the four normal DNA bases has been called.

Click a sample row to see a line plot for that dataset.

Rollover for sample name

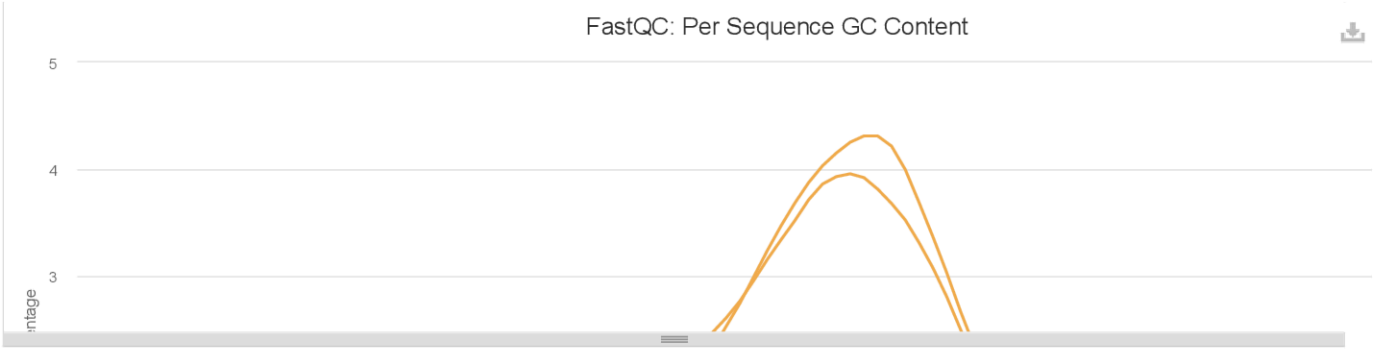
Position: - %T: - %C: - %A: - %G: - Export Plot

Per Sequence GC Content 2

Help

The average GC content of reads. Normal random library typically have a roughly normal distribution of GC content.

Percentages | Counts

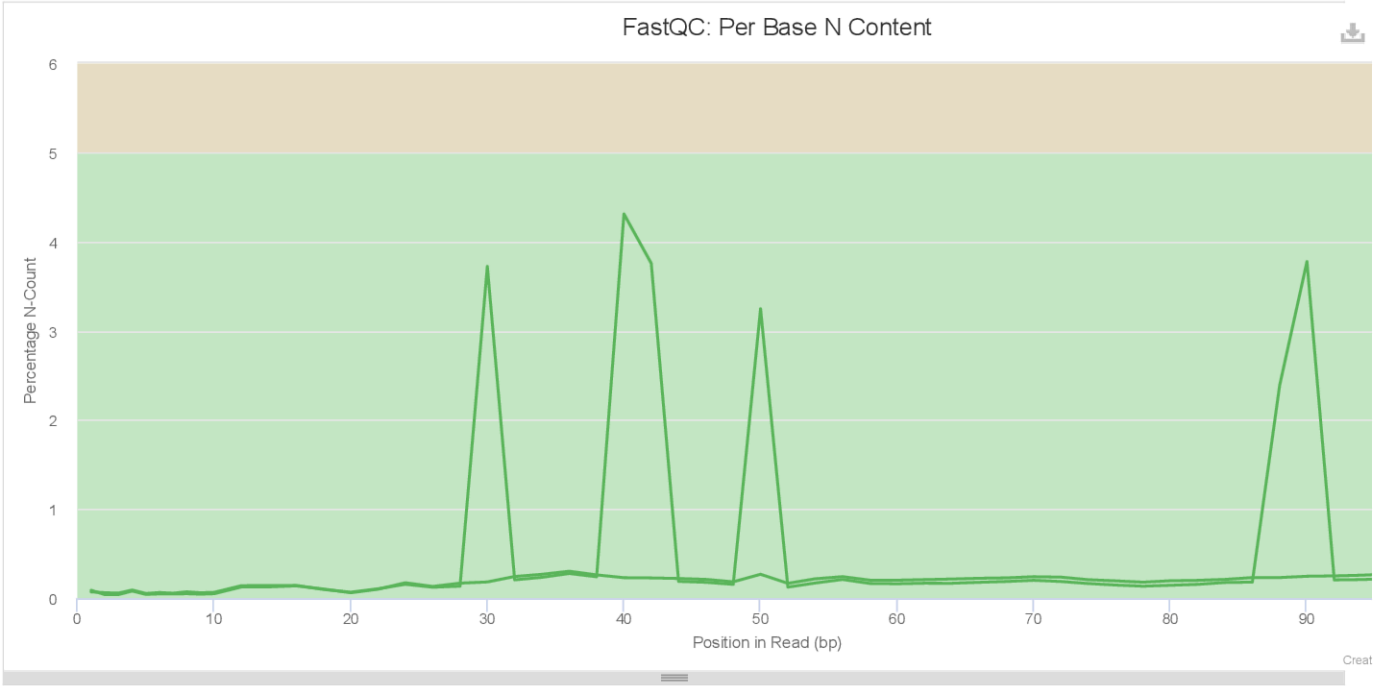


Per Base N Content

2

Help

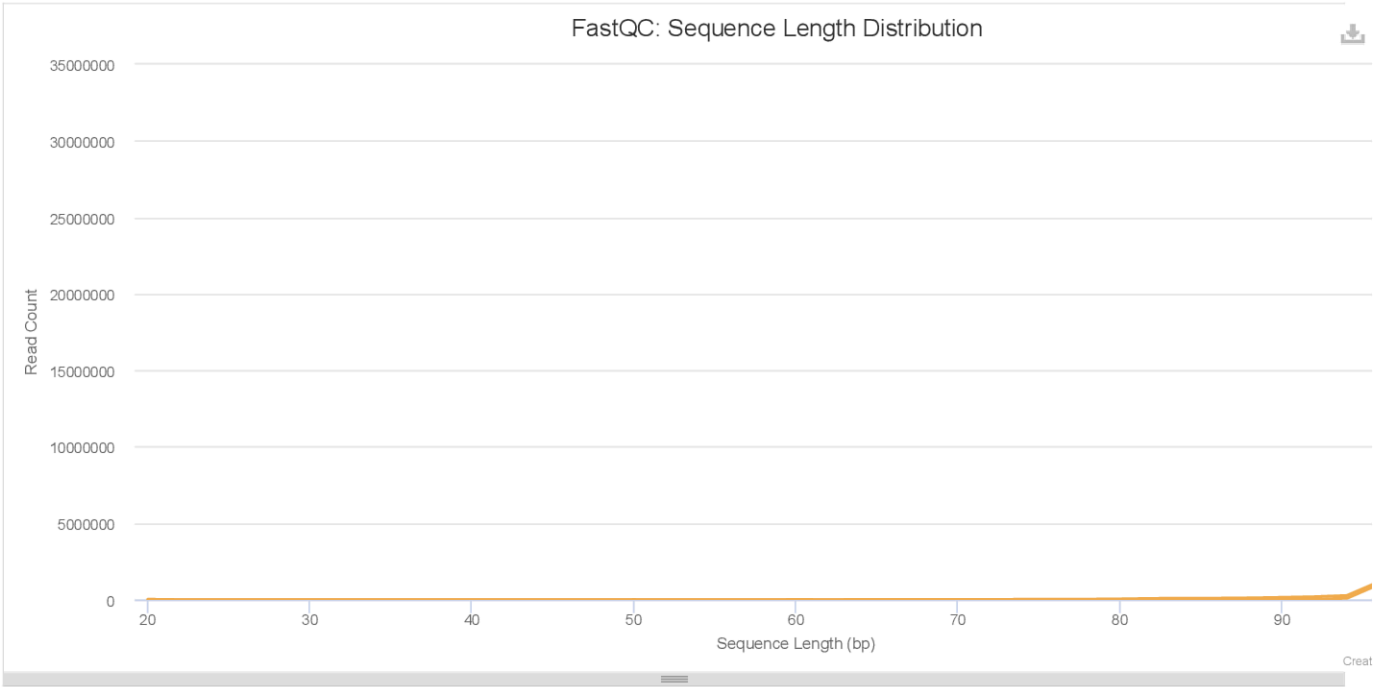
The percentage of base calls at each position for which an N was called.



Sequence Length Distribution

2

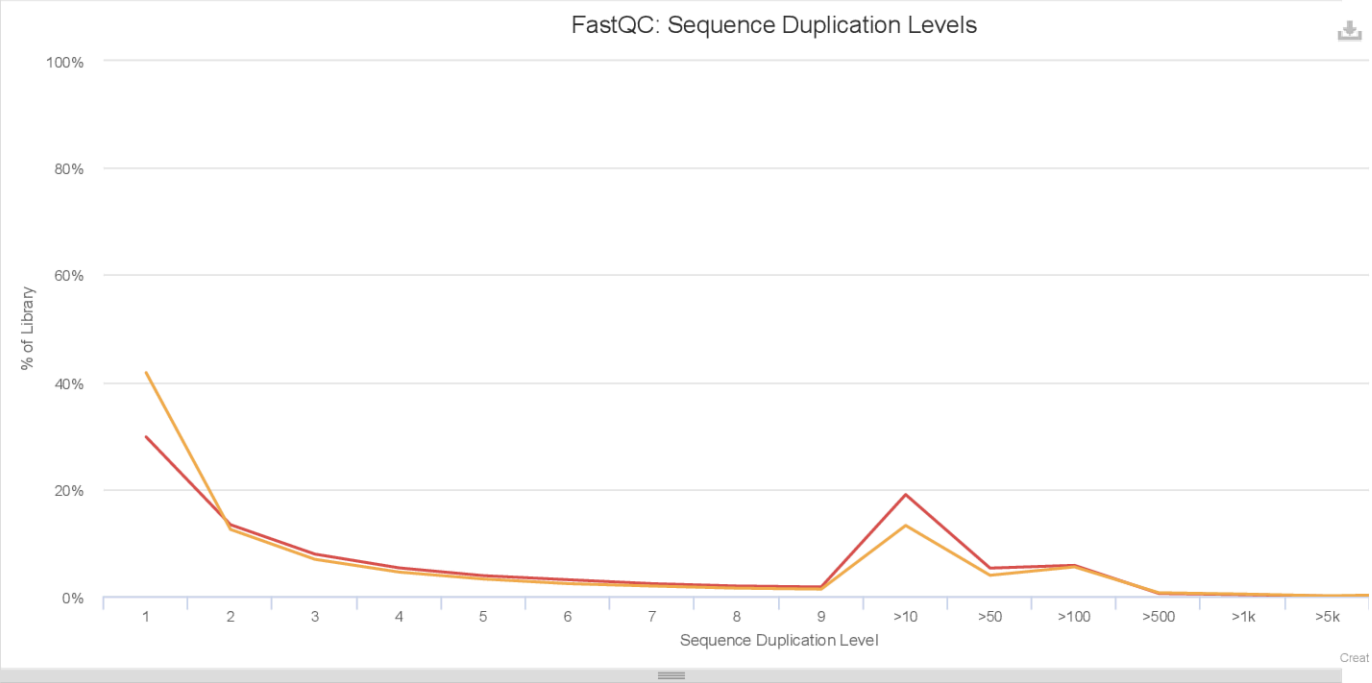
The distribution of fragment sizes (read lengths) found. See the FastQC help



Sequence Duplication Levels 1 1

Help

The relative level of duplication found for every sequence.



Overrepresented sequences 1 1

Help

The total amount of overrepresented sequences found in each library.

2 samples had less than 1% of reads made up of overrepresented sequences

Adapter Content 2

Help

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

No samples found with any adapter contamination > 0.1%

Status Checks

Help

Status for each FastQC section showing whether results seem entirely normal (green), slightly abnormal (orange) or very unusual (red).

Sort by highlight

nf-core/rnaseq Software Versions

are collected at run time from the software output.

bedtools
2.29.2
deseq2
1.28.0
dupradar
1.18.0
fastqc
0.11.9
nextflow
21.04.1
nf-core/rnaseq
3.0
picard
2.23.9
preseq
2.0.3
qualimap
2.2.2-dev
rsem
1.3.1
rseqc
3.0.1
samtools
1.10
stringtie
2.1.4
subread
2.0.1
trimgalore
0.6.6
ucsc
377

nf-core/rnaseq Workflow Summary

- this information is collected when the pipeline is started.

Core Nextflow options

revision
3.0
runName
marvelous_noyce
containerEngine
singularity
launchDir
/work/laurier/PROJET_NEXTFLOW/MORUE
workDir
/work/laurier/PROJET_NEXTFLOW/MORUE/work
projectDir
/home/laurier/.nextflow/assets/nf-core/rnaseq
userName
laurier
profile
genotoul
configFiles
/home/laurier/.nextflow/assets/nf-core/rnaseq/nextflow.config, /home/laurier/work/PROJET_NEXTFLOW/MORUE/sm_config.cfg

Input/output options

input
/home/laurier/work/PROJET_NEXTFLOW/MORUE/inputs.csv

Reference genome options

fasta
/home/laurier/work/PROJET_NEXTFLOW/MORUE/GENOME_REF/Gadus_morhua.gadMor3.0.dna.toplevel.fa

gtf
/home/laurier/work/PROJET_NEXTFLOW/MORUE/GENOME_REF/Gadus_morhua.gadMor3.0.110.gtf
save_reference
true
igenomes_ignore
true

Alignment options

aligner
star_rsem

Institutional config options

config_profile_description
The Genotoul cluster profile
config_profile_contact
support.bioinfo.genotoul@inra.fr
config_profile_url
http://bioinfo.genotoul.fr/

Max job request options

max_cpus
48
max_memory
120 GB
max_time
4d