seqcluster Documentation

Release 1.2.2

Lorena Pantano

August 23, 2016

Contents

Analysis of small RNA sequencing data. It detect unit of transcription over the genome, annotate them and create an HTML interactive report that helps to explore the data quickly.

Contents:

CHAPTER 1

Installation

1.1 Seqcluster

With bcbio installed

If you already have **'bcbio'_**, seqcluster comes with it. If you want the last development version:

```
/bcbio_anaconda_bin_path/seqcluster_install.py --upgrade
```

Binstar binary

install conda if you want an isolate env:

```
wget http://repo.continuum.io/miniconda/Miniconda-latest-Linux-x86_64.sh
bash Miniconda-latest-Linux-x86_64.sh -b -p ~/install/seqcluster/anaconda
```

You can install directly from binstar (only for linux):

```
~/install/seqcluster/anaconda/conda install seqcluster bcbio-nextgen -c bioconda
```

With that you will have everything you need for the python package. The last step is to add seqcluster to your PATH (see below).

Go to Tools dependecies below to continue with the installation.

Step by step

If you want to install step by step from a new conda environment:

```
~/install/seqcluster/anaconda/bin/conda install -c bioconda bcbio-nextgen
```

Link binary to any folder is already in your path:

```
ln -s ~/install/seqcluster/anaconda/bin/seqcluster* ~/install/seqcluster/linuxbrew/bin/.
```

Note: After installation is highly recommended to get the last updated version doing:

```
seqcluster_install.py --upgrade
```

1.2 Tools dependecies

For seqcluster command:

- bedtools

- samtools
- rnafold (for HTML report)

For some steps of a typical small RNA-seq pipeline (recommended to use directly **'bcbio'_**):

- STAR, bowtie
- fastqc
- cutadapt (install with `bioconda` using the same `python` env than seqcluster.

You will need to link the `cutadapt` binary to your `PATH`)

easy installation

Strongly recommended to use bcbio installation if you work with sequencing data. But if you want a minimal installation:

```
pip install fabric
seqcluster_install --upgrade
mkdir -p $PATH_TO_TOOLS/bin
seqcluster_install --tools $PATH_TO_TOOLS
```

After that you will need to add to your path: `export PATH=$PATH_TO_TOOLS/bin:$PATH`

1.3 Data

Easy way to install your small RNA seq data with cloudbiolinux. Seqcluster has snipped code to do that for you. Recommended to use **'bcbio'_** for the pipeline since will install everything you need in a single step `bcbio_nextgen.py upgrade -u development --tools --genomes hg19 --aligners bowtie`.

But If you want to run `seqcluster` step by step an example of hg19 human version it will be (another well annotated supported genome is mm10):

Download genome data:

```
seqcluster_install --data $PATH_TO_DATA --genomes hg19 --aligners bowtie2 --datatarget smallrna
```

If you want to install STAR indexes since gets kind of better results than bowtie2 (warning, 40GB memory RAM needed):

```
seqcluster_install --data $PATH_TO_DATA --genomes hg19 --aligners star
```

1.4 R package

Install isomiRs package for R using devtools:

```
devtools::install_github('lpantano/isomiRs')
```

To install all packages used by the Rmd report:

```
Rscript -e 'source(https://raw.githubusercontent.com/lpantano/seqcluster/master/scripts/install_libra
```

CHAPTER 2

Citation

Please if you use seqcluster make sure to cite the other tools are integrated here:

A non-biased framework for the annotation and classification of the non-miRNA small RNA transcriptome. Pantano L1, Estivill X, Martí E. Bioinformatics. 2011 Nov 15;27(22):3202-3. doi: 10.1093/bioinformatics/btr527. Epub 2011 Oct 5. PMID: 21976421

SeqBuster is a bioinformatic tool for the processing and analysis of small RNAs datasets, reveals ubiquitous miRNA modifications in human embryonic cells. Pantano L, Estivill X, Martí E. Nucleic Acids Res. 2010 Mar;38(5):e34. Epub 2009 Dec 11.

Quinlan AR and Hall IM, 2010. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 26, 6, pp. 841–842.

Dale RK, Pedersen BS, and Quinlan AR. Pybedtools: a flexible Python library for manipulating genomic datasets and annotations. Bioinformatics (2011). doi:10.1093/bioinformatics/btr539

Li H.*, Handsaker B.*, Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. and 1000 Genome Project Data Processing Subgroup (2009) The Sequence alignment/map (SAM) format and SAMtools. Bioinformatics, 25, 2078-9. [PMID: 19505943]

Li H A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics. 2011 Nov 1;27(21):2987-93. Epub 2011 Sep 8. [PMID: 21903627]

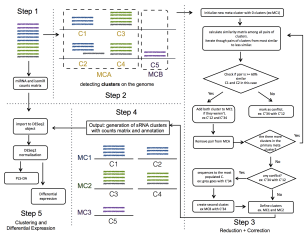
CHAPTER 3

Getting started

Best practices are implemented in a python framework.

3.1 clustering of small RNA sequences

seqcluster generates a list of clusters of small RNA sequences, their genome location, their annotation and the abundance in all the sample of the project



REMOVE ADAPTER

I am currently using `cutadapt`:

```
cutadapt --adapter=$ADAPTER --minimum-length=8 --untrimmed-output=sample1_notfound.fastq -o sample1_c
```

COLLAPSE READS

To reduce computational time, I recommend to collapse sequences, also it would help to apply filters based on abundances. Like removing sequences that appear only once.

```
seqcluster collapse -f sample1_clean.fastq -o collapse
```

Here I am only using sequences that had the adapter, meaning that for sure are small fragments.

PREPARE SAMPLES

```
seqcluster prepare -c file_w_samples -o res --minl 17 --minc 2 --maxl 45
```

the file_w_samples should have the following format:

```
lane1_sequence.txt_1_1_phred.fastq      cc1
lane1_sequence.txt_2_1_phred.fastq      cc2
lane2_sequence.txt_1_1_phred.fastq      cc3
lane2_sequence.txt_2_1_phred.fastq      cc4
```

two columns file, where the first column is the name of the file with the small RNA sequences for each sample, and the second column in the name of the sample.

The fastq files should be like this:

```
@seq_1_x11
CCCCGTTCCCCCCTCCTCC
+
QUALITY_LINE
@seq_2_x20
TGCGCAGTGGCAGTATCGTAGCCAATG
+
QUALITY_LINE
</pre>
```

Where _x[09] indicate the abundance of that sequence, and the middle number is the index of the sequence.

This script will generate: `seqs.fastq` and `seqs.ma`. * seqs.fastq: have unique sequences and unique ids * seqs.ma: is the abundance matrix of all unique sequences in all samples

ALIGNMENT

You should use an aligner to map seqs.fa to your genome. A possibility is bowtie or STAR. From here, we need a file in BAM format for the next step. VERY IMPORTANT: the BAM file should be sorted

```
bowtie -a --best --strata -m 5000 INDEX seqs.fastq -S | samtools view -Sbh /dev/stdin | samtools sort
```

or

```
STAR --genomeDir $star_index_folder --readFilesIn res/seqs.fastq --alignIntronMax 1  --outFilterMulti
```

CLUSTERING

```
seqcluster cluster -a res/Aligned.sortedByCoord.out.bam  -m res/seqs.ma -g $GTF_FILE  -o res/cluster
```

- *-a* is the SAM file generated after mapped with your tool, which input has been seqs.fa
- *-m* the previous seqs.fa
- *-b* annotation files in bed format (see below examples) [deprecated]
- *-g* annotation files in gtf format (see below examples) [recommended]
- *-i* genome fasta file used in the mapping step (only needed if -s active)
- *-o* output folder
- *-ref* genome fasta file. Needs `fai` file as well there. (i.e `hg19.fa`, `hg19.fa.fai`)
- *-d* create debug logging
- *-s* construction of putative precursor (NOT YET IMPLEMENTED)
- *–db* (optional) will create sqlite3 database with results that will be used to browse data with html web page (under development)

Example of a bed file for annotation (the fourth column should be the name of the feature):

```
chr1    157783  157886  snRNA   0       -
```

Strongly recommend gtf format. Bed annotation is deprecated. Go here to know how to download data from hg19 and mm10.

Example of a gtf file for annotation (the **third** column should be the name of the feature and the value after *gene name* attribute is the specific annotation):

```
chr1    source  miRNA      1       11503   .       +       .       gene name 'mir-102' ;
```

hint: scripts to generate human and mouse annotation are inside seqcluster/scripts folder.

OUTPUTS

- `counts.tsv`: count matrix that can be input of downstream analyses
- `size_counts.tsv`: size distribution of the small RNA by annotation group
- `seqcluster.json`: json file containing all information
- `log/run.log`: all messages at debug level
- `log/trace.log`: to keep trace of algorithm decisions

3.2 Interactive HTML Report

This will create html report using the following command assuming the output of *seqcluster cluster* is at *res*:

```
seqcluster report -j res/seqcluster.json -o report -r $GENONE_FASTA_PATH
```

where *$GENOME_FASTA_PATH* is the path to the genome fasta file used in the alignment.

Note: you can try our new visualization tool!

- `report/html/index.html`: table with all clusters and the annotation with sorting option
- `report/html/[0-9]/maps.html`: summary of the cluster with expression profile, annotation, and all sequences inside
- `report/html/[0-9]/maps.fa`: putative precursor

An example of the output is below:

3.3 Easy start with bcbio-nextgen.py

Note:If you already are using bcbio, visit bcbio to run the pipeline there.

To install the small RNA data:

```
bcbio_nextgen.py upgrade -u development --tools --datatarget smallrna
```

Options to run in a cluster

It uses ipython-cluster-helper to send jobs to nodes in the cluster

- *–parallel* should set to *ipython*
- *–scheduler* should be set to *sge,lsf,slurm*
- *–num-jobs* indicates how much jobs to launch. It will run samples independently. If you have 4 samples, and set this to 4, 4 jobs will be launch to the cluster
- *–queue* the queue to use
- *–resources* allows to set any special parameter for the cluster, such as, email in sge system: *M=my@email.com*

Read complete usability here: https://github.com/roryk/ipython-cluster-helper An examples in slurm system is:

```
--parallel ipython --scheduler slurm --num-jobs 4 --queue general
```

Output

- one folder for each analysys, and inside one per sample
- adapter: **clean.fastq* is the file after adapter removal, **clean_trimmed.fastq* is the collapse *clean.fastq*, **fragments.fastq* is file without adapter, **short.fastq* is file with reads < 16 nt.
- align: BAM file results from align *trimmed.fastq*
- mirbase: file with miRNA anotation and novel miRNA discovery with mirdeep2
- tRNA: analysis done with tdrmapper [citation needed]
- qc: **_fastqc.html* is the fastqc results from the uncollapse fastq file
- seqcluster: is the result of running seqcluster. See its documentation for further information.
- *report/srna-report.Rmd*: template to create a quick html report with exploration and differential expression analysis. See example here

CHAPTER 4

Outputs

4.1 seqcluster

- `counts.tsv`: count matrix that can be input of downstream analyses. *nloci* will be 0 always that the meta-cluster has been resolved successfully. For instance, it can happen that you got sequences you have a bunch of sequences mapping to hundreds of different places on the genome, then seqcluster doesn't resolve that, and put everything under the larger region covered by those sequences. So, mainly, 0 all are good rows. The *ann* column is just where the meta-clusters overlap with. It can happen that one name appears many times if different locations of the meta-cluster map to different copies of that feature. OR if the annotation file used had multiple lines for that.
- `read_stats.tsv`: number of reads for each sample after each step in the analysis. Meant to give a hint if we lose a lot of information or not.
- `size_counts.tsv`: size distribution of the small RNA by annotation group. (position, reads, cluster)
- `seqcluster.json`: json file containing all information. This file is used as the input of the report suit.
- `log/run.log`: all messages at debug level
- `log/trace.log`: to keep trace of algorithm decisions

4.2 Report

Beside the static HTML report that you can get using `report` subcommand, you can download this HTML. (watch the repository to get notifications of new releases.)

- Go inside `seqclusterViz` folder
- Open `reader.html`
- Upload the `seqcluster.db` file generated by `report` subcommand.
- Start browsing your data!

Meaning of different sections:

- Top-left table shows list of meta-clusters, user can filter by number ID or keywords.
- Top-right table shows positions where this meta-cluster has been detected.
- Expression profile along precursor: Lines are number of reads in that position of the precursor. It is sum of the log2 RPM of the expression for each sample.
- Table: raw counts for each sample and sequence. Only top 100 are shown.

- secondary structure: The region with more sequences inside meta-cluster is used to plot the secondary structure. Colors refers to abundance in each position. Darker means more abundance.

An example of the HTML code: _ ..examples

CHAPTER 5

Examples of small RNA analysis

5.1 miRQC data

About

mirRQC project

samples overview:

>> Universal Human miRNA reference RNA (Agilent Technologies, #750700), human brain total RNA (Life Technologies, #AM6050), human liver total RNA (Life Technologies, #AM7960) and MS2-phage RNA (Roche, #10165948001) were diluted to a platform-specific concentration. RNA integrity and purity were evaluated using the Experion automated gel electrophoresis system (Bio-Rad) and Nanodrop spectrophotometer. All RNA samples were of high quality (miRQC A: RNA quality index (RQI, scale from 0 to 10) = 9.0; miRQC B: RQI = 8.7; human liver RNA: RQI = 9.2) and high purity (data not shown). RNA was isolated from serum prepared from three healthy donors using the miRNeasy mini kit (Qiagen) according to the manufacturer's instructions, and RNA samples were pooled. Informed consent was obtained from all donors (Ghent University Ethical Committee). Different kits for isolation of serum RNA are available; addressing their impact was outside the scope of this work. Synthetic miRNA templates for let-7a-5p, let-7b-5p, let-7c, let-7d-5p, miR-302a-3p, miR-302b-3p, miR-302c-3p, miR-302d-3p, miR-133a and miR-10a-5p were synthesized by Integrated DNA Technologies and 5 phosphorylated. Synthetic let-7 and miR-302 miRNAs were spiked into MS2-phage RNA and total human liver RNA, respectively, at 5 × 106 copies/μg RNA. These samples do not contain endogenous miR-302 or let-7 miRNAs, which allowed unbiased analysis of cross-reactivity between the individual miR-302 and let-7 miRNAs measured by the platform and the different miR-302 and let-7 synthetic templates in a complex RNA background. Synthetic miRNA templates for miR-10a-5p, let-7a-5p, miR-302a-3p and miR-133a were spiked in human serum RNA at 6 × 103 copies per microliter of serum RNA or at 5-times higher, 2-times higher, 2-times lower and 5-times lower concentrations, respectively. All vendors received 10 μl of each serum RNA sample.

Commands

Data was download from GEO web with this script. The following 2 configs were used for the two sets: mirqc samples and non mirqc samples. Samples were analyzed with bcbio with the following commands

report

Report showing part of the output report of bcbio pipelines together with some validations are here.

CHAPTER 6

miRNA annotation

miRNA annotation is running inside bcbio small RNAseq pipeline together with other tools to do a complete small RNA analysis.

For some comparison with other tools go here.

You can run samples after processing the reads as shown below. Currently there are two version: JAVA and PYTHON/C.

Naming

It is a working process, but since 10-21-2015 isomiR naming has changed to:

- Changes at 5' end: `0/NA` means no modification. `UPPER CASE LETTER` means nucleotide insertions (sequence starts before miRBase mature position). `LOWWER CASE LETTER` means nucleotide deletions (sequence starts after miRBase mature position).
- Changes at 3' end: `0/NA` means no modification. `UPPER CASE LETTER` means nucleotide insertions (sequence ends after miRBase mature position). `LOWWER CASE LETTER` means nucleotide deletions (sequence ends before miRBase mature position).
- Additions at 3' end: `0/NA` means no modification. `UPPER CASE LETTER` means addition at the end. Note these nucleotides don't match the precursor. So they are post-transcriptional modification.
- Nucleotide substitution: `NUMBER|NUCLEOTIDE_ISOMIR|NUCLEOTIDE_REFERENCE` means at the position giving by the number the nucleotide in the sequence has substituted the nucleotide in the reference. This, as well, is a post-transcriptional modification.

There are two different outputs right now: 1) tab limited format, where each column will refer to the previous 4 points, or 2) a merged format, where these 4 points (mirna, substitution, addition, 5 trimming, 3 trimming) are separated by `:`. For instance: `hsa-let-7a-5p:0:0:GT:t` means `hsa-let-7a-5p` has a 5' trimming event (tarts 2 nts before the reference miRNA) and a 3' trimming event (ends 1 nt before the reference miRNA).

6.1 Processing of reads

REMOVE ADAPTER

I am currently using `cutadapt`.

```
cutadapt --adapter=$ADAPTER --minimum-length=8 --untrimmed-output=sample1_notfound.fastq -o sample1_
```

COLLAPSE READS

To reduce computational time, I recommend to collapse sequences, also it would help to apply filters based on abundances. Like removing sequences that appear only once.

```
seqcluster collapse -f sample1_clean.fastq -o collapse
```

Here I am only using sequences that had the adapter, meaning that for sure are small fragments. The output will be named as `sample1_clean_trimmed.fastq`

6.2 Prepare databases

For human or mouse, follows this instruction to download easily miRBase files. For other species you only need hairpin.fa and miRNA.str from miRBase site. **Highly recommended to filer hairpin.hsa to contain only the desired species.**

6.3 miRNA/isomiR annotation with JAVA

MIRALIGNER

Download the tool from miraligner repository.

Download the mirbase files (hairpin and miRNA) from the ftp and save it to *DB* folder.

You can map the miRNAs with.

```
java -jar miraligner.jar -sub 1 -trim 3 -add 3 -s hsa -i sample1_clean_trimmed.fastq -db DB  -o outpu
```

Cite

SeqBuster is a bioinformatic tool for the processing and analysis of small RNAs datasets, reveals ubiquitous miRNA modifications in human embryonic cells. Pantano L, Estivill X, Martí E. *Nucleic Acids Res. 2010 Mar;38(5):e34. Epub 2009 Dec 11.*

6.4 miRNA/isomiRs annotation with python

A new function to annotate miRNA/isomiR sequences using BAM files aligned to miRBase precursors or fastq files to align from scratch has been added to `seqcluster`:

```
seqcluster seqbuster --out results --hairpin hairpin.fa --mirna miRNA.str --species hsa input_file.fa
```

If the input file is a BAM file, seqcluster will parse it to produce miRNA annotation, including isomiRs. If the input is FASTQ/FASTA file, seqcluster will map with the new C implementation of miraligner and annotate miRNAs and isomiRs as before.

Multiple files can be given to analyze all of them serially. Files inside the output folder are:

- raw mirna annotation to all posible mirnas (`*.premirna`)
- count file for miRNAs (`counts_mirna.tsv`)
- count file for isomiRs (`counts.tsv`)

NOTE: Check comparison of multiple tools for miRNA annotation.

6.5 Post-analysis with R

Use the outputs to do differential expression, clustering and descriptive analysis with this package: isomiRs

6.6 Manual of miraligner(JAVA)

options

Add `-freq` if you have your fasta/fastq file with this format and you want a third column with the frequency (normally value after x character):

```
>seq_1_x4
CACCGCTGTCGGGGAACCGCGCCAATTT
```

Add `-pre` if you want also sequences that map to the precursor but outside the mature miRNA

- Parameter *-sub*: mismatches allowed (0/1)
- Parameter *-trim*: nucleotides allowed for trimming (max 3)
- Parameter *-add*: nucleotides allowed for addition (max 3)
- Parameter *-s*: species (3 letter, human=>hsa)
- Parameter *-i*: fasta file
- Parameter *-db*: folder where miRBase files are(one copy at miraligner-1.0/DB folder)
- Parameter *-o*: prefix for the output files
- Parameter *-freq*: add frequency of the sequence to the output (just where input is fasta file with name matching this patter: >seq_3_x67)
- Parameter *-pre*: add sequences mapping to precursors as well

input

A fasta/fastq file reads:

```
>seq
CACCGCTGTCGGGGAACCGCGCCAATTT
```

or tabular file with counts information:

CACCGCTGTCGGGGAACCGCGCCAATTT 45

output

Track file *.mirna.opt: information about the process

Non mapped sequences will be on *.nomap

Header of the *.mirna.out file:

- seq: sequence
- freq/name: depending on the input this column contains counts (tabular input file) or name (fasta file)
- mir: miRNA name
- start: start of the sequence at the precursor
- end: end of the sequence at the precursor
- mism: nucleotide substitution position | nucleotide at sequence | nucleotide at precursor
- addition: nucleotides at 3 end added:

```
    precursor         => cctgtggttagctggttgcatatcc
    annotated miRNA   =>   TGTGGTTAGCTGGTTGCATAT
    sequence add:  TT =>   TGTGGTTAGCTGGTTGCATATTT
```

- tr5: nucleotides at 5 end different from the annonated sequence in miRBase:

```
precursor             => cctgtggttagctggttgcatatcc
annotated miRNA   =>   TGTGGTTAGCTGGTTGCATAT
sequence tr5:  CC => CCTGTGGTTAGCTGGTTGCATAT
sequence tr5:  tg =>     TGGTTAGCTGGTTGCATAT
```

- tr3: nucleotides at 3 end different from the annotated sequence in miRBase:

```
precursor         => cctgtggttagctggttgcatatcc
annotated miRNA   =>   TGTGGTTAGCTGGTTGCATAT
sequence tr3: cc  =>   TGTGGTTAGCTGGTTGCATATCC
sequence tr3: AT  =>   TGTGGTTAGCTGGTTGCAT
```

- s5: offset nucleotides at the begining of the annotated miRNAs:

```
precursor         => agcctgtggttagctggttgcatatcc
annotated miRNA   =>     TGTGGTTAGCTGGTTGCATAT
s5                => AGCCTGTG
```

- s3:offset nucleotides at the ending of the annotated miRNAs:

```
precursor         =>  cctgtggttagctggttgcatatccgc
annotated miRNA   =>    TGTGGTTAGCTGGTTGCATAT
s3                =>                      ATATCCGC
```

- type: mapped on precursor or miRNA sequences
- ambiguity: number of different detected precursors

Example:

```
seq                 miRNA           start   end     mism    tr5     tr3     add     s5      s3      I
TGGCTCAGTTCAGCAGGACC    hsa-mir-24-2    50      67      0       qCC     0       0       0       0
ACTGCCCTAAGTGCTCCTTCTG  hsa-miR-18a*    47      68      0       0       0       tG      ATCTACTG
```

CHAPTER 7

Collapse fastq(.gz) files

Definition

Normally quality values are lost in small RNA-seq pipelines due to collapsing after adapter recognition. This option allow to collapse reads after adapter removal with `cutadapt` or any other tool. This way the mapping can use quality values, allowing to map using `bwa` for instance, or any other alignment tool that doesn't support FASTA files.

Methods

The new quality values are the average of each of the sequence collapse.

Example

```
seqcluster collapse -f sample_trimmed.fastq -o collapse
```

- `-f` is the fastq(.gz) file
- `-o` the folder where the outout will be created. A new FASTQ file, where the name stand for:

```
    @seq_[0-9]_x[0-9]
```

The number right after `_x` means the abundance of this sequence in the sample

CHAPTER 8

Handling multi-mapped reads

Definition

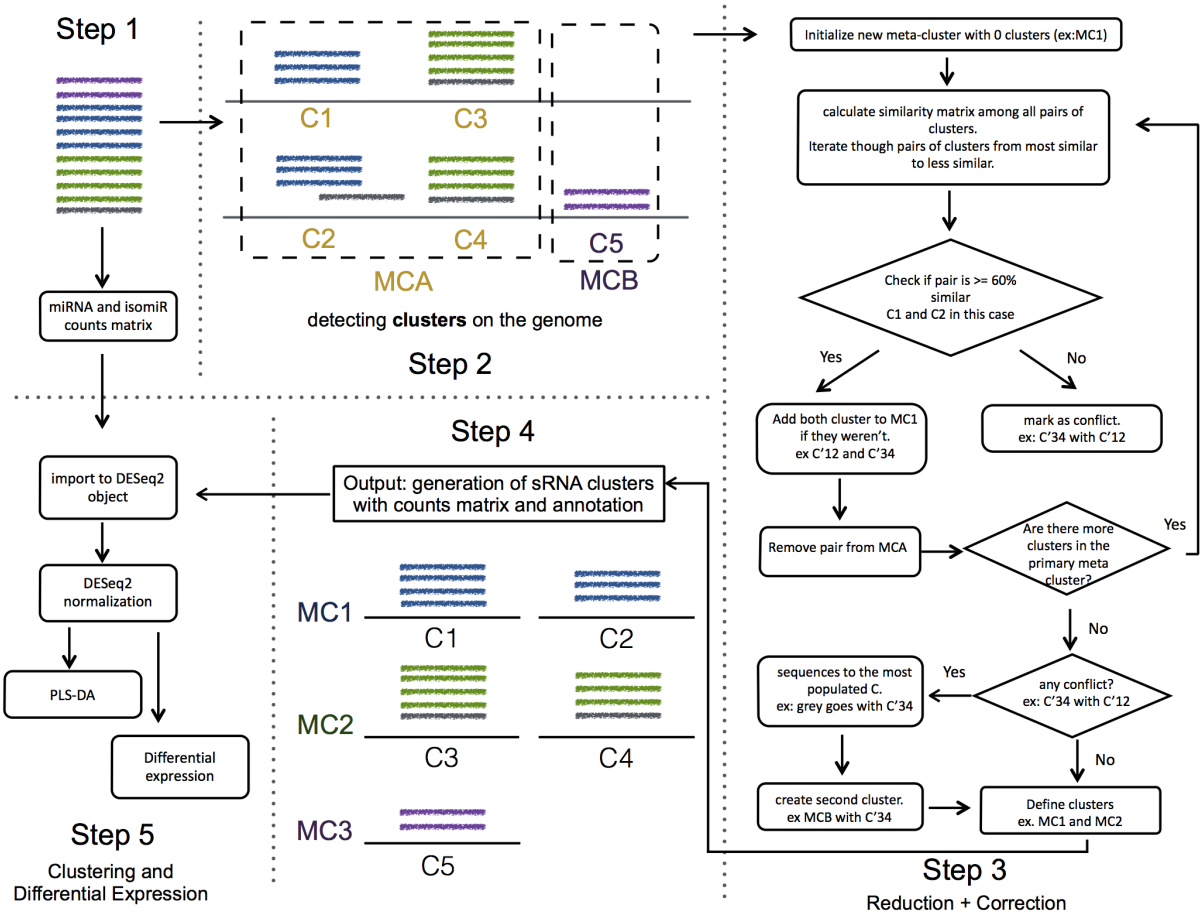
multi-mapped reads are the sequences that map more than one time on the genome, for instance, because there are multiple copies of a gene, like happens with tRNA precursors

Consequence

Many pipelines ignores these sequences as defaults, what means that you are losing at leas 20-30% of the data. In this case is difficult to decide where these sequences come from and currently there are three strategies:

- ignore them
- count as many times as they appear: for instance, if a sequences map twice, just count it two times in the two loci. This will due an over-representation of the loci abundances, and actually is against the assumption of all packages that perform differential expression in count data.
- weight them: divide the total count by the number of places it maps. In the previous example, each loci would get 1/2 * count. This produces weird dispersion values for packages that fit this value as part of the model.

Our implementation



We try to decide the origin of these sequences. The most common scenario is that a group of sequences map two three different regions, probably due to multi-copies on the genome of the precursor.

We introduce two options:

- most-voting strategy: In this case, we just count once all sequences, and we output this like one unit of transcription with multiple regions. This is the option by default.
- bayes inference: we give the same prior probability to all locations, and use the number of sequences starting in the same position than the one we are trying to predict its location as P(B|A). With this we calculate the posterior that will be used to get the proportion of counts to the different locations. We apply the code from the book: "Think Bayes" (Allen B. Downey). This is still under development. To activate this option, the user just needs to add *–method babes*

The main advantage of this, it is that it can be the input of any downstream analysis that is applied to RNA-seq, like DESeq, edgeR ... As well, there is less noise, because there is only one output coming from here, not three.

CHAPTER 9

Tools for downstream analysis

9.1 Web-servers

TFmiR: disease-specific miRNA/transcription factor co-regulatory networks v1.2. It uses results from UP/DOWN regulated miRNA/Genes and allows to focus in only one disease to create different type of relationships between miRNA/TF/Gene. Easy to use. Probably need to filter the output sometime due to the big networks that can result from an analysis.

Diana-TarBase v7.0: Database for validated miRNA targets. Many filter options. Good for small candidate miRNAs set studies.

StarScan: Database to browse the targets of miRNAs from degradome data. It has a fancy interface, and many species and data from GEO.

miRtex gives targets from literature. Good for finding validated targets to help discussion in papers or further functional experiment based on new hypothesis.

piRBase: Database for piRNA annotation and function. Published last year, for now the best I can find out there.

chimira: Web tool to analyze isomiR. It gives you a quick idea of you samples.

MicroCosm: MiRNA target database. Updated and download option.

IsomiR Bank: isomiR database from many species and tissues. For single queries is useful.

9.2 Command-lines

miRVaS : tools to predict the functional changed due to nt changes in the miRNA sequence.

CHAPTER 10

Relevant papers about isomiRs and other novel small RNAs with functional relevance

10.1 IsomiRs

miR-142-3p isomiR: "We furthermore demonstrate that miRNA 5-end variation leads to differential targeting and can thus broaden the target range of miRNAs."

A highly expressed miR-101 isomiR is a functional silencing small RNA.

A challenge for miRNA: multiple isomiRs in miRNAomics.

miR-183-5p isomiR changes in breast cancer. Validated target regulation of new genes different from the reference miRNA.

A comprehensive survey of 3' animal miRNA modification events and a possible role for 3' adenylation in modulating miRNA targeting effectiveness.

PAPD5-mediated 3 adenylation and subsequent degradation of miR-21 is disrupted in proliferative disease.

High-resolution analysis of the human retina miRNome reveals isomiR variations and novel microRNAs.

10.2 General

A novel piRNA mechanism in regulating gene expression in highly differentiated somatic cells.

Differential and coherent processing patterns from small RNAs to detect changes in profiles of processing small RNAs.

10.3 Targets

Identification of factors involved in target RNA-directed microRNA degradation.

10.4 Techonolgy

miRQC: work studying the accuracy and specificity of different technologies to detect miRNAs.

Important features affecting the detection of small RNA biomarkers: How the sample can affect the detection of biomarkers (like RIN value, concentration, ...)

Comparison of alignment and normalization . I will take the message that TMM and DESeq/2 normalization are the best to avoid strong bias if we consider to have a small proportion of DE miRNAs. For the alignments, here you have another comparison for miRNAs annotation: https://rawgit.com/lpantano/tools-mixer/master/mirna/mirannotation/stats.html

review of tools for detect miRNA-disease network.

review of tools for miRNA de-novo and interaction analysis

BIG meeting on Dec,3 2015: bcbio-srnaseq-BIG-20151203.pdf

CHAPTER 11

Documentation

seqcluster.prepare_data.**_create_matrix_uniq_seq**(*sample_l*, *seq_l*, *maout*, *out*, *min_shared*)

create matrix counts for each different sequence in all the fasta files

Parameters

- **sample_l** – list_s is the output of _read_fasta_files
- **seq_l** – seq_s is the output of _read_fasta_files
- **maout** – is a file handler to write the matrix count information
- **out** – is a file handle to write the fasta file with unique sequences

Returns Null

seqcluster.prepare_data.**_read_fasta_files**(*f*, *args*)

read fasta files of each sample and generate a seq_obj with the information of each unique sequence in each sample

Parameters **f** – file containing the path for each fasta file and

the name of the sample. Two column format with *tab* as field separator

Returns

- seq_l: is a list of seq_obj objects, containing the information of each sequence
- sample_l: is a list with the name of the samples (column two of the config file)

seqcluster.prepare_data.**_read_fastq_files**(*f*, *args*)

read fasta files of each sample and generate a seq_obj with the information of each unique sequence in each sample

Parameters **f** – file containing the path for each fasta file and

the name of the sample. Two column format with *tab* as field separator

Returns

- seq_l: is a list of seq_obj objects, containing the information of each sequence
- sample_l: is a list with the name of the samples (column two of the config file)

seqcluster.prepare_data.**prepare**(*args*)

Read all seq.fa files and create a matrix and unique fasta files. The information is

Parameters

- **args** – options parsed from command line

- **con** – logging messages going to console
- **log** – logging messages going to console and file

Returns files - matrix and fasta files that should be used with and aligner (as bowtie) and run *seqcluster cluster*

CHAPTER 12

Classes

class seqcluster.libs.classes.**sequence_unique**(*idx*, *seq*)
Object to store the sequence information like: **counts**, **sequence**, **id**

add_exp(*gr*, *exp*)
Function to add the counts for each sample

Parameters

- **gr** – name of the sample
- **exp** – counts of sample **gr**

Returns dict with key,values equally to name,counts.

class seqcluster.libs.classes.**cluster_info_obj**(*clus_obj*, *clus_id*, *loci_obj*, *seq_obj*)
Object containing information about clusters(clus_obj), positions(positions) and sequences(sequences)

class seqcluster.libs.classes.**sequence**(*seq_id*, *seq=None*, *freq=None*)
Object with information about sequences, counts, size, position, id and score

add_pos(*pos_id*, *pos*)

set_freq(*freq*)

set_seq(*seq*)

total()

class seqcluster.libs.classes.**position**(*idl*, *chr*, *start*, *end*, *strand*)
Object with information about position: chr,start,end,strand as well, with annotation information throuhg dbannotation object

add_db(*db*, *ndb*)

list()

class seqcluster.libs.classes.**annotation**(*db*, *name*, *strand*, *to5*, *to3*)
Object with information about annotation: database, name of the feature, strand, distance to 5' end, distance to 3' end

class seqcluster.libs.classes.**dbannotation**(*na*)
Object with information about annotation: containg one dict that store all features for each database type

add_db_ann(*ida*, *ndba*)

class seqcluster.libs.classes.**cluster**(*id*)
Object with cluster information. This is the main object.

add_id_member(*ids*, *idl*)

get_freq(*seqL*, *force=False*)

normalize(*seq*, *factor*)

set_freq(*seqL*)

set_ref(*r*)

update(*id=None*)

class seqcluster.libs.classes.**bedaligned**(*l*)
Object that has the bed format attributes

class seqcluster.libs.classes.**mergealigned**(*l*)
Object that has bed format after merge sequence positions

Visit GitHub code

I am in the process to document all classes and methods

CHAPTER 13

Indices and tables

- genindex
- modindex
- search

S

Symbols

A

B

C

D

G

L

M

N

P

S

T

U