Requirement specification “Java Metagenomik”

Dr. Colin Davenport
davenport.colin@mh-hannover.de
27.9.2010

Motivation and goals

The advent of cheap DNA sequence reads is accelerating discovery in medicine, molecular biology, genetics and genomics. Sequencing machines currently produce millions of short reads (30-1000+ bp), text file sizes 10 Mb – 10+GB.

These reads can be used in metagenomics, that is, to identify bacterial species based on their DNA in a sample of many different bacteria, and in transcriptomics, to find the genes which are used (expressed) by one single bacterium. The applications are different in terms of the scale of the data which needs to be stored, with metagenomics involving many and transcriptomics just one genome.

The large datasets produced hereby require extensive IT skills to analyse. The program described here, perhaps called Metatie2 (other suggestions welcome), should make analysis easier for non-technically orientated biologists. A perl script pipeline named Metatie, supplied in “datenpaket1.zip”, which runs the read alignment program bowtie, a self-written Java parser, the statistical language R for visualisation, and the linux program wc has been programmed. This should be rewritten in Java, without rewriting the complex external aligner (bowtie). Goal is primarily to visualise and analyse large datasets from SAM/BAM files. Further goals are to allow running of alignments in external alignment programs such as bowtie and others, which generate
Datenaustausch

http://genomics1.mh-hannover.de/tmp/exchange/

SAM/BAM files.

Sam format is very important for the project. It's binary counterpart is BAM. The Samtools software allows you to work with this format and analyse results.
http://samtools.sourceforge.net/SAM1.pdf

Picard is a project which provides a java tool and also a library for reading SAM and the binary BAM files.
http://picard.sourceforge.net/index.shtml

Open source sequence read visualisation projects

Integrated Genome Browser IGB (efficient enough for large datasets?)
IGB - http://www.bioviz.org/igb/

Geneious
http://www.geneious.com/
Attractive, but a commercial product – free trials are possible. A lot easier to use than the free academic software.

Integrative Genomics Viewer IGV
IGV - http://www.broadinstitute.org/igv/

Apollo -open source, quite simple visualisation though
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2705230/?tool=pubmed

Tablet
http://bioinf.scri.ac.uk/tablet/

MagicViewer

NGSView
http://ngsview.sourceforge.net/

SeqMonk

Functional requirements

The following functional requirements have been identified to date. These are prioritised numerically according to their importance, from 1 (high priority) to 5 (low priority).

Requirement: Good looking graphs
Priority: 1
Graphs must be optically pleasing. Options must be present for background, text size.
Example features:

- Nice readable text, perhaps 2-3 text options.
- Text size can be changed
- Background colour can be changed
- overlapping reads are shifted up so as not to overlap
- information window about read clicked on (sequence of read, start, stop)
- reads in + or – directions coloured differently
-

Requirement: Graphs as summary histogram of all species
Priority: 1
Initially the number of reads to each bacterial taxon are displayed as a summary graph, without read positions.
Example features:

- bacteria taxon names shown
- summed read hits shown
- summed reads in left hand text field

Requirement: Graphs as histogram of reads aligned to one selected species
Priority: 1
When a bacterial name is clicked upon the reads aligned to that genome are displayed as a bar graph.
Example features:

- x as genome string position, y as number of reads.
- Read positions are shown.
- Reads can be displayed as a horizontal bar with accurate start/stop positions if zoom level is high enough.

Requirement: Graphs should be zoomable
Priority: 1
Histograms bin sizes should be adjustable.
Example features:

- Option – automatically adjust histogram bin size to window viewed
- Option – fix histogram size to x.

Requirement: Graphs should be fast to navigate through
Priority: 2
During analysis a lot of time should be spent scrolling through graphs.
Example features:

- Option – jump to next aligned read
- Option – jump to genome position specified as integer

Requirement: Feature annotation track
Priority: 2
Features read in from a GFF file can be displayed as horizontal bars together with their orientation, positive or negative.
Example features:

- Gene features may overlap, i.e. several frames are needed in the track.
- Genes are correct to their start and stop positions.

Requirement: Map species RefSeq code to species name
Priority: 2
Use table metatie_fastalines.txt to map RefSeq code eg NC_002516 to name "Pseudomonas aeruginosa PAO1"
Example features:

- Display names and refseq numbers in left hand panel.

Requirement: Map genera name to lineage name
Priority: 2
Genus, eg. Pseudomonas, should be mapped to lineage, in this case Gammaproteobacteria.
Example features:

- Display lineage in left hand panel.

Requirement: SNP visualisation
Priority: 5
SNPs, i.e. mismatches between the letter in the reads and the letter in the genome string, should be flagged in the visualisation.
Example features:

- Positions might be easiest to find with an external program to save memory. eg. Samtools, Varscan
- should be optional
- Should only be run for one genome or transcriptome.

Requirement: Text export
Priority: 2
Export text analysis and summary statistics from the left hand panel.
Example features:

- Export as tab separated csv format.

Requirement: Sequence export
Priority: 4
Export the sequence of a particular region (drag selection of region).
Example features:

- Export sequence of DNA ACTG to textbox, where it can be used for further analysis

Requirement: Analysis of read numbers
Priority: 2
Count number of reads read in and print in left hand text box.
Example features:

- Display numbers of reads attributed to each species / gene.
- Metagenomics: Normalise number of reads which hit each genome by genome length (summary file still to be prepared by CD).
- Transcriptomics: Normalise number of reads which hit each gene by gene length to 1000bp and normalise total number of reads to one million (Reads per Kilobase per Million RPKM).

Requirement: Graph export
Priority: 4
Preferred export of graphs to vector formats.
Example features:

- SVG or PDF
- PNG

Requirement: Run external alignment
Priority: 3
Run alignment of specified reads file (in fastA, fastQ format) against a built reference database of bacterial species. Linux environment as most aligners are linux based.
Example features:

- Can set alignment settings, i.e. set number of cores used, input file, number of mismatches allowed
- Can change aligner to be used – first bowtie, then bwa, soap2
- Upon alignment completion read SAM BAM file in.
- Progress bar of alignment (may take hours), can be estimated by number of lines in the SAM file being output by the aligner.

Requirement: Analysis for transcriptomes
Priority: 4
Can find reads that group together into continuous segments outside of genes, and list them as new potential sRNAs or ORFs
Example features:

- Find previously undefined continously covered regions (not in annotation) covered by at least x reads of coverage x.
- Output list of coordinates and sequences.

Non-functional requirements

Requirement: Usability
Priority: 1
Functions in the program should be easily accessible to non-technically oriented biologist users.
Example features:

- Clear menu system
- Install guide

Requirement: Memory efficiency
Priority: 1
The program should be able to read between 0.1 million and 100 million reads from a SAM/BAM file into RAM rapidly. A maximum of 2-4 GB of RAM should be used.
Example features:

- Fast parser
- Memory usage indicator in % of x GB

Requirement: Multi processor capable
Priority: 4
It would be useful if the program could be sped up by using unused processor cores.
Example features:

- Parallel reading of large input files.
- Parallel processes for background tasks, such as generating statistics.
- Parallel loading of adjacent windows for smooth scrolling.

Requirement: Documentation
Priority: 1
The entire project should be documented.
Example features:

- Source code documentation
- Doxygen documentation
- All documentation in english, will be proofread by CD
- How to tutorial
- Tips and tricks

Requirement: Java
Priority: 1
The program code and any code for running external programs on the command line should be written in Java.
Example features:

- Pure Java preferred.
- Clear organisation of open source external packages used.

Requirement: Platform independent
Priority: 2
It must be possible to easily install and run the program on Linux and Windows, MacOS is also desirable.
Example features:

- Linux, Windows, MacOS install wizard.

- Well organised and packaged code, eg. as JAR files.

Requirement: Website
Priority: 2
A simple website in html/php describing the program and allowing download. Servers are available at Hannover Medical School for this purpose.
Example features:
- explanation
- screenshots
- download page, multiple versions

Requirement: Open source
Priority: 2
Software source code should be made available sometime in or after the main development process.
Example features:
- Sourceforge project, or simply source code available on website.