

Perjantai 11.1. : Assembly, Scaffolding.

- Connect to hippu.csc.fi using PuTTY, enable Xming.

Set a name to a path to the work folders prepared by the teacher:

```
set COURSEDIR=/fs/lustre/wrk/somervuo/ngskurssi
set DATDIR=$COURSEDIR/data
set REF=$DATDIR/NC_000913.fna
set SCRIPTS=$COURSEDIR/scripts
```

Data: Illumina PE, 600x E.coli (SRA: ERP000092).

```
zcat $DATDIR/ERR022075pe.fasta.gz | head -1500000 > subset.fasta
```

- zcat uncompresses on the command line and concatenates files and print on the standard output
- head -1500000 shows first 1500000 bytes of the file.

Data quality checking:

```
module load fastx
zcat $DATDIR/ERR022075_1.fastq.gz | head -1500000 > reads1.fq
zcat $DATDIR/ERR022075_2.fastq.gz | head -1500000 > reads2.fq
fastx_quality_stats -i reads1.fq -o reads1.qv
```

- *fastx* -i input -o output. -> txt-file with quality scores on different lines.

Task: How average quality goes towards the end of reads?

- The average quality drops towards the 3'-end. Incorrect bases negatively impact assembly, mapping, and other downstream analysis.

Genome de novo assembly with Velvet:

Velvet is an assembler designed for short read sequencing. Velvet currently takes in short read sequences, removes errors then produces high quality unique contigs. It then uses paired-end read and long read information, when available, to retrieve the repeated areas between contigs.

k-mers: The sequence split into units of length k. These k-mers are entered into a hash table and contigs are constructed using a deBruijn graph.

When choosing k-mers for Velvet:

- Number must be odd.
- It must be below or equal to MAXKMERLENGTH variable in the global.h file.
- It must be strictly inferior to read length.

Example 1: Velvet with single read information.

```
module load velvet  
velveth vdirS 23 -fasta -short subset.fasta  
velvetg vdirS -exp_cov 15 -min_contig_lgth 1000
```

Velveth reads in these sequence files and produces a hashtable, and populates the hashtable with k-mers of length 23, and makes two output files (Roadmaps and Sequences).

Velvetg is the core of Velvet where the de Bruijn graph is built then manipulated. It's ran with the expected coverage of 15 and min. contig length is allowed to be 1000nt. When velvetg finishes it outputs the number of nodes, n50 (average contig length), and max and total size of the assembly created. In the auto_* directory, you will also see a few files: contigs.fa, LastGraph, PreGraph, Sequences, Graph2, Log Roadmaps, stats.txt .

Output: Final graph has 12344 nodes and n50 of 3405, max 13878, total 4677483, using 684519/750000 reads

```
$SCRIPTS/fastalen.pl vdirS/contigs.fa > contigsS.len  
$SCRIPTS/n50.pl contigsS.len
```

Perl script fastalen.pl calculates contig lengths and n50.pl calculates n50 value of summary of contig lengths.

```
contigS.len :  
>NODE_1_length_1888_cov_14.120233      1910  
>NODE_2_length_2072_cov_11.869691      2094  
>NODE_3_length_2157_cov_12.690774      2179  
>NODE_4_length_3644_cov_12.656696      3666  
>NODE_14_length_5235_cov_11.746514      5257  
>NODE_15_length_4972_cov_13.487329      4994  
>NODE_16_length_1751_cov_12.133067      1773  
...
```

Example 2: Velvet assembly with paired end information.

```
velveth vdirP 23 -fasta -shortPaired subset.fasta  
velvetg vdirP -exp_cov 15 -ins_length 500 -min_contig_lgth 1000
```

Velvet only handles interleaved or "shuffled" fasta and fastq files, where each pair is seen one after the other. Our data was already in this format, but velvet has a script to do it too.

```
shuffleSequences_fastq.pl input1 input2 > output
```

Output: Final graph has 10227 nodes and n50 of 57746, max 199412, total 4747681, using 723696/750000 reads

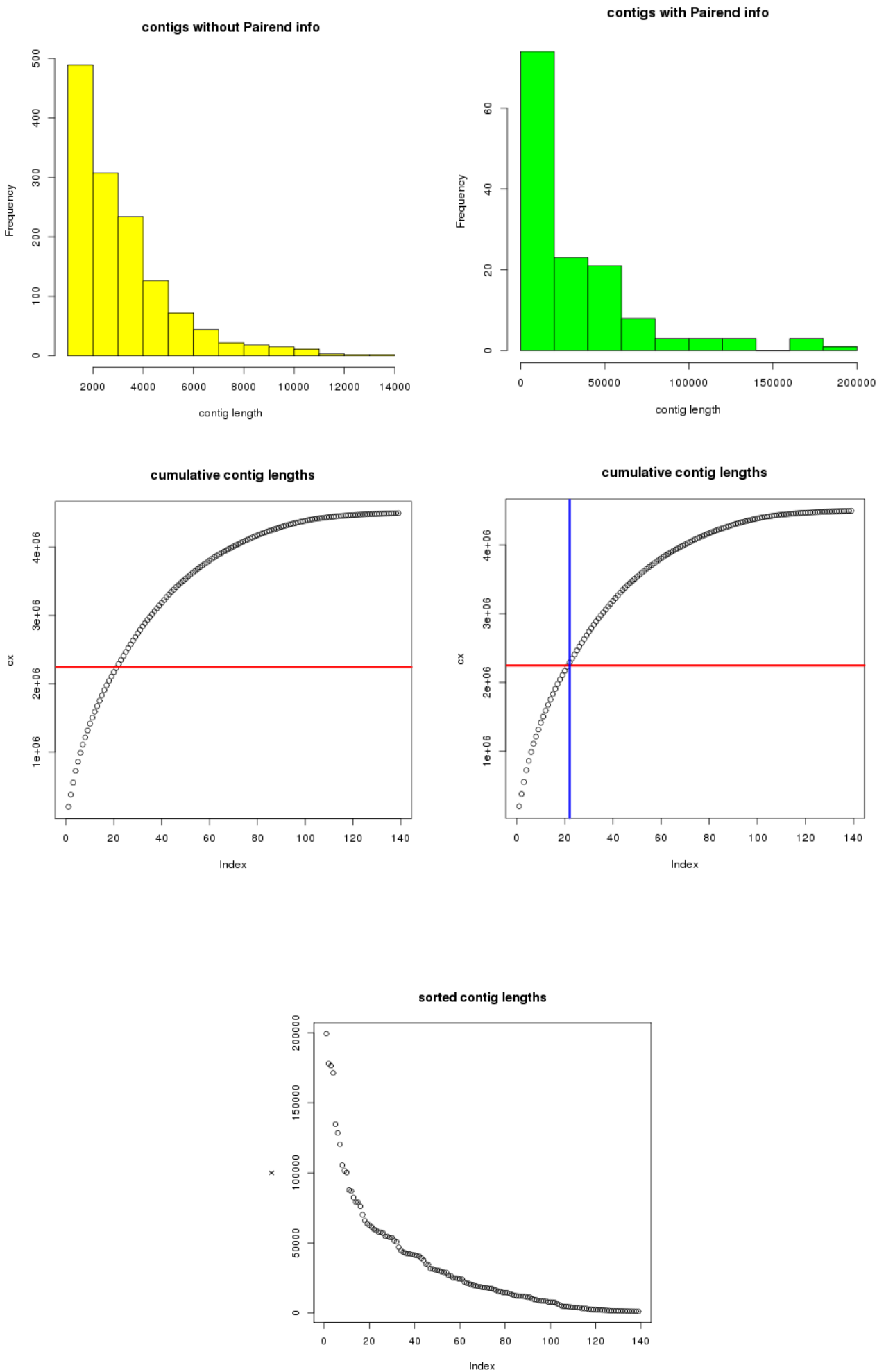
```
$SCRIPTS/fastalen.pl vdirP/contigs.fa > contigsP.len  
$SCRIPTS/n50.pl contigsP.len
```

Visualisation:

We can plot contig lengths e.g. using R which is a software for general purpose statistics and data analysis. "contigsS.len" and "contigsP.len" must be available in working directory of R.

```
module load R  
R  
a=read.table("contigsS.len", header=F)  
b=read.table("contigsP.len", header=F)  
hist(a[,2],col="yellow",main="contigs without Paired info",xlab="contig  
length")  
x11()  
hist(b[,2],col="green",main="contigs with Paired info",xlab="contig  
length")  
d=b[,2]  
x=sort(d,dec=T)  
cx=cumsum(x)  
plot(x,main="sorted contig lengths")  
dev.set(2)  
plot(cx,main="cumulative contig lengths")  
v=sum(d)/2  
abline(h=v,col="red",lwd=3)  
i50=min(which(cx>v))  
# this is N50:  
x[i50]  
dev.set(3)  
abline(v=i50,col="blue",lwd=3)
```

Graphs:



Compare assemblies with and without paired read information (number & length of contigs, N50).

- Paired end run produces less contigs that are longer. There were 139 contigs in paired and 1345 contigs in single. n50 values of paired run were much higher (57746) than the single run (3405). This means paired end information produces better quality contigs.

How the choice of kmer length affects the assembly?

- It's a trade-off between specificity and sensitivity. Longer kmers bring you more specificity and velvet will be much slower to run.

How parameters exp_cov and cov_cutoff affect the assembly?

- velvet doesn't know if a read it has is unique (f.ex. contains SNP and is hard to align) or from a repetitive region. With expected coverage you can guide velvet to make good guesses as to what the reads are. With low exp_cov, velvet will dismiss more reads as repetitive and tries to assemble all of them separately. If exp_cov is high it will assume they all come from the same genomic region and will take weighted average of the reads to build up the region.
- cov_cutoff removes low coverage nodes during assembly.
- both can be run as "auto", so velvet will determine the required values.

Can you think reasons why there are gaps in assembly, i.e. why we don't get a single long contig representing the entire genome even when using high sequencing depth? How we could improve assembly?

- Low coverage, bad read quality (sequencing errors, other technical errors), repetitive genomic sequences, reads deteriorate from both ends, methods are heuristic and very memory intensive: final answer is not always absolutely true.
- Longer reads, high coverage, minimize errors.

Example 3: assembly with 60X coverage data:

```
zcat $DATDIR/ERR022075pe.fasta.gz | head -5400000 > datape.fasta
velveth vdirP23 23 -fasta -shortPaired datape.fasta
velvetg vdirP23 -ins_length 500 -exp_cov 60 -min_contig_lgth 1000 -
scaffolding no
$SCRIPTS/fastalen.pl vdirP23/contigs.fa > len1
$SCRIPTS/n50.pl len1

grep '^>' vdirP23/contigs.fa | sort -n -k6 -t"_"
grep '^>' vdirP23/contigs.fa | sort -n -k4 -t"_"
```

Assembly validation:

To determine how good our contigs are we can map them against the genome. We use bwa:

```
module load bwa  
  
bwa index -a is $REF  
  
bwa bwasmw -t 4 $REF vdirP23/contigs.fa > contigs.sam
```

BWA is a software package for mapping low-divergent sequences against a large reference genome. We use the bwa-sw algorithm, that's designed for long sequences (70bp-1Mbp). bwa index indexes database sequences in the fasta format. Then the alignment algorithm bwasmw is run, with -t for multithreading with 4 threads on the hippu-server. It creates a SAM-file, that contains all the information on where the reads aligned and how well they aligned.

One line:

```
@SQ      SN:gi|49175990|ref|NC_000913.2|      LN:4639675  
NODE_38_length_1246_cov_14.805779      16  
      gi|49175990|ref|NC_000913.2|      734360      250      1268M      *  
      0      0  
      aTGGAATTTTTATCAGTATCCGCTGAATCCGGTTCAGTATATAGATTCAATGGGACTGGCATC  
AAAATATGGACACTTAAATAATGGCGGATATGGAGCGAGACCCAACAAACCGCCTACGCCCCGATCCAAGTAAA  
TTGCCGGACATAGCGAAACAATTAAGACTGCCATATCCTATTGACCAGGCCAGTAGTGCGCCTAATGTTTTCA  
AAACATTCTTCAGAGCATTAAGCCCTTACGACTACACACTGTATTGCAGGAAGTGGGTAAAACCAAATCTGAC  
TTGTACGCCACAGGATGATTCCCAGTATCCAGGGATGGATACAAAGACAGCAAGTGATTACCTGCCACAGACA  
AATTGGCCAACAACCTCAATTACCACCAGGATATACTTGTGCAGAACCCTATTTATTTCCAGACATTAATAAAC  
CCGATGGGCCAGCAACAGCAGGGATAGATGATTTGGGTGAAATTTTAGCTAAGATGAAACAGAGAACATCGAG  
AGGAATAAGAAAATGAAAAGAGTTTTGTTCTTTTTGCTGATGATATTTGTTAGTTTTGGTGTATAGCTGATT  
GCGAAATACAAGCTAAAGATCATGATTGTTTTACTATTTTCGCTAAGGGTACTATTTTTTCCGCGTTTCTCTGT  
CTTAAATAATAAAGCCATGTGGAGATGGTATCAGAATGAGGACATCGGTGAGTATTATTGGCAAACAGAACTG  
GGTACATGTAAAAATAATAAATTCACACCAAGCGGGGCGAGACTCTTAATTCGCGTCGGGTCACTACGCCTTA  
ATGAAAATCATGCCATCAAAGGAACGTTGCAGGAGTTGATAAATACAGCAGAAAAAACGGCGTTTCTCGGCGA  
TCGGTTTtagAAGTTATATAAGAGCGGGTATCTATCAAAAAAAGAGCAGTGATCCAGTTCAGTTGTTGGCAGTC  
CTTGATAATTCAATCATGGTAAAATATTTTAAAGATGAGAAACCAACTTATGCTCGAATGACAGCACATCTCC  
CGAACAAAAATGAGTCTTACGAATGTTTAAATAAAGATACAGCATGAACCTAATTCGTAGTGAAGAGAAATAAAA  
TCTCTTCTGAGAAAACAAATTAATCATTGTGAAAAATTATATAACTCATGGAATAGGTGTTTATTTTGC GTT  
CCATTTGCAGGGAAAGATCACGTAACGCTACTTTTTTGTACTAAATAATTGCAATTTTAAAGTTCAACAATTGA  
GATACTACTTATTGTCTAAAGCTGTTTTTCATAGCTT      *      AS:i:1268 XS:i:0  
      XF:i:0      XE:i:26      NM:i:0
```

Fields are tab separated and first line with @SQ is the header line for the whole file, containing information relative to all reads. Then one line of the contigs we had, contig name (NODE_38 (...)). SAM-flag, reference name, where it mapped on the reference, mapping quality... etc.

Where contigs mapped was the visualized with tablet and hawkeye.

Assembly with SOAPdenovo

SOAPdenovo is a novel short-read assembly method that can build a de novo draft assembly for the human-sized genomes. The program is specially designed to assemble Illumina GA short reads.

Task 1. Create config file soap.config with text editor. I used “nano”. This information is what you have about your sequencing run, like read length, insert sizes, etc. You can set cutoffs to ignore the ends of reads.

```
#maximal read length
max_rd_len=100
[LIB]
#average insert size
avg_ins=500
#if sequence needs to be reversed
reverse_seq=0
#in which part(s) the reads are used
asm_flags=3
#use only first 75 bps of each read
rd_len_cutoff=75
#in which order the reads are used while scaffolding
rank=1
# cutoff of pair number for a reliable connection (default 3)
pair_num_cutoff=3
#minimum aligned length to contigs for a reliable read location (default 32)
map_len=32
#fastq file for read 1
#q1=/path/**LIBNAMEA**/fastq_read_1.fq
#fastq file for read 2 always follows fastq file for read 1
#q2=/path/**LIBNAMEA**/fastq_read_2.fq
#fasta file for read 1
#f1=/path/**LIBNAMEA**/fasta_read_1.fa
#fastq file for read 2 always follows fastq file for read 1
#f2=/path/**LIBNAMEA**/fasta_read_2.fa
#fastq file for single reads
#q=/path/**LIBNAMEA**/fastq_read_single.fq
#fasta file for single reads
#f=/path/**LIBNAMEA**/fasta_read_single.fa
#a single fasta file for paired reads
#p=/path/**LIBNAMEA**/pairs_in_one_file.fa
p=datape.fasta
```

```
module load soapdenovo
SOAPdenovo-3lmer all -s soap.config -K 23 -o soap23b -p 4
```

SOAPdenovo uses the config file to find the information it needs. You set it with `-s`. K-mer is set with `-k` and `-p` is number of cpu (threads) for use.

Perjantai 18.1.

Lecture notes:

Gene prediction, a pattern recognition problem

- Ab Initio (Itseoppivat: etsii geenisignaaleja)
 - Prokaryootit: esim. S-D, -10/-35. Alku(Met) ja lopetuskodoni samassa lukukehyksessä. Ei eksoni-/intronirakennetta, esim. S-D on aina läsnä. -> helpompaa.
 - Eukaryootit: 3% koodaustiheys, eksoni-/introni-rakenne, isot genomit, kaikki signaalit (kuten Kozak) eivät joka geenissä -> vaikeuttaa.
Geenisignaalit: esim. RNAPolII prom. elementit, Kozak (RBS), PolyA, ..
 - Sisältösignaalit:
 - GC% korkeampi koodaavassa alueessa (45-50%), mutta ei riitä yksinään koska myös CpG-saarekkeet ovat GC tiheitä. Pelkän GC-perusteella ~63% oikein.
 - "Codon Usage Bias". Jotkut kodonit käytetympiä koodaavalla alueella ja joitakin ei esiinny juuri ollenkaan. Voidaan laskea todennäköisyys sille, että kyseinen sekvenssi ei ole koodaava alue ja sen log-likelihood. (- => ei ole koodaava, + => koodaava) Pelkkä kodon-bias ennustustarkkuus ~87%.
 - Splice-sitet. 5' GU (Donor), 3' AG (Acceptor). [Exot₁]GU ---- AG[Exon₂]. Vain ~0.5%:n ei päde.
 - Bakteeridatan elementit tunnetaan hyvin, mutta eukaryoottidata on niin vaikeaa, että kehitys hitaampaa.
 - HMM: Tarvitaan kaksi opetusjoukkoa CpG-aluetta ja ei-CpG-aluetta. Muodostetaan +/- matriisi, jossa siirtymätodennäköisyydet. Kysytään mitä mallia sekvenssi noudattaa? Jos siirtymät vastaavat paremmin +/- matriisin arvoja -> koodaava.
 - GHMM: Monia tiloja joiden on toteuduttava ja ovat luonteeltaan sellaisia, että kun kaavion läpi käydään "oikein" järjestyksessä niiden toteutuminen varmentaa onko kyseessä e. eksoni/introni.
- Olemassaoleva data, e. RNASeq. Ei voi löytää geenejä, joita ei ekspressoida.
- Paras käyttää molempia menetelmiä.

Set correct paths for programs prodigal, RNAmmer, tRNAscan, Genemark. Then get the genome with wget. Then check if there are Windows linebreaks in file with the perl script, because those will cause an error in the next step.

```
setenv PATH $PATH:/p/appl/molbio/course/prodigal"
setenv PATH $PATH:/p/appl/molbio/course/rnammer"
source /p/appl/molbio/course/tRNAscan-SE/tRNAscan-SE-
1.23/setup.tRNAscan-SE
/p/appl/molbio/course/GeneMark/gmes
wget
http://ekhidna.biocenter.helsinki.fi/downloads/teaching/spring2013
/SCC3193.fasta
perl -pe '$_ =~ s/\r//' SCC3193.fasta > SCC3193_no_breaks.fasta
```

Predict genes: Run Prodigal:

```
prodigal -a SCC3193.pep -i SCC3193_no_breaks.fasta -f gff -o SCC3193.gff3
```

Predict tRNAs:

tRNAscan-SE was designed to make rapid, sensitive searches of genomic sequence feasible using the selectivity of the Cove analysis package.

```
tRNAscan-SE -B -o SCC3193_tRNA.out SCC3193_no_breaks.fasta
```

-o *file*

: save final results in *file*

Specify this option to write results to *file* rather than standard output.

-B *number*

: use *number* nucleotides padding for first-pass tRNA predictions

By default, tRNAscan-SE adds 7 nucleotides to both ends of tRNA predictions when first-pass tRNA predictions are passed to covariance model (CM) analysis. CM analysis generally trims these bounds back down, but on occasion, allows prediction of an otherwise truncated first-pass tRNA prediction.

We would like to predict all different rRNA molecules (5/8s, 16/18s and 23/28s) from *P. wasabiae* SCC3193 genome and we would like to have results in gff format. Scan both strands at the same time. What parameters we should use in this case?

```
rnammer -S bac -gff SCC3193_rrna.gff2 -multi -m tsu,ssu,lsu  
SCC3193_no_breaks.fasta
```

-multi

Runs all molecules and both strands in parallel

-m Molecule type can be 'tsu' for 5/8s rRNA, 'ssu' for 16/18s rRNA, 'lsu' for 23/28s rRNA or any combination separated by comma.

-gff output gff file

Specifies filename for output in GFF version 2 output

-S Specifies the super kingdom of the input sequence. Can be either 'arc', 'bac', or 'euk'.

ARGO:

Bakteeri Genetic Code: 11

Count SSC_tRNA.gff2 77

Soft-Masked sequence:

01/17/2013 12:33 8,904,437
[Drosophila melanogaster.BDGP5.70.dna_sm.chromosome.3R.fa.gz](#)

Hard-Masked sequence:

01/17/2013 12:33 8,284,475
[Drosophila melanogaster.BDGP5.70.dna_rm.chromosome.3R.fa.gz](#)

➔ gunzip

Chromosome Statistics

Length (bps)	27,905,053
Protein coding gene count	3,370
Non coding gene count	239
Pseudogene count	26
Variations	1,397,489

GeneMark ES (itseoppiva menetelmä):

The HMM architecture of eukaryotic GeneMark.hmm consists of hidden states for initial, internal and terminal exons, introns, intergenic regions and single exon genes located on both DNA strands. It also includes hidden states for initiation site, termination site, as well as donor and acceptor splice sites. GeneMark.hmm has been frequently used for annotation of plant and animal genomes.

Friday 25.1. : Genome structural annotation (Apollo)

Lecture notes:

Annotation

Repeats: Low-complexity Repeats (poly-purine, high AT/GC, tandem repeats), Retro-/DNA-transposons. Pseudogenes/multicopy genes...

Known repeats can be masked (soft/hard) to avoid low-complexity areas to peak/mask under important reads.

- Repbase database for known repeats

De novo repeat prediction: calculate consensus representing repetitive DNA from study species.

Pairwise alignment of the genome -> cluster alignments and find/define elements to align again, group substrings into repeat families.

RepeatScout calculates consensus local alignment Q (Mapping Quality?) for large fragments

-> fit alignment for smaller fragments -> fit-preferred alignment score for medium sized fragments

Finds a window in which to search for similar repeats with maximum score.

RepeatMasker

Remove repeats from the genome by masked by the RepeatScout information and already known repeats from similar species etc.

Esim. Repeat masking of *M.cinxia* genome

Pelkällä RepBasella vain ~7.5%, lisäksi RepeatScout/Recon/Piler... -> ~40% contigeista, ~32% scaffoldista.

Gene prediction with MAKER

(ab initio last fri) -> matemaattinen malli, hmm ja opetusjoukko esim. sellaisia eksoneita, jotka tunnetaan varmasti -> niiden perusteella lasketaan todennäköisyysmatriisi (e. miten usein yhden eksonin geeni /kahden/etc muita tapahtumia.) Opetusjoukko tärkeä.

With MAKER -> käytetään tunnettua tietoa databaseista (proteiinit etc) etc. Maskataan pois tunnetut ja ennustetut repeatit, RNA-Seq assembly/mapping.

MAKER maskaa high-complexity hardilla ja low-complexity softilla.

MAKER ab initio gene prediction with SNAP.

SNAP uudelleeniteroi opetusjoukkoaan

Protein evidence: Uniprot tarkistettut (refseq) sekvenssit läheisistä suvuista, tai konservoituneet proteiinit

BlastX löytää konservoituneiden proteiinien sekvenssit / Exonerate pakottaa splice/sitet oikein (gt....ag erittäin konservoitunut signaali) .

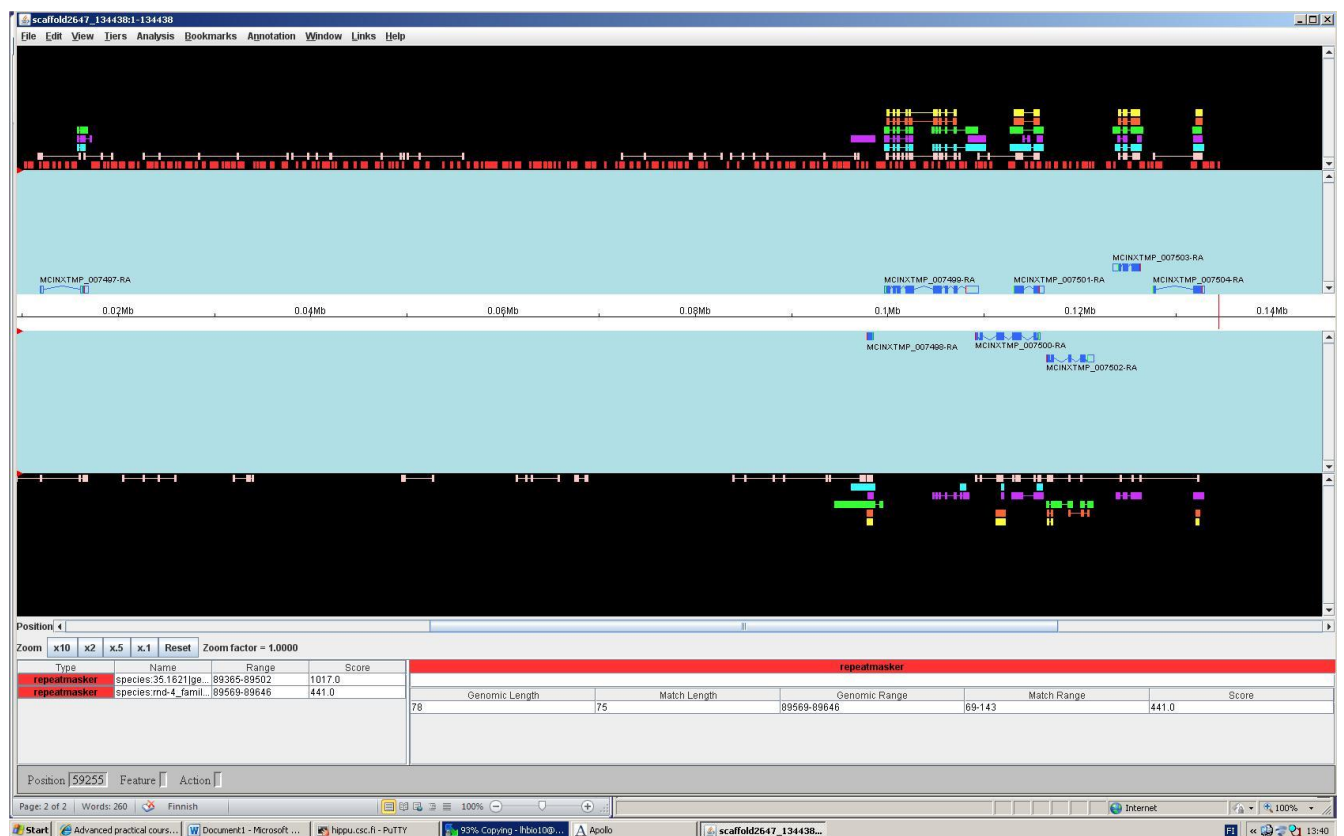
Exercises:

Open scaffold2647_134438.gff in wordpad. What does the file contain ?

- File contains everything what maker has done: standard orientation, frames, exon, all info from SNAP, RepeatMasker, etc. Everything is in gff3 format.

Open scaffold2647_134438.gff in gff3 format in Apollo. Change the colors for tiers and save them to type preferences.

- All different types of results are in different color. Boxes are exons, lines introns.



Check and correct, if necessary, the gene model MCINXTMP_007504-RA in Apollo. Save the corrected model in gff3 format.

- After altering the exons a bit according to blastx and blastn results and ignoring cufflinks only results I BLASTed the sequence against nr:
- 1 ribosomal protein S5 [Manduca sexta] 446 446 98% 1e-157 99% ACY95347.1
- 2 ribosomal protein S5 [Trichoplusia ni] 442 442 98% 8e-156 98% AAX63830.1
- 3 ribosomal protein S5 [Euphydryas aurinia] 441 441 98% 1e-155 97% ADT80675.

-

Score	Expect	Method	Identities	Positives	Gaps	Frame
446 bits(1147)	1e- 157()	Compositional matrix adjust.	216/219(99%)	216/219(98%)	1/219(0%)	

The protein is correct and I have edited it very well (nearly full identity). I also made a MAFFT-alignment with this and other similar proteins to confirm.

Open scaffold825_259930.gff and check the model MCINXTMP_015555-RA. Check the predicted protein similarly than in 6).

I removed two exons from the start of the predicted gene, because they overlapped with another gene and they had no other supporting data other than SNAP. Results from blastx etc. started after two exons. These two exons were very far away from the rest of the predicted gene.

I made a new starting point by expanding according to est2genome and est_gff_cufflings and made the first exon a bit longer than predicted, because the suggested starting codon would have not made a complete protein.

All exons except the last one (with stop) looked very good and supported by most data. The last exon was unclear/strange and it wasn't supported by blast results, but was supported by SNAP. I left it there, because it seemed okay otherwise and stop codon was good.

Blast results against uniprot:

- 1 RecName: Full=60S ribosomal protein L13a 400 400 100% 1e-141 94% Q8MUR4.1
- 2 RecName: Full=60S ribosomal protein L13a 391 391 100% 4e-138 92% Q962U0.1
- 3 RecName: Full=60S ribosomal protein L13a 289 289 100% 6e-98 65% Q9VNE9.1

All of the best results are butterflies, similarity goes down when results are other insects.

Score	Expect	Method	Identities	Positives	Gaps	Frame
400 bits(1028)	1e- 141()	Compositional matrix adjust.	191/204(94%)	200/204(98%)	0/204(0%)	

Very similar to the best result, good edit.

Check and correct the gene model MCINXTMP_015558-RA from scaffold825_259930. Use blast and Ensembl sequence search for checking the predicted protein.

I rearranged the middle exons length because it wasn't according to any of the "known" results, like blasts. I also realigned the last exon to be more according to blast and SNAP results.

BLAST results produce a Serendipity A-protein family protein.

Results from Ensembl Metazoa:

Heliconius melpomene [HE670847](#) 169 495 65 % 4E-53 [View](#) [HMEL008918](#)

```

Query:475 GluCysLysIleValSerAsnLeuLeuSerSerAlaAsnGlyHisIleTyrGluLeuLys Query:494
|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45417 GAATGTAAAATCGTCTGTAACATGTTAAGTTCTAAAGGTGATCATTTGTACGACGTGAAA Sbjct:45476
Query:495 LysLysThrGlnAsnThrThrValThrValAspGlnLeuIleLysArgLeuLysLeuLeu Query:514
||||! !! ! !!!!!!!:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45477 AAGACAAAAGCACAAAACAACGTTACTACAGAACAATTAATAAAACGCTCTCAAATTATTA Sbjct:45536
Query:515 TyrThrLeuValLysArgIleAsnAspLeuPheAsnProLysGluAsnAspGluGluPhe Query:534
|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45537 TACACATTGGTCAAGAAGATCAATACTATAATGAATCATAAA---AATGACGACGAGTTT Sbjct:45593
Query:535 TyrGluGluProValGluAsnLysAsnLysThrHisThrIleAsnPheValAsnGlyAsn Query:554
|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45594 TATGAAGAACCAATAGAA-----AATAACACTCACACAGTAGATTTAAAAAATGGCGAC Sbjct:45647
Query:555 ThrTyrIleAsnSerProLysArgGlnThrTyrAsnValThrArgSerIlePheProArg Query:574
|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45648 ACTTATATCAATTCACCGAAAAAA---ATTAATAACGTTACCAGAAGTATTTTGTCTAGA Sbjct:45704
Query:575 ThrAsnIleArgSerSerThrSerArgLeuProLeuAlaMetLeuThrLysArgLeuArg Query:594
|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45705 ACTAATATAAGGTCGTCGACAGGAAGATTTCTCTTGGCTATATTAACATAACATTTAAAA Sbjct:45764
Query:595 SerLysLysIleAsnAspLeuSerPheSerIleGluPheAspGluPheCysAsnLeuSer Query:614
|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45765 TCGAAACAAAATAATGATTTAAGTTTCTCATTTGCAATTGGATGAAGTTGTAACATATCG Sbjct:45824
Query:615 GluIleLysSerLysLysArgGluThrSerIleLeuTyrSerProPheLysProArgSer Query:634
|||||! |||| ! ||||| !||||| ||||| ! !! ! !!!!!!!
Sbjct:45825 GAGATTAAAATAAAACATCGAGAACAAATCGATTTTGTATCATACGCCATTAAAAAATAGA Sbjct:45884
Query:635 SerLeuArgLysAlaValLeuSerArg Query:654
|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||:|||||
Sbjct:45885 TGTCTGAGAAAAGCTGTTCTAAGCAGA Sbjct:45944

```

The prediction was okay, I've probably made some slight errors in the middle codon when I altered it.

Check and correct the gene model MCINXTMP_007503-RA from scaffold2647_134438

I changed the length of the last two exons so that the splice site is correct between them and they are more in line with blast/protein2genome results.

I then BLASTed the sequence and the result was a NADPH:adrenodoxin oxidoreductase, mitochondrial.

Score	Expect	Method	Identities	Positives	Gaps	Frame
444 bits(1143)	5e- 151()	Compositional matrix adjust.	232/446(52%)	297/446(66%)	21/446(4%)	

Identity was not perfect, The alignment shows most of the inconsistencies in the end of the protein. Everyone in class got the same result.

Friday 1.2: Genome structural annotation (Apollo)

Set paths and load all needed programs and files:

```
setenv PATH $PATH:"/v/users/ahola/bin/maker/bin"
setenv PATH $PATH:"/v/users/ahola/bin/RepeatMasker"
module load exonerate
module load bioperl
module load blast+
module load snap
module load augustus
echo $PATH
cp /v/users/ahola/bin/kurssi/* .
```

Run RepeatMasker:

```
RepeatMasker -dir repeats_RepBase -species all scaffold3244.fasta
RepeatMasker -dir repeats_denovo -lib denovo_repeats.fasta scaffold3244.fasta
```

Results:

.masked file that has hard masked repeats (areas with repeats are replaced with N's).

.tbl file that says where and how many repeats have been masked.

- Multiple runs find slightly different amounts of repeats, so as results there's slightly differently masked end files.
- (all) Total interspersed repeats: 9751 bp 1.95 %
 - Unclassified vain ~0.6%.
- (denovo)Total interspersed repeats: 101653 bp 20.31 %
 - Unclassified 18.09% -- Tuntemattomia repeatteja.

Denovo repeat masking is much harder and produces more uncertain results.

Run Maker:

Generate empty control files for Maker: maker -ctl

Edit this control file with an text editor: nano maker_bobts.ctl

Change BLAST-type from wublast to ncbi.

Then give MAaker RNASeq Data and ask it to make gene predictions. Change maker_opts.ctl to say:

```
genome=scaffold3244.fasta
est=RNASeq_trinity.fasta
est_gff=RNASeq_cufflinks.gff
est2genome=1
```

Then run maker. When maker is finished the log file says:

```
less scaffold3244.maker.output/scaffold3244_master_datastore_index.log
ls -l scaffold3244.maker.output/scaffold3244_datastore
```

You get the path of the produced files from log or with `ls -l`. Path in my case was:

```
scaffold3244_datastore/B2/EE/scaffold3244_500584/
```

I took the .gff file from that directory and trained SNAP with it. Training SNAP means to use known values to readjust the probability values for changing nodes in SNAP.

```
maker2zff scaffold3244_500584_RNASeq.gff
fathom -categorize 1000 genome.ann genome.dna
fathom -export 1000 -plus uni.ann uni.dna
forge export.ann export.dna
hmm-assembler.pl snap1 . > snap1.hmm
```

Then run maker so that it makes SNAP ab initio predictions using previous data. Edit the maker control files to use the previous .hmm file and the run maker with those settings.

```
est= "leave empty"
est_gff= "leave empty"
snaphmm=snap1.hmm
est2genome=0
```

Then we configured new control files so that Maker would use SNAP, RNA-Seq, and protein evidence:

```
est=RNASeq_trinity.fasta
est_gff=RNASeq_cufflinks.gff
protein=scaffold3244_prots.fasta
model_org=all
rmlib=denovo_repeats.fasta
snaphmm=snap1.hmm
```


Open the gff files including the maker gene predictions in Apollo and compare predictions made only using RNASeq data, and all evidence data + SNAP ab initio predictions. What can you conclude ?

In the files produced by the two runs:

snap1_only: no exon predictions, only snap_masked layer.

snap1.gff: many datapoints, very good looking genes.

We also used Augustus for ab initio gene predictions, it had good looking exons.

Conclusions: Ab initio gene prediction is hard, especially with a previously fairly unknown species because there's so little data and then you have to make many predictions based on predictions, and your results might end up not being very reliable and requiring a lot of manual annotation. When you have protein data available the task becomes easier and more reliable.

Friday 8.2. Artemis and RNA-Seq data in manual annotation

Lecture notes:

fathom käsittelee sekvenssin, jossa korkeintaan 1kb. Lajittelee.

forge estimoi

hmm-assembler -> tekee hmm-mallin.

Sekvensointivirheet NGS:sä yleisiä. 454, 1% keskittyneitä, homopolymeerejä. Assemblerit yrittävät korjata sekvensointivirheitä.

BAM files are binary SAM-files. They are very compressed, but can be viewed in specialized viewers like Artemis. Artemis can display pairs against the reference (both pairs joined by lines), coverage of the reads, the actual sequence of the reads, etc. You can zoom in to see the read sequences against the reference and see the constructed consensus. Artemis can also show exons and read frames. To be viewed BAM-files must be indexed. There are many other tools to parse BAMs with. One very good toolset is "bamtools".

Checklist for making gene models in Apollo:

- Check opposite strand
- Start/Stop sites
- Missing/additional exons
- Edges of Exons
- Splice sites
- RNASeq for 5', 3'.
- Pull out SEQ > BLAST > MSA.

Cufflinks

- Very long introns, bad alternative transcripts, merged genes.
- Check Cufflinks data for errors.

Manual annotation:

maker-scaffold3244_500584-augustus-gene-4.4-mRNA-1

I removed the exons that were only in the cufflinks data. I added one exon, that was in all blast*-data and protein2genome-data. I also added one flanking extra exon, that was in RNASeq/Protein2genome data.

BLAST results: putative furrowed [Danaus plexippus]

Alignment statistics for match #1						
Score	Expect	Method	Identities	Positives	Gaps	Frame
1759 bits(4556)	0.0()	Compositional matrix adjust.	837/932(90%)	881/932(94%)	1/932(0%)	

augustus_masked-scaffold3244_500584-abinit-gene-3.3-mRNA-1

I removed the beginning of the gene to be according to protein2genome and blast-results. I removed parts from start and end that were only shown in cufflinks and weren't in RNASeq.

BLAST results: hypothetical protein KGM_18655 [Danaus plexippus]

Alignment statistics for match #1						
Score	Expect	Method	Identities	Positives	Gaps	Frame
538 bits(1385)	0.0()	Compositional matrix adjust.	264/266(99%)	265/266(99%)	0/266(0%)	

maker-scaffold3244_500584-augustus-gene-2.4-mRNA-1

I removed a very long predicted area that was only based on cufflinks data and was not repeated in RNASeq/prot2genome/Blast... I tried to find a good first exon that had a good start codon and was in all the RNASeq/prot2genomes.

BLAST results: putative receptor tyrosine phosphatase type r2a [Danaus plexippus]

Alignment statistics for match #1						
Score	Expect	Method	Identities	Positives	Gaps	Frame
3367 bits(8731)	0.0()	Compositional matrix adjust.	1657/1933(86%)	1710/1933(88%)	177/1933(9%)	

I think I removed a bit too much from the start. In the alignment it shows that I start at base number 144 in the reference, so I've gone over that sequence when I was looking for a good start place.

MCINXTMP_008074-RA from scaffold scaffold2760_48179.gff.

I removed a couple of exons from both the start and the end of the predicted gene, because they had no hits in prot2genome or RNASeq or BLAST. I added a new final exon from RNASeq data.

My first result was not very good:

BLAST result: UDP-glycosyltransferase UGT33F3 [Helicoverpa armigera]

Alignment statistics for match #1

Score	Expect	Method	Identities	Positives	Gaps	Frame
557 bits(1435)	0.0()	Compositional matrix adjust.	263/475(55%)	341/475(71%)	28/475(5%)	

I then elongated one of the middle exons and the last exon:

UDP-glycosyltransferase UGT33F3 [Helicoverpa armigera]

Alignment statistics for match #1

Score	Expect	Method	Identities	Positives	Gaps	Frame
615 bits(1585)	0.0()	Compositional matrix adjust.	284/475(60%)	363/475(76%)	6/475(1%)	

The gaps reduced a lot. Everyone else also had similar results.

Friday 15.2. Functional Annotation

Lecture notes:

DE annotation – Descriptive functional annotation.

GO annotation – Kolme pääluokkaa, hierarkinen rakenne, sisältää tiedon siitä miten tieto on hankittu IEA (Inferred from Electronic Annotation) epäluotettavin, ihmisten tarkistamat luotettavimpia. Malliorganismit usein hyvin annotoituja, mutta esim. bakteerien annotaatiot ovat elektronisesti annotoituja ja epäluotettavempia verrattuna hiivoihin kuten *Candida albicans*.

GO Informaatioisisältö: Lasketaan kuinka monta geenii määrittävät GO-luokan. Sisänode sisältävät aina kaikki siitä polveutuvien lehtien geenit. Jaetaan jokainen lehti ja sisänode juuriarvolla. Otetaan –lg niin saadaan suuri arvo kuvaamaan harvinaista GO-luokkaa ja sitä kuvaavampi se on luonteeltaan. Mitä lähempänä juurta sitä abstraktimpi GO on.

Voidaan laskea semanttisia etäisyyksiä kertomaan kuinka samanlaisia GO-luokat ovat keskenään.

Resnik = yhteinen ”esi-isä”.

Lin = Yhteinen vanhempi kerrotaan kahdella ja jaetaan kahden ontologian informaattiosisällön summalla.

Korrelaiko semanttinen etäisyys sekvenssisamanlaisuuden kanssa? Riippuen laskentatavasta ja GO-luokasta. Esim. Resnik toimii parhaiten Molecular Function-yläluokassa, Lin toimii parhaiten Cellular Component –luokassa, jne.

EC annotation – Enzyme Commission number . Proteiinien entsyymiaktiivisuus.

Domain annotation – Lokaali yhdenkaltaisuus.

Virheiden rikastuminen tietokannoissa:

Tarkista aina: Linjauksen kattavuus (huom. BLAST-kattavuus vain yhdensuuntainen).

Weakening Signal

- Manuaalisesti kuratoitu proteiini (oikea) lisätty tietokantaan, sitä kohti liitetään elektronisesti annotoituja sekvenssejä, joita kohti lisätään... jne. Etäisyys viimeisestä ”oikeasta” annotaatiosta kasvaa ja funktionaalisuudet eivät enää ehkä vastaa toisiaan.

Domain Walking

- Yhden domainin proteiini käytetään annotoimaan monen domainin proteiinia, kun monen domainin proteiinia kohtaa linjataan lisäproteiini se saa myös väärän funktionaalisuuden.

Funktion määrittäminen: Sekvenssihomologia:

GOTCHA:

BLAST haku -> Tehdään GO luokkia BLAST tuloksista -> $-\log(E\text{-value})$ käytetään BLASTin pisteytykseen -> Yhdistä tulokset geeneistä, jotka kuuluvat samaan GO-luokkaan (puskee tietoa ylemmäs puussa)

- Ei aina spesifejä tuloksia, koska tulos joka oikeasti kuuluisi löytää lehdestä (spesifinen toiminnallisuus) saattaa löytyä hyvin yhäältä rootin läheltä, koska myös kaikki ylemmät luokat sisältävät kaiken funktionaalisuuden alemmista luokista.

Argot:

Korjaa GOTCHA:n ongelmia antamalla myös painotusarvon informaattiosisällölle (kuinka pitkällä ollaan puussa). Clusteroi samanlaisia geeniontologioita.

PANNZER:

Lisää taksonomisen etäisyyden (NCBI-taksonomiapuu), k-nearest,

Vähemmän menestyvät metodit ennustavat suuria GO-luokkia, pieni informaattiosisältö. Hyvin menestyvät metodit: JonesUCL, Argot2, PANNZER.

Exercizes:

FigFAM scan example:

Query Sequence	>ECOLI_0 ...
Hit Family	FIG000885
Hit Family Function	Aspartokinase (EC 2.7.2.4) / Homoserine dehydrogenase (EC 1.1.1.3)

Run batch job with RAST server using Perl script that automates the annotation calls:

```
perl /fs/appl/hippu/users/jpkoskin/RAST/anno_call.pl <
P_wasabiae_SCC3193.fasta > 250_output
```

output →

W5S_0001	Chromosomal replication initiator protein DnaA	55
W5S_0002	DNA polymerase III beta subunit (EC 2.7.7.7)	44
W5S_0003	DNA recombination and repair protein RecF	42
W5S_0004	DNA gyrase subunit B (EC 5.99.1.3)	88
W5S_0005	Fosmidomycin resistance protein Serratia marcescens Db11	42
...		
W5S_0250	Cystine-binding periplasmic protein precursor Serratia marcescens Db11	17

GO Annotation:

Using the p53 protein sequence, try different methods to functionally annotate it.

GOTCHA: It stays on a very basic level, it does not dwell deep into the leaves. It has many categories like: organelle, cell, cellpart, etc. but for example does not have very fine tuned categories like apoptotic signaling pathway is missing.

PFP: Better and more specific than GOTCHA. It had probability values for all the go terms it predicted, the cut-off point was ~20%. In total it made 95 molecular predictions, 260 biological, and 85 cellular.

FFPred: One of the most detailed results, also very visual. Shows High to Low Reliability intervals and probability values. Very specific categories.

Batch annotation with ARGOT2:

```
module load blast+
blastp -outfmt "6 qseqid sseqid evalue " -query only5.fa -db uniprot -out
blast_only5_output -num_threads 10
```

Output:

```
W5S_0001    tr|K4FP15|K4FP15_PECSS 0.0
W5S_0001    tr|DoKCB7|DoKCB7_PECWW 0.0
W5S_0001    tr|J8T1X4|J8T1X4_9ENTR 0.0
W5S_0001    tr|J7KW69|J7KW69_PECCE 0.0
W5S_0001    sp|Q6CYR4|DNAA_ERWCT  0.0
...
```

We had 250 sequences. I initially tried to do this with 5 sequences, but it took forever and showed no sign of completeing, so I redid it with 2.

```
wget ftp://ftp.sanger.ac.uk/pub/databases/Pfam/releases/Pfam26.0/Pfam-A.hmm.gz
gunzip Pfam-A.hmm.gz
module loadh hmmer/3.0
hmmcompress Pfam-A.hmm
hmmsearch --tblout hmm_output_only2 Pfam-A.hmm only2.fasta
```

Results:

```
#                                     --- full sequence ---- --- best 1 domain ---- ---
domain number estimation ----

# target name    accession query name    accession  E-value score bias  E-
value  score bias  exp reg clu  ov env dom rep inc description of target

#-----
-----

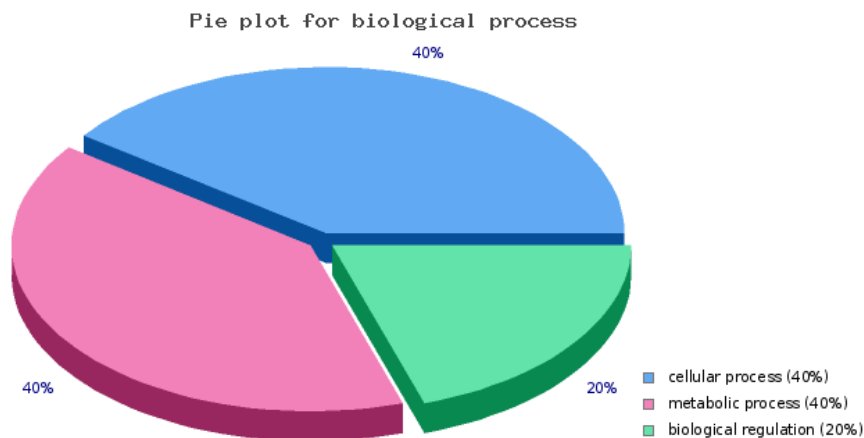
Bac_DnaA        PF00308.13 W5S_0001    -          2e-100 334.8 0.1 3.1e-100
334.2 0.1 1.3 1 0 0 1 1 1 1 Bacterial dnaA protein

Bac_DnaA_C      PF08299.6  W5S_0001    -          3.5e-35 119.5 0.1 9.8e-35
118.1 0.1 1.8 1 0 0 1 1 1 1 Bacterial dnaA protein helix-turn-helix
```

DnaA_N	PF11638.3	W5S_0001	-	6.6e-20	70.3	0.2	1.7e-19	
69.0	0.2	1.8	1 0 0 1 1 1 1	DnaA N-terminal domain				
IstB_IS21	PF01695.12	W5S_0001	-	7.3e-13	48.2	0.0	1.3e-12	
47.4	0.0	1.3	1 0 0 1 1 1 1	IstB-like ATP binding protein				
AAA	PF00004.24	W5S_0001	-	2e-06	27.8	0.0	1.2e-05	
25.3	0.0	2.1	2 0 0 2 2 2 1	ATPase family associated with various cellular activities (\$				
AAA_14	PF13173.1	W5S_0001	-	4.7e-05	23.1	0.2	0.0019	
17.9	0.0	2.4	2 1 0 2 2 2 1	AAA domain				

...

Then I downloaded the files to my computer, compressed them to .zip, and submitted them to ARGOT2. Examples from ARGOT results:

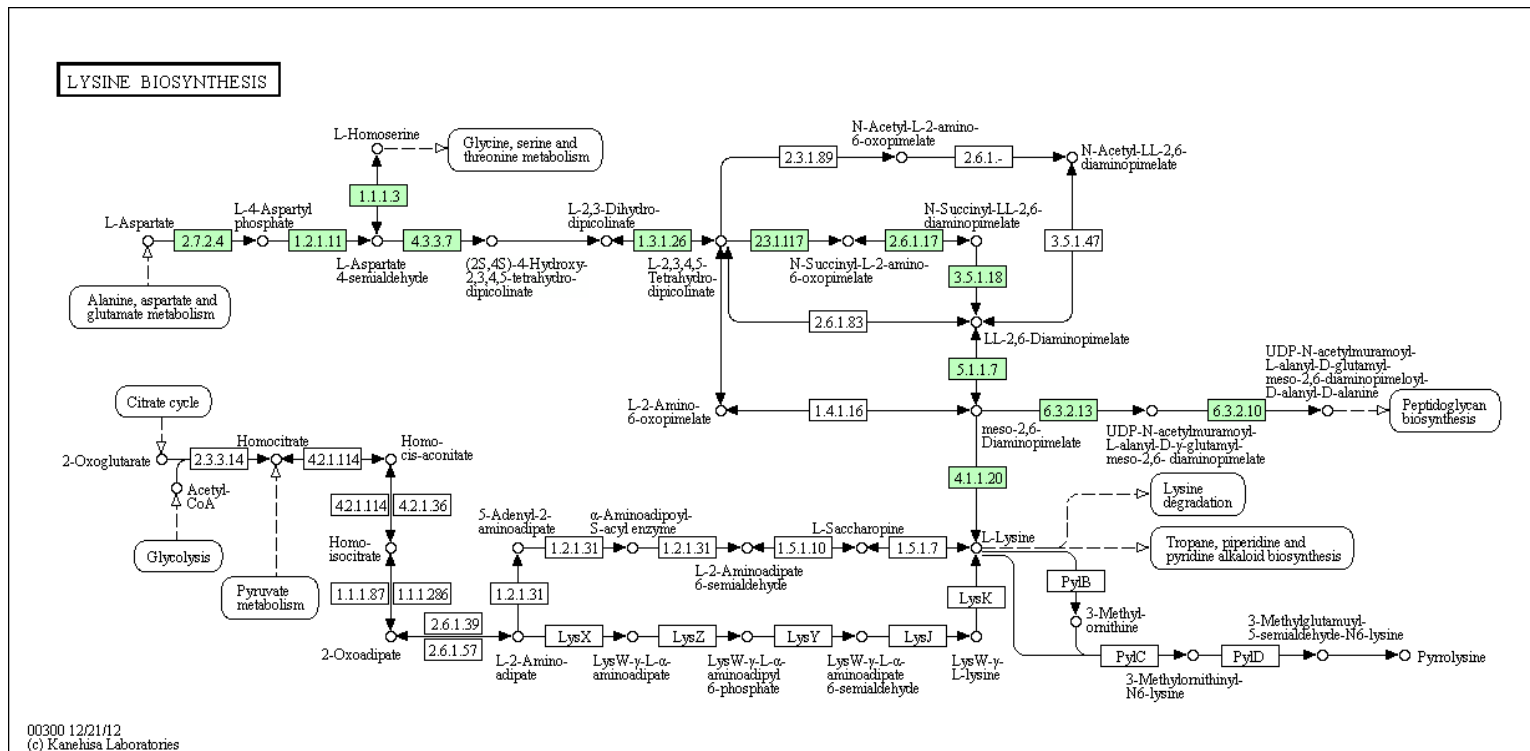


Both proteins had catalytic activity, metabolic activity, only one binding, regulatory, part of a complex.

Pathway mapping:

We had E.Coli sequences and we tried to find out if there's any complete protein pathways that can be built from those sequences. We used KEGG Automatic Annotation Server and mapped the sequences against the KEGG database using BBH.

The most complete, best pathway was 00300 Lysine biosynthesis (14).



There's also three back to back proteins in Cysteine and Methionine Metabolism, from L-Aspartate to L-homoserine. And in the Citrate Cycle from Citrate to Isocitrate and back.

Friday 22.2. RNASeq mapping and analysis

Lecture notes:

FASTQ: 4 riviä per read. @id, seq, + (id), quality-string (ASCII koodi käännettävissä hexa arvoille)

Phred – Q = 30 on 1/1000 väärin. Tämä on hyvä luotettavuusmäärä, alemmat voidaan poistaa.

SAM/BAM: Yksi rivi, tab-välitetty + headerit. CIGAR-tag esim: 8M2I4M ...
8match,2Insertion,4Match... Match on mikä vain emäs, ei välttämättä a-a.

Filtteröinti:

- Average Quality (e. MAPQ (Phred))
- Enemmän kuin haluttu määrä tuntemattomia nukleotideja (N).
- Duplikaatit
- Low complexity regions, ym. huonot alueet. (DUST-score. Paljastaa suurella arvolla esim. polyA-hännät tai muuten repetiiviset alueet)
- Virheet, kontaminaatiot (esim. keskiarvoinen GC-content lajilla verrataan readejen keskipoikkeamaan)
- Readien alut ja loput voivat olla huonolaatuisia, moni trimmausohjelma leikkaa vain lopun.

Burrows-Wheeler transform. Mappers: BWA, Bowtie

- Suffix tree, suffix array. (Substring from any position from the end of the string)
- Suffix tree: Juuresta lähtee jokainen alisekvenssi. Suuri rakenne, vie paljon muistia.
- Suffix array: Järjestetty array, kaikkien suffixien stringi ei ole haettavissa vaan niiden alkuindeksi.
- B-W: suffixi sylinteri (kaikki permutaatiot käydään läpi), viimeinen kolumni sylinterisanasta on BW, järjestetään aakkosjärjestykseen niin, että sisäinen järjestys säilyy (mitkä kirjaimet ympäröivät). Kaksiulotteinen matriisi, josta
 - o Muistiin tallennetaan: BW-arvo ja järjestetty. Eli tunnetaan emäs ja aina sen viereinen kirjain. Koko string voidaan käydä läpi seuraamalla viereistä arvoa päätepisteeseen asti (\$).
 - Etsitään GAT. -> Etsitään järjestetystä arraystä kaikki T:t, (etsi T:t joiden järjestämättömässä arrayssä viereinen (suunta!) kirjain on A), ...

Tophat, käyttää Bowtietä mappamiseen. Etsii introni-eksoni-rajoja.

- e. Readit linjautuu kahteen alueeseen referenssissä. (Jakautuu kahteen eksoniin).

Cufflinks

- Rakentaa kokonaisia transkriptejä.

Exercises:

Load all needed programs:

```
module load samtools
module load bwa
module load tophat2
module load cufflinks
module load fastx
```

Filter reads that have phred quality of under 30 and base quality under 33:

```
fastq_quality_filter -Q 33 -q 30 -i
Cinx_PoolA_RNAseq_GCCAAT_L001_R2_001_FILTERED.fastq -o
Cinx_PoolA_RNAseq_GCCAAT_L001_R2_001_FILTERED.fastq_high_Q

fastq_quality_filter -Q 33 -q 30 -i
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq -o
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq_high_Q
```

Mapping with BWA:

Bwa indexing, bwa aln generates SA coordinates for the reads, bwa sampe makes paired end alignment for the reads:

```
bwa index Cinxia_scaffold.fasta

bwa aln Cinxia_scaffold.fasta
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq_high_Q >
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq_high_Q.sai

bwa aln Cinxia_scaffold.fasta
Cinx_PoolA_RNAseq_GCCAAT_L001_R2_001_FILTERED.fastq_high_Q >
Cinx_PoolA_RNAseq_GCCAAT_L001_R2_001_FILTERED.fastq_high_Q.sai

bwa sampe Cinxia_scaffold.fasta
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq_high_Q.sai
Cinx_PoolA_RNAseq_GCCAAT_L001_R2_001_FILTERED.fastq_high_Q.sai
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq_high_Q
Cinx_PoolA_RNAseq_GCCAAT_L001_R2_001_FILTERED.fastq_high_Q > aln.sam
```

Using samtools convert to bam:

```
samtools view -bS aln.sam > aln.bam
```

If you want to view the .bam file, you need to further sort and index it to view it in Artemis.

```
samtools sort aln.bam sorted
samtools index sorted.bam
```

If my reads were longer than 200bp, I'd have to use BWA-SW or BWA-MEM. BWA-MEM is the latest and the recommended one.

Mapping with Bowtie:

Alignment and indexing with Bowtie:

```
bowtie2 -x Cinxia_scaffold.index -1  
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq_high_Q -2  
Cinx_PoolA_RNAseq_GCCAAT_L001_R2_001_FILTERED.fastq_high_Q -S bowtie.sam  
  
bowtie2-build Cinxia_scaffold.fasta Cinxia_scaffold.index
```

Discovering exon junctions with Tophat:

```
tophat Cinxia_scaffold.index  
Cinx_PoolA_RNAseq_GCCAAT_L001_R1_001_FILTERED.fastq_high_Q,Cinx_PoolA_RNAseq_GCC  
AAT_L001_R2_001_FILTERED.fastq_high_Q
```

To view the result file: go to tophat_out folder and check your result file accepted_hits.bam.

If you want to turn the .bam file to a .sam file:

```
samtools view -h accepted_hits.bam > accepted_hits.sam  
  
For Artemis:  
  
samtools sort accepted_hits.bam sorted_accepted_hits  
samtools index sorted_accepted_hits.bam  
  
Using TView:  
  
samtools tview sorted_accepted_hits.bam ../Cinxia_scaffold.fasta
```

Most reads are aligned towards the end of the reference, most of the reference is blank. TopHat suggests splice sites in places where there are peaks in coverage of reads.

Cufflinks:

```
cufflinks -o cufflinks_output accepted_hits.bam  
../../../../v/users/ahola/bin/maker/bin/cufflinks2gff3  
tophat_out/cufflinks_output/transcripts.gtf > transcripts.gff3
```

--overlap-radius maximum gap size to fill between transfrags (in bp) [default: 50]

If the gap between two units is larger than n, they are combined.

I tried using it with 10 and 100.

```
cufflinks --overlap-radius 100 -o cufflinks_with_overlap100
accepted_hits.bam

../../../../v/users/ahola/bin/maker/bin/cufflinks2gff3
tophat_out/cufflinks_with_overlap100/transcripts.gtf >
transcripts100.gff3

cufflinks --overlap-radius 10 -o cufflinks_with_overlap10
accepted_hits.bam

../../../../v/users/ahola/bin/maker/bin/cufflinks2gff3
cufflinks_with_overlap10/transcripts.gtf > transcripts10.gff3
```

But I didn't notice any difference in Argo. ? ☹

Adding gene predictors as input to Cufflinks:

<code>-G/--GTF</code>	quantitate against reference transcript annotations
<code>-g/--GTF-guide</code>	use reference transcript annotation to guide assembly

Masking:

<code>-M/--mask-file</code>	ignore all alignment within transcripts in this file
-----------------------------	--

Files Cufflinks generates:

transcripts.gtf : file to be visualized

isoforms.fpkms_tracking : Expression values for the transcripts expressed

genes.fpkms_tracking: Expression values for the genes expressed

skipped.gtf : reads that weren't mapped

Friday 1.3. Comparative Sequence Analysis

Lecture Notes:

Problems:

- sequence sampling (what sequences we should choose? is it worth it to have many very similar sequences (due to evolutionary time) or should there be more variety?)
- what level of conservation is statistically significant?

Solutions:

- codon models: separates mutation process from selection, corrects for relatedness
 - o selection happens in aminoacids.
 - o Completely different codons rated 0 to empty the matrix and reveal codon clusters (synonymous substitutions -> same aminoacid).
 - o omega value for mutations: harmful < 1 (reduces value), neutral = 1, w > 1 (increases value -> big signal). Most mutations are harmful.
 - Omega estimation:
 - Branch: e.g. gene duplication -> new function.
 - Site: Assumes that all sequences have same function

Maximum Likelihood:

Using observation calculate the probability of p. Maximise (L) by finding the value of p that maximizes the function. (occurrences of p)/(all occurrences).

- In phylogenetics: try to find tree that maximizes the probability of the sequence data -> produces a “model” that best describes the data.

Substitutions are stochastic processes:

- mechanistic or empirical models to evaluate evolutionary distance between two sequences.
 - o mechanistic model: matrix with values for substitutions (A->T == T->A).
 - o Amino acid models:
 - Frequencies estimated from data.
 - f.ex. PAM/BAM
 - o For nucleotide sequences:
 - Take into account different things like: transitions/transversions, GC% etc...
 - HKY85 is a good model. Different rates for transitions and transversions.

Method: SLR

- Sitewise
- Estimates common parameters from full data. Assumes all the sites are neutral and tests if that site different than 1. -> Produces confidence intervals and point estimates.
 - o Results sites that are under selection (pos/neg).
 - If upper limit is under 1, under negative selection.
 - If upper limit is above 1, under positive selection.

Problems with SLR:

- Produces an alignment with no gaps (usually viral sequences have lots of gaps).
- Alignment is an estimate

Genetic evolutionary distance:

- needs to be corrected for multiple hits (Jukes and Cantor)
- Correction increases when p increases.

INDELible = data simulator.

- File ready for indel with preset values created by Ari Löytynoja.
 - o [MODEL] 50 bins with 0.02 probability, 50 values of omega
 - o [treedepth] 1.4396 (evolutionary time, lower number lower differences, less gaps etc.)
 - o [PARTITIONS] 300
 - o Output -> Fasta

Exercizes:

Reduce memory requirement for BioLinux to orange range.

Login to Biolinux -> gz/tar file.

Biolinux is basic Ubuntu.

HIVPol

Used the visualization programs like Jalview and Seaview in biolinux to visualize alignments and Forester and Treeview to view trees from the HIVpol dataset. The pdf files can be visualized with “evince pdfname”.

I tried to understand the plotSlr.R perl-script, but I don't know any R. All I can say is that it saves to a pdf-file. Using the R-script:

```
R --vanilla < plotSlr.R
```

Execute indelible with command “indelible”. LOG.txt describes what values were used as input (ready file made by Ari) and what the values are. All the values are unique (generated random).

bp_translate_seq jordan_small_TRUE.fas > jordan_small_TRUE_pep.fas to make the nucleotide sequence (codon alignment) to a protein sequence.

OR

seaview Jordan_small_TRUE.fas

Props -> View as Proteins -> Save

```
Look at the file jordan_small_RATES.txt that lists the rate categories used
across the sequence sites. Category 0 is the most conserved and categories
47--49 are the ones with omega greater than 1. On command line, try
commands
'grep ^[1-9] jordan_small_RATES.txt |sort -k 2 -n|less' and
'grep ^[1-9] jordan_small_RATES.txt |sort -k 2 -n -r|less'
```

```
grep ^[1-9] jordan_small_RATES.txt |sort -k 2 -n|less >
jordan_small_RATES_sorted.txt
```

➔ sort by column “Class” (smallest to largest)

➔ 126 0 1

➔ 144 0 1 INSERTION

```
➔ 201 0 1 INSERTION
➔ 217 0 1 INSERTION
```

```
grep ^[1-9] jordan_small_RATES.txt |sort -k 2 -n -r|less
- sort by column "Class" (2) (largest to smallest)
```

SLR data

Convert format with PRANK:

```
prank -convert -d=jordan_small_TRUE.fas -o=jordan_small_TRUE -f=paml
```

Good way of viewing information is with awk:

```
awk 'BEGIN{OFS="\t";print "Site\tOmega\tlower\tupper";} /[0-9]/ &&
$5>1 {print $1,$4,$5,$6}' out.res
```

Site	Omega	lower	upper
20	99.0000	2.1416	99.0000
85	2.1453	1.2072	3.7392
176	1.9224	1.0442	3.4083
234	1.7717	1.0204	2.9627
433	1.9598	1.0620	3.5245
671	2.6573	1.3577	5.2959
680	2.2348	1.1610	4.4409

Here there's a table created with fields Site/Omega/Lower/Upper and the lines where the field Lower is (\$5) larger than 1 are chosen and then the correct fields are printed to the table (\$1, \$4, \$5, \$6).