2.1 Calculate basic statistics

The next part is an analysis performed on a FASTA formatted file containing complete genomic DNA (`*.dna`), not genes or proteins. Calculate the AT content (`Per.AT`), number of replicons (`ContigCount`), deviation of AT across replicons (`StDevAT`), percentage of unknown bases (`Per.Unknowns`) and total size in bp (`TotalBases`).

Listing 2.1: Calculate basic genomic DNA statistics

```
# Syntax:
$ stats_genomeDNA <name>.gbk.dna
# Example:
$ stats_genomeDNA Neisseria_meningitidis_Z2491_ID_AL157959.gbk.dna
Filename    TotalBases: Per.AT: StDevAT:    ContigCount:    Per.Unknowns:   Per.LargestSeq  N25 N50 N75
Neisseria_meningitidis_Z2491_ID_AL157959.gbk.dna    2184406 48.19   0.0000  1   0.0000  100.000 2184406
    2184406 2184406
```

Output is by default written to the screen. You can run the command in a `for`-loop and store the data from each genome in a text file.

Listing 2.2: Calculate basic genomic DNA statistics - loop

```
$ for x in *.gbk.dna
> do
> echo $x
> stats_genomeDNA $x >> genomeStats.all
> done
```

Note the ">>" signs, for appending to a file. When using redirect, ">", a file is created if not already found or overwritten if already found. When appending, the output is added if the file is not found or appended if the file is already found. Copy the genome statistics into a spreadsheet (Gnumeric).

2.1.0.2 Exercises

1. Calculate basic statistics for all `*.gbk.dna` files and store results in file
2. Which genome/organism has the highest AT content?
3. Which genome/organism has the lowest AT content?
4. Which genome/organism is the largest?
5. Which genome/organism is the smallest?

2.2 Genome atlas

The genome atlas presented here is an implementation of the atlas presented earlier by Jensen et al. 1999. Below is a short description of each of the parameters shown in the genome atlases. Color scales for all parameters follow the same system. The DNA sequence is read and an output file is generated for the various calculated parameters. For each nucleotide in the genome a numerical value is calculated. This file is then read by the `GeneWiz` program, which calculates the average and standard deviation for each parameter, if the average value of the window is more than 3 standard deviations on either side of the overall average the window is maximally colored. In order to plot the data on a circular map a "window size" is used for longer genomes, which effectively smooths the data for better graphics. For the parameters `Stacking Energy`, `Position Preference` and `Intrinsic Curvature`, the window is 0.002 x genome length. The window is 0.001 x genome length for `Percent AT` and `GC skew`. Each of these are calculated separately, wrapped into a pipeline and visualized in a circular plot, called an atlas. The gene annotations are taken directly from a GenBank coding regions; if no such information is found the `CDS-/+` lanes will be blank. The following lists explanations to each of the lanes in a genome atlas: `Percent AT` is the percent of A's and T's in the genome. `GC skew` is calculated as *((G-C)/(G+C))*, with a window size of 10 000 bp and is useful for determining the origin and terminus of replication [7, 13]. `Global Direct Repeats` and `Global Inverted Repeats` refer to a sequence that is present in at least two copies on the same or opposite strands, respectively. `Intrinsic Curvature` is a measure of DNA curvature and is calculated using the CURVATURE program [11, 2]. The values are scaled from 0 (e.g. no curvature) to 1, which is the curvature of DNA when wrapped around the nucleosome. `Stacking Energy` is derived from the di-nucleotide values provided by Ornstein et al. [8]. The scale is in kcal/mol, and the di-nucleotide values range from -3.82 kcal/mol (will unstack easily) to -14.59 kcal/mol (difficult to unstack). A positive peak in base-stacking (i.e., numbers closer to zero) reflects regions of the helix which would de-stack or melt more readily. Conversely, minima (larger negative numbers) in this plot would represent more stable regions of the chromosome. `Position Preference` is a measure of preferential location of sequences within nucleosomal core sequences [10]. The tri-nucleotide values range from essentially zero (0.003, presumably more flexible), to 0.28 (considered rigid). Since very few of the tri-nucleotide have values close to zero (e.g. little preference for nucleosome positioning), this measure is considered to be more sensitive towards the low ("flexible") end of the scale.

Now you will construct a genome atlas (See Figure 3.2). A genome atlas can be made from a GenBank file and uses the gene/protein annotations published with the genome DNA sequence. It is important to have only one replicon in your GenBank file (count number of `LOCUS` if you are not sure). The reason for this is that multiple replicons cannot be visualized in a single atlas and the program will not execute. In order to construct an atlas, the DNA sequence is scanned for all kinds of patterns. This means that it takes time to prepare the files necessary for a genome atlas. For each genome, investigate if the genome contains more than replicon (use `grep` to locate how many `LOCUS` lines are in the GenBank files). Move to the directory where you have the GenBank files. First, it is necessary to construct a set-up file from which to draw the atlas. This file is called a `configuraton` file and holds informations on which calculations to run and which files to use as input. The file is then used as an input to the program `atlas` which draws the circular genome

atlas figure.

Listing 2.3: Create genome atlas

```
# Syntax:
$ atlas_createConfig -ref <name>.gbk
# The file <name>.gbk.atlas.cf is automatically generated

$ atlas
# USAGE:
#   atlas -f <genbankFile> -c <configFile> -o <outputFile> -tidy
#
# Creates a genome/blast atlas of the organism given in <genbankFile> based on the setup in the config
    file.
#
# -f    The name of the gbk file to use <REQUIRED>
# -c    The name of the config file created by atlasCf <REQUIRED>
# -o    The name of the plot file to create.#   If not specified no plot will be made.

$ atlas -f <name>.gbk -c <name>.gbk.atlas.cf -o <name>.gbk.atlas.ps

# Example:
$ atlas_createConfig -ref Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk

# The file Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.atlas.cf is generated
# Following command must be one one command-line
$ atlas -f Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk
-c Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.atlas.cf
-o Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.atlas.ps
```

Using `ls -l` you will see that a number of files have been generated. These should be kept as they will be used later. Find the postscript file `*.ps` in the file-manager and open it, save as a PDF. To do a zoom of a specific region, open the file called <name>.genomeatlas.cf and add the line described below. Note that this procedure should be run in the same directory as the whole chromosome (not-zoomed) atlas, as it uses the same files.

Listing 2.4: Create zoom of genome atlas

```
$ cp <name>.genomeatlas.cf <name>.zoom.genomeatlas.cf
$ gedit <name>.zoom.genomeatlas.cf
# Syntax:
circlesection <start> <end>;
# Example:
circlesection 515000 535000;
# This line should be added under the line that looks like this:
circletics auto;
# Save the file as something else, like
# Now re-run the atlas picture
$ genewiz -p <name>.zoom.genomeatlas.ps <name>.zoom.genomeatlas.cf
```

Find the postscript file `*.ps` in the file-manager and open it. Save the plots as a PDF format and open a word processor (Word, Pages) or presentation software (PowerPoint, Keynote) on your LOCAL computer. Get the PDF from the shared folder and put the picture into your presentation.

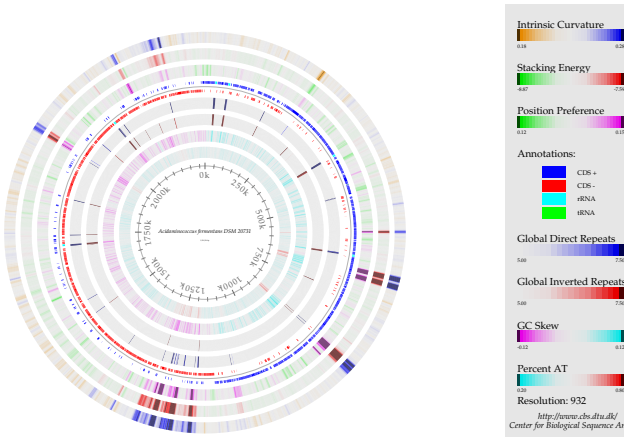

Figure 2.1: **Genome atlas, DNA structures.** A DNA structural atlas. DNA, RNA and gene annotations are from the published GenBank data. Each lane of the circular atlas shows a different DNA feature. From the innermost circle: size of genome (axis), percent AT (red = high AT), GC skew (blue = most G's), inverted and direct repeats (color = repeat), position preference, stacking energy and intrinsic curvature.

2.2.1 Exercises

1. Create a genome atlas for 2 different replicons, plasmids/chromosomes
2. Create a zoom of one of the atlases
3. Save plots as PDF and insert into presentation

3.1 Amino acid and codon usage

The amino acid and codon usage is calculated using `BioPerl` modules, and is a simple calculation of the fraction of each amino acid or codon count of the total count of amino acids or codons. The bias in third position is found by counting the number of each base on each position in each codon, divided by the total number of codons. The bias in the third position between `G/C` and A/T is calculated as `sum(GC)-sum(AT)`, so that 100 % GC in third codon position is +1 and -1 for 100 % AT. The calculations have been implemented in a program called `stats_usage` and takes gene FASTA files as input.

Listing 3.1: Calculate amino acid and codon usage

```
# Syntax:
stats_usage <name>.gbk.fna /usr/bin/gnuplot
# Example:
stats_usage Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.fna /usr/bin/gnuplot
# Loop
$ for x in *fna; do stats_usage $x /usr/bin/gnuplot; done
```

The output is a number of images files and a text file. Open the image files in the file manager and investigate the content. Use `gedit` to investigate the text file.

Listing 3.2: Calculate amino acid and codon usage - output

```
# Image files
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.fna.aa-usage.pdf
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.fna.codon-usage.pdf
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.fna.basicInfoPS.pdf
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.fna.all.pdf
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.fna.bias.pdf
# Text file
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.fna.CodonAaUsage
```

3.1.1 Exercises

1. Calculate the codon and amino acid usage for all genomes
2. Create a folder for the PDF files (`PDF`) and one for the usage text files (`USAGE`)
3. Move all `*pdf` files to the `PDF` folder and all `*.CodonAaUsage` to the `USAGE` folder

3.1.2 Compare amino acid and codon usage

Comparing the amino acid and codon usage between many genomes also usually involves clustering genomes with similar usage. It is therefore useful to compare these numbers using a so-called heat-map constructed in `R` (The R Project for Statistical Computing). First we prepare the data from the `*CodonAaUsage` files and collect them into one file. You can of course choose to only include some of your file if you wish to do a smaller comparison, or more data if you want.

Listing 3.3: Comparing amino acids and codon usage - preparations

```
grep aa *CodonAaUsage > aaUsage.all
```

```
grep Total *CodonAaUsage > statistics.all
cut -f2,3,4,5,6,7,8 statistics.all > tmp.all
mv tmp.all statistics.all
grep codon *CodonAaUsage > codonUsage.all
```

Start R in the directory where the statistics files are stored and read in the data. Drawing heat-maps of amino acid and codon usage (See Figure 3.1):

Listing 3.4: Comparing codon usage - heat-map

```
$ R # This is typed on the prompt
# As a result R opens it's own prompt with the ''>'' symbol in the start of the line
# On this prompt only R commands work and NOT the normal unix commands

library(gplots)
codon <- read.table("codonUsage.all")
colnames(codon) <- c( 'Name', 'codon', 'score', 'count')
codon <- codon[1:3]
test <- reshape(codon, idvar="Name", timevar="codon", direction="wide")
codonMatrix <- data.matrix(test[2:length(test)])
rownames(codonMatrix) <- test$Name

# R allows you to run one long command like the following
codon_heatmap <- heatmap.2(codonMatrix, scale="none", main="Codon usage",
xlab="Codon fraction", ylab="Organism", trace="none", margins=c(8, 25)) # Command finished

dev.print(postscript, "codonUsage.ps", width = 25, height=25)
dev.off()
```

Listing 3.5: Comparing amino acids usage - heat-map

```
library(gplots)
aa <- read.table("aaUsage.all")
colnames(aa) <- c( 'Name', 'aa', 'score')
test <- reshape(aa, idvar="Name", timevar="aa", direction="wide")
aaMatrix <- data.matrix(test[2:length(test)])
rownames(aaMatrix) <- test$Name

# R allows you to run one long command like the following
stat_heatmap <- heatmap.2(aaMatrix, scale="none", main="Amino acid usage", xlab="Amino acid fraction",
ylab="Organism", trace="none", margins=c(8, 25), col = cm.colors(256))  # Command finished

dev.print(postscript, "aaUsage.ps", width = 25, height=25)
dev.off()
```

3.1.2.1 Exercises

1. Create heat-maps for amino acid and codon usage
2. Save images as PDF and insert into presentation

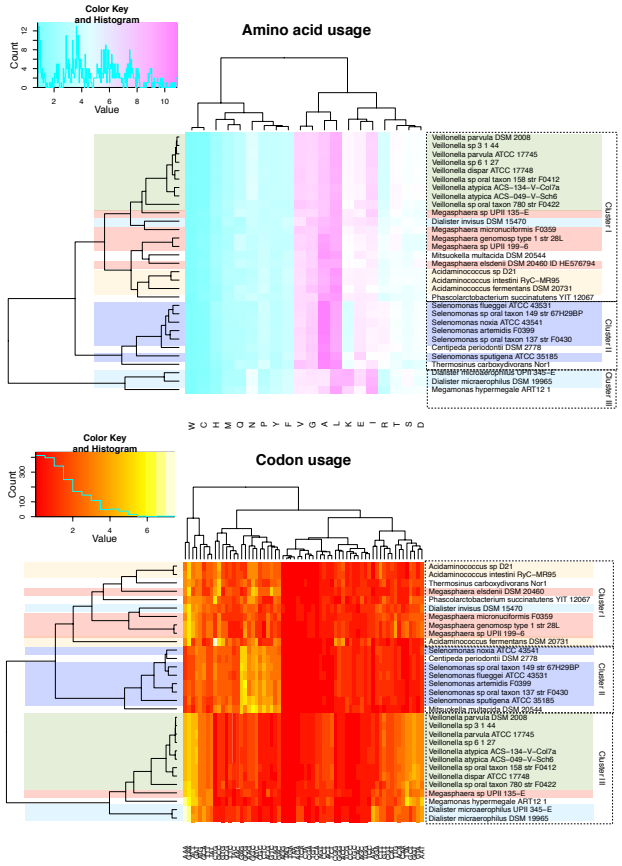

Figure 3.1: **Amino acid and codon usage heat-maps.** Amino acid and codon usage were for 31 genomes calculated based on the genes identified by gene finding (Prodigal). The percentage of codon and amino acid usage was plotted in two heat-maps using `R`. The heat-maps were clustered in 2D, thus reordering the organisms and the amino acids/codon to show the shortest distance between them. Dendograms are drawn for both and can be used to visualize the difference in usage between organisms.