**1) Mapping based on nucleotide translation in 6 frames**

Pros:

- Maps reads based on protein sequences, which are generally more conserved than nucleotide sequences -> can map sets of reads that are more divergent from current model organisms

Cons:

- Doesn’t map UTRs very well

- Computationally intensive (translation in 6 frames)

**BioPerl**

BioPerl (and BioPython) has useful tools to translate nucleotide sequences in 6 frames to be used in alignment programs like BLAT, BLAST, or PSI-BLAST.

A short example on how to use a BioPerl module is given in the script 6frames.pl

**2) Mapping with an aligner that allows for divergent reads/iterative mapping**

Pros:

- Mapping can include UTR regions

- Computationally not more intensive than regular mapping

Cons:

- Introduces mapping bias for regions that are conserved at the nucleotide level

Stampy:

- Maps single and paired Illumina reads to a reference genome/transcriptome

- High sensitivity for indels and divergent reads, up to 10-15%

- Input: Fastq and Fasta; gzipped or plain; SAM and BAM

- Output: SAM, Maq's map file

Currently the linux x86\_64 platform is supported, and support for Mac OS-X 10.6 (x64\_64 only) is experimental. Stampy needs Python version 2.6 or 2.7. Both 2-byte and 4-byte Unicode encodings are supported. For Mac, only Python 2.6 with 2-byte Unicode is supported.

Building a genome (.stidx) file (or a transcriptome file):

./stampy.py --species=mouse -G mouse /data/genomes/mouse/\*.fa.gz

Building a hash (.sthash) file:

./stampy.py -g mouse -H mouse

Single-end mapping:

./stampy.py -g mouse -h mouse --inputformat=fastq -f sam -o test\_output.txt -M read.fastq

Paired-end mapping:

./stampy.py -g mouse -h mouse --inputformat=fastq -f sam -o test\_output.txt -M read1.fastq read2.fastq

--substitutionrate = 0.03

Modify this parameter if you want to allow more mismatches during the alignment. Default is 0.001 (0.001 substitutions per site).

**Iterative mapping**

If there are no closely related organisms to the one from which the sequences derive, one can align reads to multiple reference transcriptomes and combine the results afterwards.

Example: star-nosed mole RNASeq data

- No genome from the order Insectivora

- Mapping sequences to two unrelated transcriptomes: *Homo sapiens* and *Mus musculus*.

We will be working with 1000 sequences derived from ganglia of the star-nosed mole (test.txt).

The alignment with transcriptomes from *Homo sapiens* and *Mus musculus* has already been performed with commands similar to the ones shown below.

./stampy.py -g human -h human --inputformat=fastq -f sam -o test\_human.txt -M ../CGRL\ 2012/test.txt

./stampy.py -g mouse -h mouse --inputformat=fasta -f sam -o test\_human\_mouse.txt -M ../CGRL\ 2012/whatever\_left\_from\_human\_mapping.txt

We will use the script stampy.pl to count the number of reads aligned to each reference sequence id present in the reference transcriptome (the script stampy\_complete.pl contains a couple of routines that also map automatically the fastq reads to the genome, it requires stampy being installed on your machine).

Stampy.pl produces two files: id\_tissue\_1\_HUMAN and id\_tissue\_1\_MOUSE

Each file contains the reference sequence id, the number of reads mapped to that id, and the gene length.

To combine the results between different species, we need to know what genes are homologous in different species (with a certain approximation).

We can run the script combine\_homologues.pl: this script reads the files generated from stampy.pl and produces a few output files:

in the log subdirectory:

- Hs\_tissue\_id\_no\_Entrez.txt/Mm\_tissue\_id\_no\_Entrez.txt

These files contain the reference sequence ids for which there is no entry available in the Entrez database

- Hs\_tissue\_id\_no\_Entrez\_description.txt/Mm\_tissue\_id\_no\_Entrez\_description

These files contain the reference sequence ids for which there is no description available in the Entrez database

Reference sequence IDs contained in these files correspond mostly to entries classified as non coding RNAs, non very well annotated sequences, or entries in old databases that are not valid anymore with the updated versions of the Entrez and UniGene databases.

in the main directory:

- tissue\_ToMouse

For each entry, the file contains:

Gene ID gene id (from NCBI Gene database)

Homologous gene ID id of the homologous gene

Homologous taxonomy ID taxonomy ID of the second species considered

Taxonomy ID taxonomy ID of the reference species

NCBI Reference sequence reference sequence id(s) to which reads aligned

Full name name of the gene

Symbol official symbol

Number of reads number of reads aligned

Gene length length of the gene in the reference species

- tissue\_forDESeq.txt

DESeq is a useful program used to analyze results from aligned reads. This file contains the gene symbol and the number of reads aligned to that gene (based on tissue\_ToMouse file) that can be used to load in R for DESeq analysis.

**The script**

How do we summarize the results obtained from aligning reads to transcriptomes belonging to different species?

Summary:

Reference seq id species 1 -> gene id species 1

\

same HomoloGene group

/

Reference seq id species 2 -> gene id species 2

**gene2accession**

<ftp://ftp.ncbi.nih.gov/gene/DATA/gene2accession>

file: gene2accession.gz

from the NCBI README

“This file is a comprehensive report of the accessions that are related to a GeneID. It includes sequences from the international sequence collaboration, Swiss-Prot, and RefSeq.“

It allows us to link the Reference Sequence id (e.g. NM\_018117) to its Gene id (e.g. 55717). Multiple Reference Sequence ids can be linked to the same Gene id (e.g. multiple isoforms).

The script uses two smaller files derived from gene2accession: g2acc\_tax\_select\_minimal\_9606 and g2acc\_tax\_select\_minimal\_10090, which contain information about the taxonomy id of the species (9606 = Homo sapiens, 10090 = Mus musculus), the Gene id and all of the Reference Sequence ids connected to it.

To make these files, one has to parse the gene2accession file downloaded from the NCBI using the script gene2accession\_parser.pl.

**gene\_info**

<ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia>

files:

Homo\_sapiens.gene\_info.gz

Mus\_musculus.gene\_info.gz

The file contains information about the gene, including its symbol, LocusTag, synonyms, dbXrefs, chromosome location, map location, a descriptive name, and type of gene.

It allows us to link the Gene id (e.g. 55717) to its symbol (e.g. WDR11) and a more useful description of the gene (e.g. WD repeat domain 11).

The script uses two smaller files derived from Homo\_sapiens.gene\_info and Mus\_musculus.gene\_info: Hs\_minimal and Mm\_minimal, which contain only the Gene id, the official symbol, and gene description.

To make these files, one has to parse the gene\_info files downloaded from the NCBI using the script gene\_info\_parser.pl (the script will parse both files, if you only need one change the appropriate lines in the script).

**HomoloGene**

<ftp://ftp.ncbi.nih.gov/pub/HomoloGene/current>

file:

homologene.xml.gz

In the HomoloGene database, genes from different species are clustered in orthologous and homologous groups based on sequence similarity. The file homologene.xml.gz contains information about the clusters such as gene and protein links and distance analysis.

It allows us to link the gene id to a HomoloGene cluster, and connect it to genes in other species that belong to the same cluster (if they exist).

The script uses two files derived from homologene.xml, homologene.data.nt.9606 and homologene.data.nt.10090, which contain:

HomoloGene cluster id HomoloGene cluster number

Taxonomy id of the species 10090 for Mus musculus, 9606 for Homo sapiens

Gene id id of the gene in the Gene database

Symbol gene symbol

Description gene description

Nucleotide gi gi number (NCBI tracking number)

Reference sequence id one reference sequence id for the gene

To make these files, one has to parse the homologene.xml file downloaded from the NCBI using the script homologene\_parser.pl (the script will generate both files, if you only need one change the appropriate lines in the script).