Lecture 13 - Introduction to Bioinformatics and Installation

What does bioinformatics mean to you?

At a practical level, what is bioinformatics?

At a practical level, what is bioinformatics?

- installing tools
- figuring out how to use tools
- reformating files
- searching for patterns

Searching for patterns

Looking for genes in a bacterial genome:

- Reference sequences for gene of interest
- The bacterial genome to search

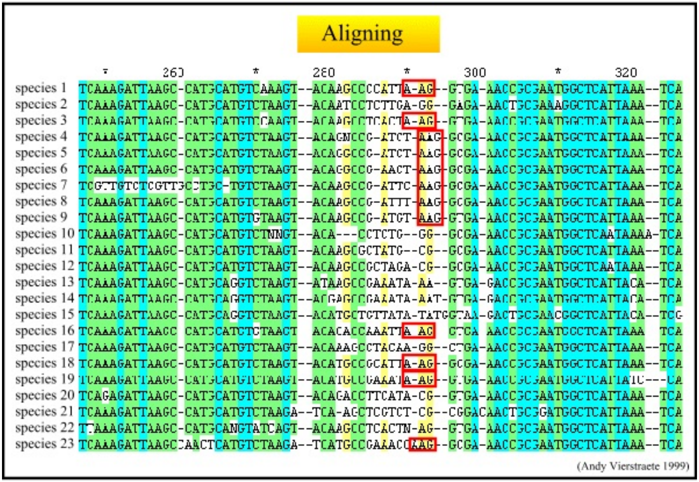
Reference sequences for gene of interest

Because genomes are often large, complicated, and prone to change (indels), we need a few bioinformatics tools to be able to conduct sequence comparisons quickly and in a probabilistic framework.

- Muscle: sequence aligner
- hmmer: search tool

Muscle: Sequence alignment

Sequence alignment attempts to evaluate the level of sequence conservation or *homology* amongst a set of sequences. Once aligned the sequences can be used to evaluate genetic change and to infer evolutionary history.



Aligning

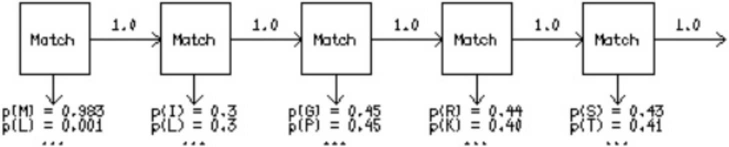
```
                   *        260         *       280          *      300         *       320
species 1  TCAAAGATTAAGC-CATGCATGTCAAAGT--ACAAGCCCCATTA-AG--GTGA-AACCGCGAATGGCTCATTAAA--TCA
species 2  TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAATCCTCTTGA-GG--GAGA-AACTGCGAAAGGCTCATTAAA--TCA
species 3  TCAAAGATTAAGC-CATGCATGTCCAAGT--ACAAGCCTCACTA-AG--GTGA-AACCGCGAATGGCTCATTAAA--TCA
species 4  TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAGNCCG-ATCT-AAG-GCGA-AACCGCGAATGGCTCATTAAA--TCA
species 5  TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAGGCCG-ATCT-AAG-GCGA-AACCGCGAATGGCTCATTAAA--TCA
species 6  TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAGGCCG-AACT-AAG-GCGA-AACCGCGAATGGCTCATTAAA--TCA
species 7  TCGTTGTCTCGTTGCCTGC-TGTCTAAGT--ACAAGCCG-ATTC-AAG-GCGA-AACCGCGAATGGCTCATTAAA--TCA
species 8  TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAAGCCG-ATTT-AAG-GCGA-AACCGCGAATGGCTCATTAAA--TCA
species 9  TCAAAGATTAAGC-CATGCATGTGTAAGT--ACAAGCCG-ATGT-AAG-GTGA-AACCGCGAATGGCTCATTAAA--TCA
species 10 TCAAAGATTAAGC-CATGCATGTCTNNGT--ACA---CCTCTG--GG--GCGA-AACCGCGAATGGCTCAATAAAA-TCA
species 11 TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAAGCGCTATG--CG--GCGA-AACCGCGAATGGCTCATTAAA--TCA
species 12 TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAAGCCGCTAGA-CG--GCGA-AACCGCGAATGGCTCAATAAA--TCA
species 13 TCAAAGATTAAGC-CATGCAGGTCTAAGT--ATAAGCCGAAATA-AA--GTGA-GACCGCGAATGGCTCATTACA--TCA
species 14 TCAAAGATTAAGC-CATGCAGGTCTAAGT--ACGAGCCGAAATA-AAT-GTGA-GACCGCGAATGGCTCATTACA--TCA
species 15 TCAAAGATTAAGC-CATGCAGGTCTAAGT--ACATGCTGTTATA-TATGGTAA-GACTGCGAACGGCTCATTACA--TCG
species 16 TCAAACATTAACC CATGCATCTCTAACT  ACACACCAAATTA AC  GTCA AACCGCGAATCGCTCATTAAA  TCA
species 17 TCAAAGATTAAGC-CATGCATGTCTAAGT--ACAAAGCCTACAA-GG--CTGA-AACCGCGAATGGCTCATTAAA--TCA
species 18 TCAAAGATTAAGC-CATGCATGTCTAAGT--ACATGCCGCATTA-AG--GCGA-AACCGCGAATGGCTCATTAAA--TCA
species 19 TCAAAGATTAAGC-CATGCATGTCTAAGT--ACATGCCGAAATA-AG--GTGA-AACCGCGAATGGCTCATTATC---CA
species 20 TCAGAGATTAAGC-CATGCATGTCTAAGT--ACAGACCTTCATA-CG--GTGA-AACCGCGAATGGCTCATTAAA--TCA
species 21 TCAAAGATTAAGC-CATGCATGTCTAAGA--TCA-AGCTCGTCT-CG--CGGACAACTGCGGATGGCTCATTAAA--TCA
species 22 TTAAAGATTAAGC-CATGCANGTATCAGT--ACAAGCCTCACTN-AG--GTGA-AACCGCGAATGGCTCATTAAA--TCA
species 23 TCAAAGATTAAGCCAACTCATGTCTAAGA--TCATGCCGAAACCAAG--GCGA-AACCGCGAATGGCTCATTAAA--TCA
```

(Andy Vierstraete 1999)

Hmmer: Profile Hidden Markov Models (HMM) for searching

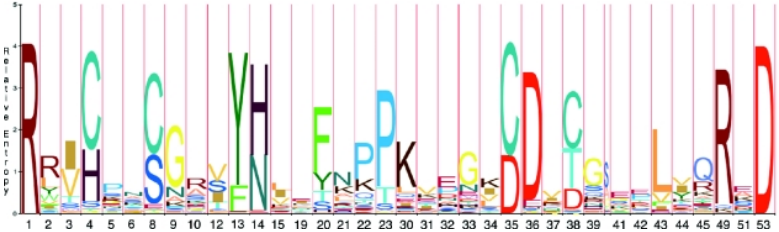
HMMs have been used to conduct flexible and fast sequence comparisons in a probabalistic framework.

For a simple model that does not allow for insertions or deletions, the probability of a given amino acid or nucleic acid is given.



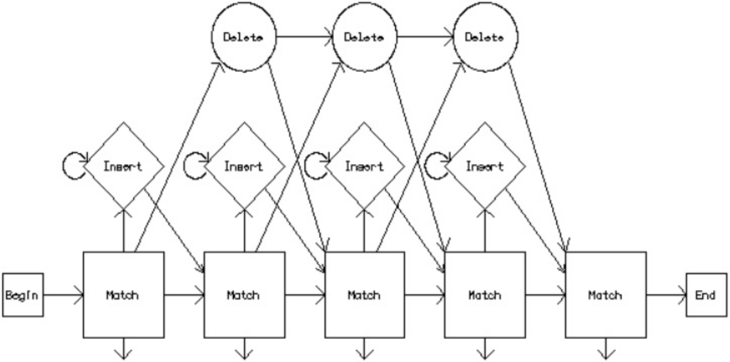
Hmmer: Profile Hidden Markov Models (HMM) for searching

This simple profile HMM is also sometimes depicted as a sequence with the size of the residue proportional to its probability.



Profile Hidden Markov Models (HMM)

More complex profile HMMs allow for insertions or deletions of amino acids or nucleic acids.



Steps for searching genomes for genes:

1. Align reference sequences of genes of interest

- using muscle

2. Build a profile HMM using the aligned reference sequences

- using hmmbuild

3. Search the bacterial genome for the genes of interest

- using hmmsearch

Let's install!

To distribute the computational burden, we will have you use one of CRC front-end machines depending on the first letter of your last name:

[A-K] - use `@crcfe01.crc.nd.edu`

[L-Z] - use `@crcfe02.crc.nd.edu`

Installing tools

This week one tool we will use is an aligner called `Muscle`. This tool is available for free on the creator's website:

`https://www.drive5.com/muscle/downloads.htm`

We'll also be using a pattern matching tool called `hmmer`. This tool is also available for free on the web:

`http://hmmer.org/download.html`

Installing tools - using `wget`

We do not have access to a graphical user interface (GUI) on the remote machine so we must download our install files using `wget`.

The easiest way to do this is go to the pages at the URLs provided for each tool. Find the version we want and copy the download link.

Then in our remote computer terminal we can execute the command:

```
wget copied_link
```

, where we paste the link we copied in place of copied_link

Installing tools - tarballs

Often the tools come in a compressed format to make it easier to download. In Unix this compressed format is called a “tarball” because the function used to generate the compressed file is `tar`. These “tarballs” are often also “zipped” to further compress them. As a result many tools you download will have the file names like `toolname.tar.gz`.

We have to “unpack” these compressed files to use the tools. This is accomplished using two tools (`gzip` and `tar`).

`gzip -d toolname.tar.gz`, which will “unzip” our file leaving us with a new file `toolname.tar`.

`tar -xf toolname.tar`, which will “untar” our file leaving us with our tool or a directory containing components of our tool.

Installing tools: source code vs. binaries

- **Source code** - this is often written in a compiled language and must be *compiled* on your machine. Many times these tools will have *dependencies* that may or may not be included with the source code. A common tool for distributing and compiling biocomputational tools is called `make`, usually three steps are required - `./configure`, `make`, `make install`.

- **Binaries** - these are previously compiled versions of the tool. These are operating system specific and sometimes may not work for your system because they were compiled ona different system. However, when these are available and work for you, it allows you to avoid the sometimes difficult compiling process.

Installing tools: specifics

- Per convention, let's install our tools in `~/Private/bin`, where ~ indicates your home directory
- Muscle: this one is a binary and that makes it easy!
- Hmmer: we'll need to compile this one...
    - one other potential issue with working on a remote computer is permissions
    - on your own computer you have the power to do most anything; this isn't the case on remote computers because they are shared by many users and a system administrator is "in charge"
    - to compile Hmmer, we'll use the following commands:

```
./configure --prefix /local/directory/path/

make

make install
```

How do we figure out how to use tools?