UNIVERSITÄT
DES
SAARLANDES

Efficient Python Implementation of Pattern Matching

Algorithms for Sequence Analysis

Sven Rahmann

Summer 2021

Overview

Previous Lectures

Bit-parallel algorithms:

- Basic **Shift-And** algorithm
- **BNDM algorithm** (backward non-deterministic DAWG matching)
- Bit-parallel algorithms for general patterns based on Shift-And

Overview

Previous Lectures

Bit-parallel algorithms:

- Basic **Shift-And** algorithm
- **BNDM algorithm** (backward non-deterministic DAWG matching)
- Bit-parallel algorithms for general patterns based on Shift-And

Today

Efficient Implementation of Bit-Parallel Algorithms in Python

- Using the `conda` package manager
- Writing applications with CLIs
- Just-in-time compiling Python
- Searching for a bipartite DNA motif

Conda

Idea and Setup

- Language-independent package manager, often used for Python
- from Anaconda, Inc.
- Separate environments for separate projects
- Get `miniconda` for your OS (64-bit version):
  `https://docs.conda.io/en/latest/miniconda.html`
- Download, install, make sure to add `conda` to your PATH.

Conda

Idea and Setup

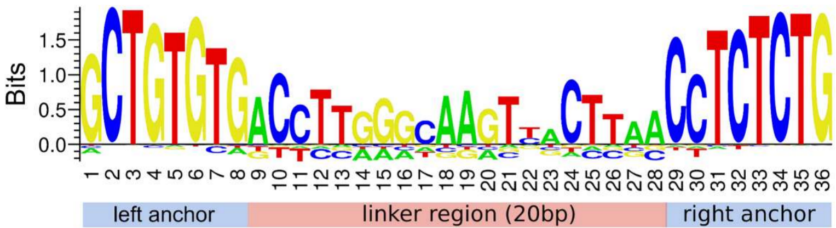
- Language-independent package manager, often used for Python
- from Anaconda, Inc.
- Separate environments for separate projects
- Get `miniconda` for your OS (64-bit version):
  `https://docs.conda.io/en/latest/miniconda.html`
- Download, install, make sure to add `conda` to your PATH.

Create and activate environment

- a) `conda create -n alsa -c conda-forge  python=3.9 numpy numba`
- b) `conda create -n alsa -c conda-forge  -f requirements.txt`
- c) `conda env create` (using `environment.yml` file)
- `conda activate alsa`

Writing DNA Motif Search Application in Python

- Search a genome (given as FASTA file)
  for a DNA motif (given as IUPAC string with variable-length gaps)
- Motif example: zinc finger protein ZNF768 binding site
  `RCTGTGYRN(17,23)CYTCTCTG`



Source: Rohrmoser et al. (2019). *Nucleic Acids Research* 47(2): MIR sequences recruit zinc finger protein ZNF768 to express genes.

Writing DNA Motif Search Application in Python

```
$ python motifmatcher.py --help
usage: motifmatcher.py [-h] [--motif MOTIF] --fasta FASTA

DNA Motif Searcher

optional arguments:
  -h, --help            show this help message and exit
  --motif MOTIF, -m MOTIF
                        DNA motif (IUPAC) with optional N(min,max) elements
  --fasta FASTA, -f FASTA
                        FASTA file of genome
```

Python

Features

- decorators like `@njit`
- bytes `b'hello'` vs. strings `'hello'`
- context managers: `with open(...) as name:`
- generator expressions, generator functions

Python

Features

- decorators like `@njit`
- bytes `b'hello'` vs. strings `'hello'`
- context managers: `with open(...) as name:`
- generator expressions, generator functions

Modules

- `collections` for non-standarad containers
- `argparse` for CLIs
- `re` for parsing regular expressions
- `numpy` for typed arrays
- `numba` for just-in-time compilation