Metapasta: scalable tool for microbial community profiling

Evdokim Kovach, Alexey Alekhin, Marina Manrique, Pablo Pareja-Tobes, Eduardo Pareja, Raquel Tobes and Eduardo Pareja-Tobes

Era7 bioinformatics, Granada, Spain

What is Metapasta?

Metapasta is a tool for microbial community profiling.

It's designed to answer questions like:

- Which species are presented in the microbial sample?
- How many different species are presented in the sample?
- How many species from the given genus are presented in the sample?

Metapasta pipeline

1. Merging paired-end reads by FLASh.
2. Mapping reads mapped against the 16S database by BLAST (or LAST).
3. Assigning each e read to a taxon or signing it as *unassigned*.

Assignment paradigm II. Best BLAST Hit

read id	taxon	score
$read_1$	43769	0.1
	1079988	0.2
$read_2$	38290	0.1
	1716	0.82
$read_3$	698966	0.3
	698973	0.4
	931089	0.35

$\rightsquigarrow$

read id	taxon
$read_1$	1079988
$read_2$	1716
$read_3$	698973

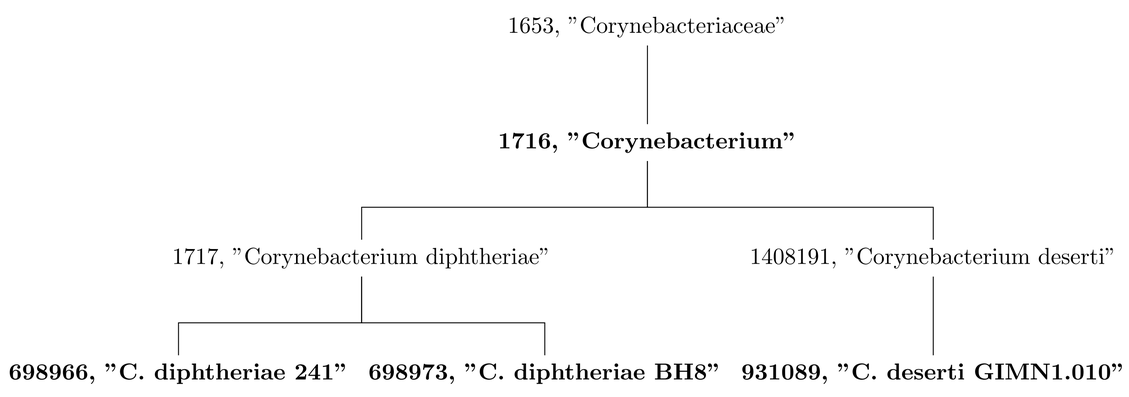
Assignment paradigm II. The Lowest Common Ancestor

read id	taxon	score
$read_1$	43769	0.1
	1079988	0.2
$read_2$	38290	0.1
	1716	0.82
$read_3$	698966	0.3
	698973	0.4
	931089	0.35

$\rightsquigarrow$

read id	taxon
$read_1$	$lca(43769, 1079988)$
$read_2$	$lca(38290, 1716)$
$read_3$	$lca(698966, 698973, \ldots)$

Taxonomy tree



Why cloud computing?

Mapping NGS reads against the 16S database is quite computationally expensive task.

For example, even on a fast computer with a SSD and a big size of RAM mapping of one read against the database with BLAST takes more than 0.25 seconds.

1 000 000 reads $\times$ 0.25 seconds each $\approx$ **70 hours**.

Architecture

- **EC2 Instance** – computational node, virtual machine that can be configured to perform arbitrary computation.
- **Auto scaling group** – group of EC2 instances configured in the same way.
- **S3 bucket** – cloud storage.
- **SQS queue** – temporary buffer to share the data between instances.

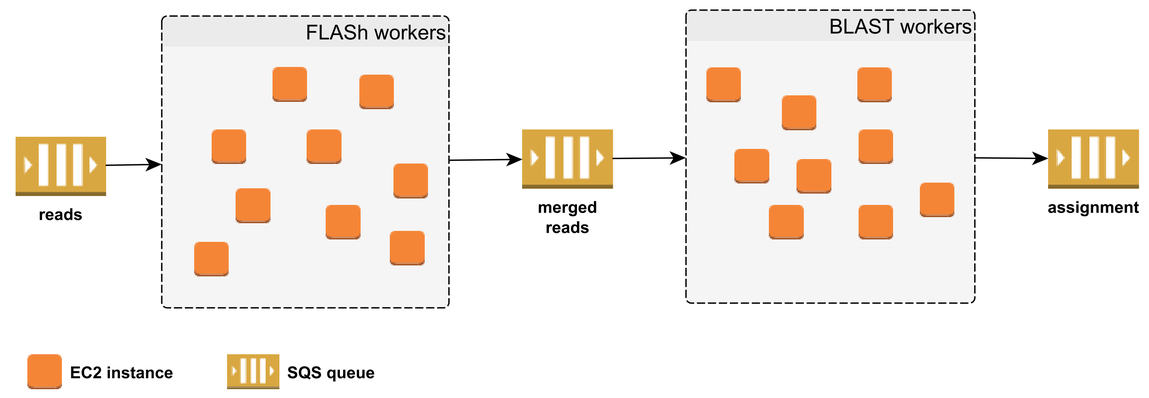


Figure 1: Workers and SQS queues for Metapasta pipeline

- Computation of pipeline components are preformed by instances from independent auto scaling groups
- every component has has input and output SQS queues
- output queue of one component can be used as input queue for another one
- for big messages S3 objects are used.

Web interface

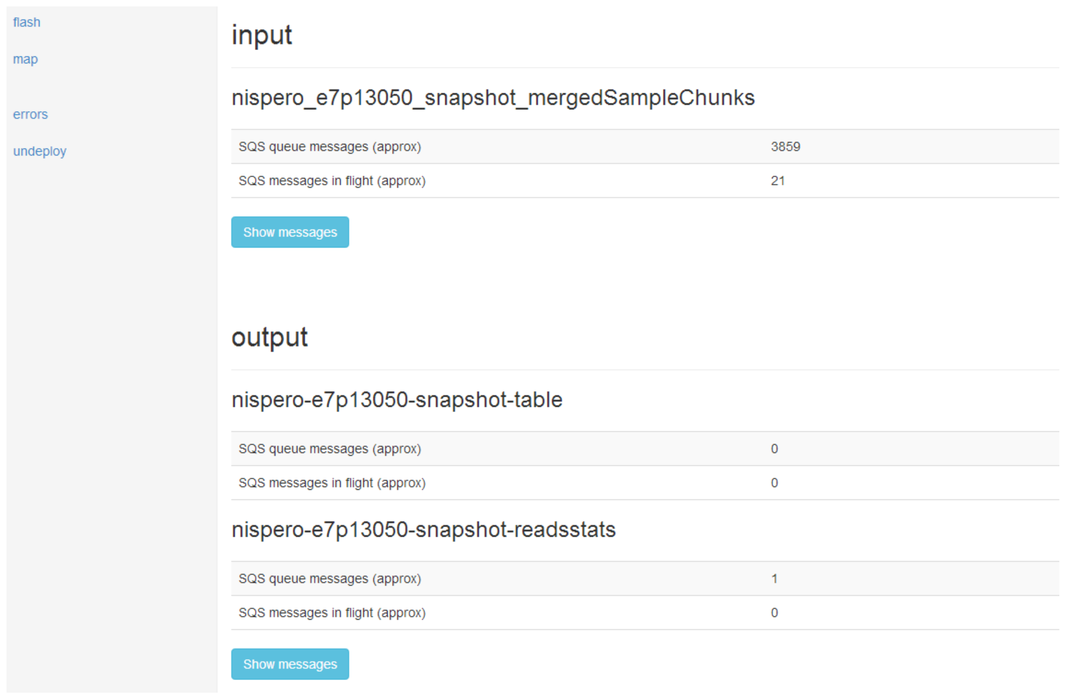


Figure 2: Web interface screenshot

INTERCROSSING

This project is funded in part by the ITN FP7 project **INTERCROSSING (Grant 289974)**.

Availability

Compota is an open-source project released under AGPLv3 license.
The source code is available at `github.com/ohnosequences/metapasta`.