PASTASpark

PASTASpark is a tool that uses the Big Data engine Apache Spark to boost the performance of the alignment phase of PASTA (Practical Alignments using SATé and TrAnsitivity). PASTASpark reduces noticeably the execution time of PASTA, running the most costly part of the original code as a distributed Spark application. In this way, PASTASpark guarantees scalability and fault tolerance, and allows to obtain MSAs from very large datasets in reasonable time.

If you use **PASTASPark**, please, cite this article:

> José M. Abuin, Tomás F. Pena and Juan C. Pichel. "PASTASpark: multiple sequence alignment meets Big Data". *Bioinformatics*, Vol. 33, Issue 18, pages 2948-2950, 2017.

PASTASpark was originally a fork from PASTA (Forked in November 2016) here and here. Later, it became a project itself in this repository. The original PASTA paper can be found with this references:

> Mirarab, S., Nguyen, N., and Warnow, T. (2014). "PASTA: Ultra-Large Multiple Sequence Alignment". In R. Sharan (Ed.), *Research in Computational Molecular Biology*, (pp. 177–191).
>
> Mirarab, S., Nguyen, N. Guo, S., Wang, L., Kim, J. and Warnow, T."PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences". *Journal of Computational Biology*, (2014).

Installation

PASTASpark only works on Linux systems.

Compilation from sources

You need Python 2.7 and git installed.

1. Clone the repository:

```
git clone https://github.com/citiususc/pastaspark.git
```

2. Enter the created directory and run the install command:

```
cd pastaspark
python setup.py develop --user
```

Running PASTASpark

Running Dependencies

1. Python 2.7 or later.
2. Java 8 (required for Opal, which is by the default used in PASTA for merging small alignments).
3. A cluster with Hadoop/YARN and Spark installed and running. Tested with Hadoop 2.7.1 and Spark 1.6.1.
4. A shared folder among the computing nodes to store the results in the cluster.

Working modes

PASTASpark can be executed as the original PASTA or on a YARN/Spark cluster. In this way, if you launch **PASTASpark** within a Spark context, it will be executed on your Spark cluster. You can find more information about this topic in the next section.

Examples

A basic example of how to execute **PASTASpark** in your local machine with a working Spark setup is:

```
spark-submit --master local run_pasta.py -i data/small.fasta -t data/small.tree
```

The following is an example of how to launch PASTASpark using a bash script and taking as input the files stored in the data directory:

```
#!/bin/bash

SPARK_COMMAND="spark-submit --master yarn --deploy-mode cluster"
DRIVER_MEM="25G"
EXEC_MEM="5G"

CURRENT_DIR=`pwd`
HOME="/home/jmabuin"

NUM_EXECUTORS="8"
DRIVER_CORES="4"
EXECUTOR_CORES="1"
ARCHIVES="pasta.zip"
PY_FILES="pasta.zip,$HOME/.local/lib/python2.7/site-packages/DendroPy-3.12.3-py2.7.egg"

INPUT_DATA="$CURRENT_DIR/data/small.fasta"
INPUT_TREE="$CURRENT_DIR/data/small.tree"

$SPARK_COMMAND --name PastaSpark_Small_8Exec --driver-memory $DRIVER_MEM --executor-memory $EXEC_MEM --num-executors $NUM_EXECUTORS --driver-cores $DRIVER_CORES --executor-cores $EXECUTOR_CORES --archives $ARCHIVES --py-files $PY_FILES run_pasta.py --temporaries=./ -i $INPUT_DATA -t $INPUT_TREE --num-cpus=$DRIVER_CORES --num-cpus-spark=$EXECUTOR_CORES --num-partitions=$NUM_EXECUTORS
```

To see the original PASTA documentation, click here.

INFORMACIÓN

Investigadores
Tomás Fernández Pena
Juan Carlos Pichel Campos
José Manuel Abuín Mosquera

Licenza

DESCARGAR

- Repositorio Gitlab
- Descargar de Gitlab
- Repositorio Github

PUBLICACIÓNS

PASTASpark: multiple sequence alignment meets Big Data
Bioinformatics, 2017

PROXECTOS DE INVESTIGACIÓN

SDNHPC: Solucións para novos desafíos en computación de altas prestacións