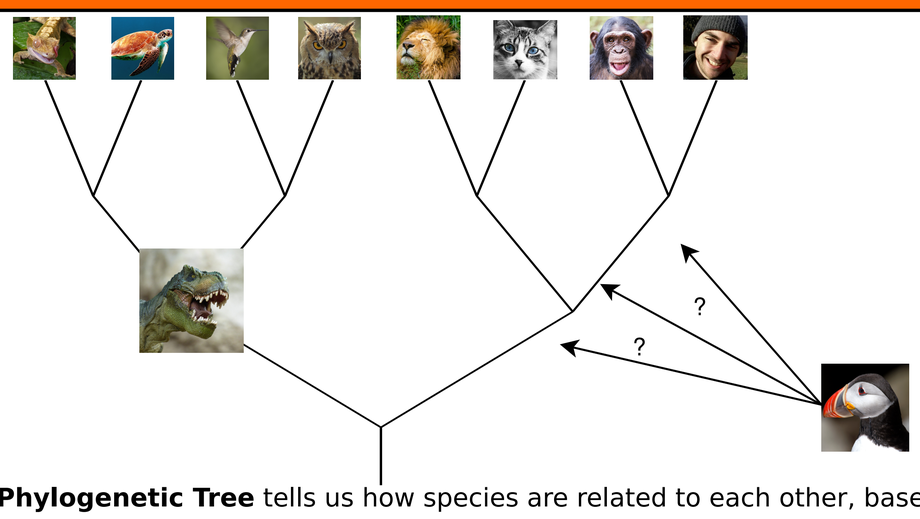
Massivley Parallel Evolutionary Placement of Genetic Sequences

Pierre Barbera[1], Alexey Kozlov[1], Tomas Flouri[1], Diego Darriba[1], Lucas Czech[1] and Alexandros Stamatakis[1,2]

[1] Scientific Computing Group, Heidelberg Institute for Theoretical Studies, Germany
[2] Institute for Theoretical Informatics, Karlsruhe Institute of Technology, Germany
Email: {pierre.barbera, alexandros.stamatakis}@h-its.org

Background and Motivation



A **Phylogenetic Tree** tells us how species are related to each other, based on their DNA sequences

Metabarcoding

Using new sequencing technologies, we can obtain sequences from samples without knowing what species we sequenced from (e.g. when sequencing any/all DNA content of a sample)

Applications so far:
- Profiling of environmental bacterial communities
- Correlations between bacterial profiles of human microbiomes and human health
- Detection of possible novel species

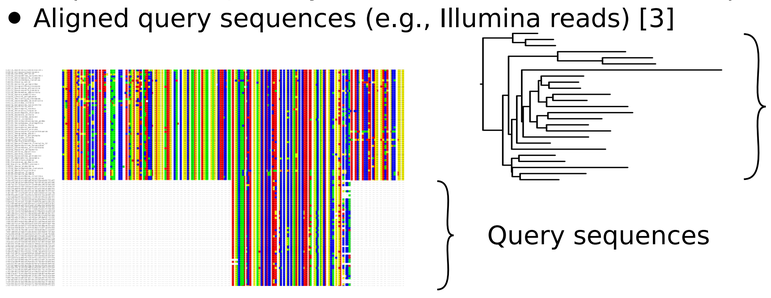
Phylogenetic Placement is the most reliable way of determining what we sequenced

Phylogenetic Placement

Input Data

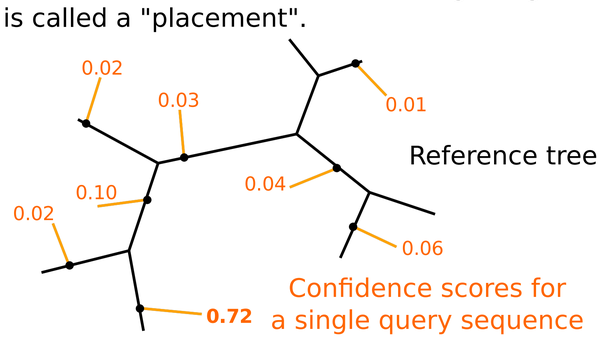
Phylogenetic placement [1, 2] takes as input:

- A multiple sequence alignment of reference sequences (e.g., the 16S or a similar barcoding gene)
- A species tree (usually inferred from the reference sequences)
- Aligned query sequences (e.g., Illumina reads) [3]



Placement Algorithm

The algorithm calculates the most likely (via maximum likelihood) insertion position for every query sequence on the reference tree. The resulting assignment of a query sequence to a branch is called a "placement".



With some precomputations for the reference tree, the likelihood score for **each branch** and sequence combination can be **computed independently.**

This means, the basic placement procedure is almost embarrasingly parallel.

However, **advanced heuristic and statistical techniques** do **require reduce-type operations** for every query sequence, such that confidence scores can be calculated.

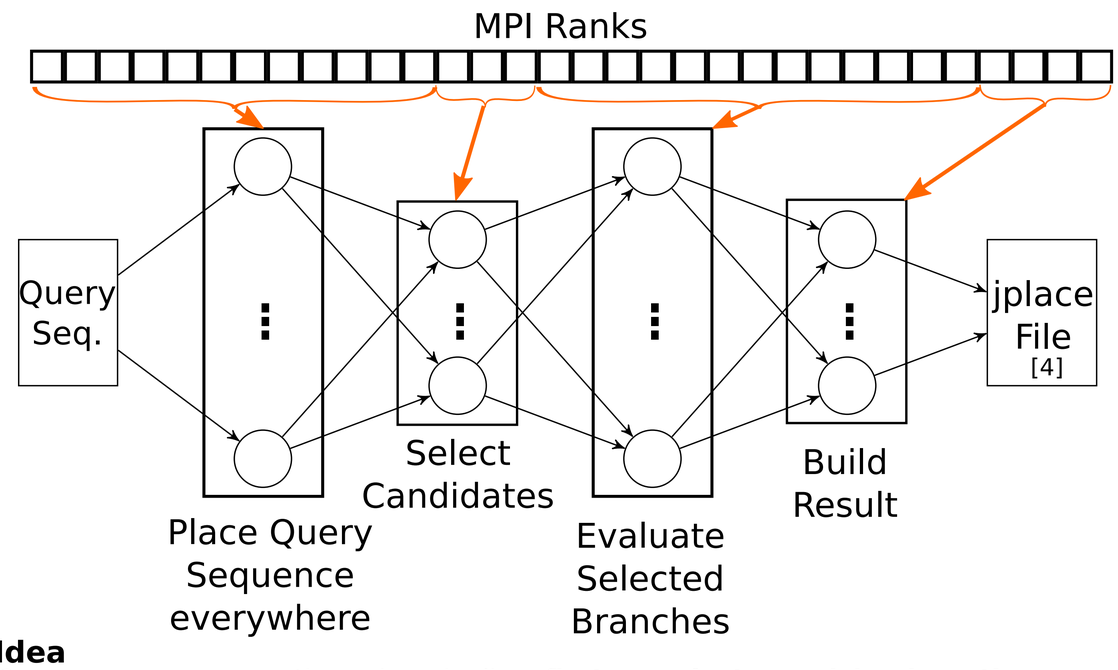
Thus, a map-reduce style approach might work in principle, however CPU utilization would be low during the reduction phase. Communication bottlenecks would also be an issue. Further, multi-stage versions of the algorithm can have imbalanced workload between stages, possibly leaving many cores underutilized.

A further challenge is the **memory footprint**, which can become prohibitively **large with large reference trees** (>10k species).

References

[1] Matsen, Frederick A., Robin B. Kodner, and E. Virginia Armbrust. "pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree." BMC Bioinformatics 11.1 (2010): 538.
[2] S. Berger, D. Krompass, and A. Stamatakis, “Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood,” Syst. Biol., vol. 60, no. 3, pp. 291–302, 2011.
[3] S. Berger and A. Stamatakis, “Aligning short reads to reference alignments and trees,” Bioinformatics, vol. 27, no. 15, pp. 2068–2075, 2011.
[4] F. A. Matsen, N. G. Hoffman, A. Gallagher, and A. Stamatakis, “A format for phylogenetic placements,” PLoS One, vol. 7, no. 2, pp. 1–4, Jan. 2012.
[5] Mahé, F., de Vargas, C., Bass, D., Czech, L., Stamatakis, A., Lara, E., Singer, D., Mayor, J., Bunge, J., Sernaker, S. and Siemensmeyer, T., 2017. Parasites dominate hyperdiverse soil protist communities in Neotropical rainforests. Nature Ecology & Evolution, 1, p.0091.

Parallel Streaming Pipeline



Idea

Query sequences pass through a pipeline, **first** we **select promising insertion branches** in the tree, **then** we **evaluate placements** on this subset reduced number of branches **more thoroughly**

If properly balanced, **all ranks always have work**, and the **memory footprint** is **distributed** across many cores.
Sources of imbalance are not easily predictable, they **depend on**:
- how many branches will qualify as "candidates" for thorough investigation
- numerical optimization (e.g. Newton-Raphson, BFGS) convergence speed
- proportion of candidate branches that are at the leafs of the tree

Advantages

- Sequences in chunks -> limited memory consumption
- Every core is always working (as opposed to classical map-reduce)
- Communication can be overlapped with computation
- Memory footprint can be spread across many cores

Challenges

- Pipeline needs to be balanced
- Code complexity relatively high

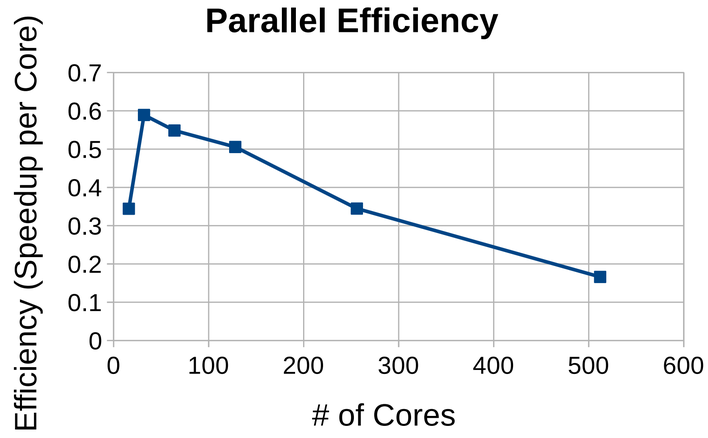
Preliminary Results

- **~4-fold sequential speedup** compared to to previous implementation
- comparable shared-memory scalability
- cluster version **can handle large trees** (tested with 10k species) efficiently
- dynamic pipeline **balancing improves parallel efficiency significantly** compared to a static schedule

Real-world data test

- 1 million sequences
- Tropical Soil Samples [5]
- Tree with 512 species

Supercomputer:
- 16 cores per node
- Intel Haswell, AVX
- Infiniband Interconnect
- BeeGFS File System



Future Work

While the current **pipeline balancing** algorithm works well, there is room for improvement.

Complete Pipeline: Placement requires aligned sequences, as opposed to competitors like BLAST (sequence similarity search). **Alignment is the bottleneck** (~100 times slower than Placement)

Placement can have good **synergy with Phylogeny-Aware Alignment** [3] methods that operate on Reference trees and alignments

Availability and Acknowledgements

PEPA (my implementation of [2]) will (soon) be available at https://github.com/Pbdas/epa

This work was financially supported by the Klaus Tschira Foundation.

The original methods and implementations were developed by S.A. Berger, D. Krompaß, and A. Stamatakis [2]

www.exelixis-lab.org

Heidelberg Institute for Theoretical Studies

HITS

Karlsruhe Institute of Technology