Tax Analysis

“Compact format” algorithm description

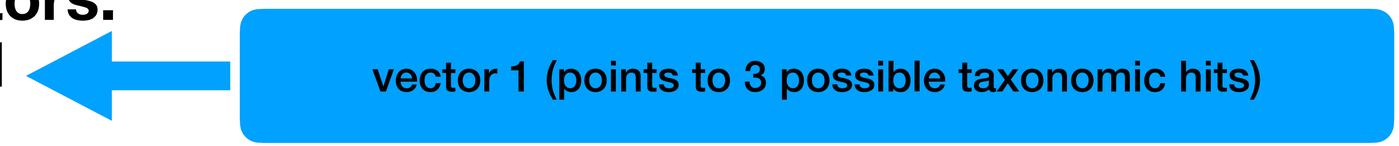
Shkeda, Shutov, Kuznetsov, Dec.2021

Intent

- Explain the STAT output format vs “compact format”
- Describe our approach as “set of sets” problem
- Use cardinality as a key set characteristic for building parallel processing

Generic (non-compact format) STAT output format

- non-compact format produces output per read/spot
- can be voluminous
- final stage formatting benefits from an additional processing (known as “compact format”, but it is an additional processing)
- Compact format is a counted set of sets of possible variants of taxonomic identification vectors
  Examples of vectors:
  1875, 9606, 10001
  1875, 9606
  1878, 9000, 10001



Compact format processing (1)

Find all unique sets and count them

- 1875, 9606, 10001
  1875, 9606
  9606
  1878, 9000, 10001
  9606
  1875, 9606

find unique variants
of tax vectors
and count them

- 1875, 9606, 10001 - 1
  1878, 9000, 10001 - 1
  9606 - 2
  1875, 9606 - 2

{ 1875, 9606 } -
we see it 2 times in the set

Compact format processing (1)

Find all unique sets and count them

- 1875, 9606, 10001
  1875, 9606
  9606
  1878, 9000, 10001
  9606
  1875, 9606

Each line in the matrix is a set of unique ids (no duplicates, sorted)

Input matrix of tax.hits (column-wise)

Each tax-vector (row in the matrix) has its Cardinality (Number of elements or Population Count)

Possible cardinalities are:
3, 2, and 1
(we don't have empty sets)

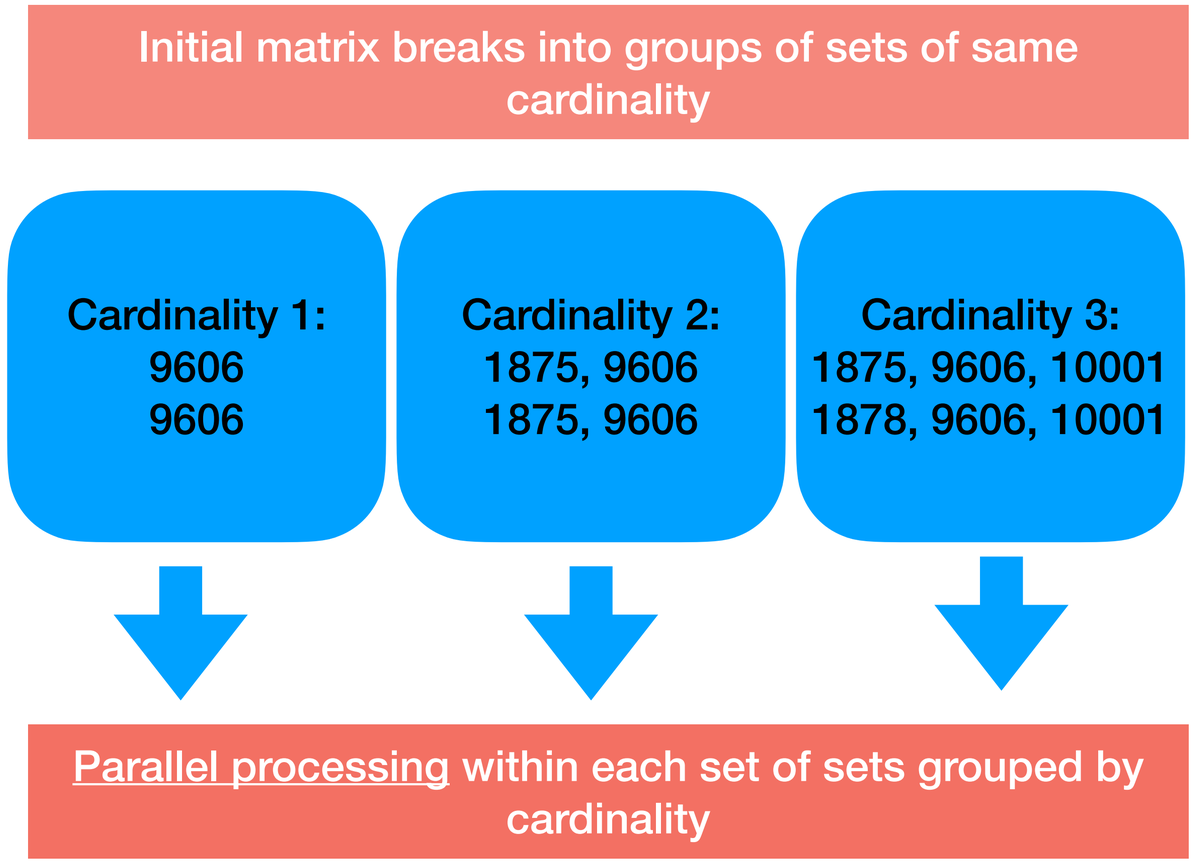
Cardinality 1:
9606
9606

Cardinality 2:
1875, 9606
1875, 9606

Cardinality 3:
1875, 9606, 10001
1878, 9606, 10001

Compact format processing (2)

Find all unique sets and count them - for each cardinality group



We do it because
two sets with different cardinality
cannot be the same set

Each set is separate stream of processing
(expect 8-15 groups)
Embarrassingly parallel, problems are completely independent:
(sort-unique-count algorithm)