Lightning - Bug #12951

FastJ aggregation into SGLF takes too much memory

01/12/2018 05:46 PM - Abram Connelly

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assigned To:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Tiling Future		

Description

the fastj2cgflib converts FastJ to SGLF by loading all the FastJ tiles into memory. If we assume the FastJ are sorted, or if we load the FastJ tiles and sort them ourselves, we can drastically reduce the memory footprint of the resulting SGLF creation.

Since each tile step only requires knowledge of other tile in the same tile step, we can free memory of data after we've processed the tile step.

Currently, even for moderate datasets, memory usage can balloon to upwards of 100G+.

History

#1 - 03/15/2018 03:42 PM - Abram Connelly

- Status changed from New to In Progress

A partial update: The new fjcsv2sglf code that converts from FastJ to SGLF still loads the whole SGLF into memory but:

- fjt that is used to provide FastJ in CSV format has been altered to allow for unordered output of FastJ which fjcsv2sglf can then take and process.
- uses a 2bit representation of genomic data in memory to cut down on space (which should be roughly 1/4 of the memory footprint)

This isn't a complete solution as there are still major savings to be had either by a differential or compressed storage in memory or by creating some type of "server" that has a cache of common tiles but otherwise stores the tile library on disk. The fjcsv2sglf is a start and serves the immediate purpose of being able to process FastJ for a reasonably sized population (2k+).

This ticket might need to be closed out in favor of another but we need to decide how to best provide the FastJ to SGLF (or other library format) conversion. I think the best option is:

- Provide a tile library server that keeps a cache of common tiles (stored in 2bit format say) in memory and keeps the rest on disk.

Some issues to consider:

- Are the list of common tiles static or dynamic?
- Should we use some "off the shelf" in memory database to store the common tile sequences?
- How many common tiles do we keep in memory?

For the last option, we can make an informed decision based on the frequency of tile and do some timing analysis of how long it takes to convert.

#2 - 04/30/2019 05:37 PM - Jiayong Li

- Target version set to Tiling Future

Current cwl implementation uses the updated fjcsv2sglf.

#3 - 04/30/2019 05:38 PM - Jiayong Li

- Status changed from In Progress to New

- Assigned To deleted (Abram Connelly)