package com.twitter.simclusters\_v2.scalding.common.matrix

import com.twitter.algebird.{Aggregator, Semigroup}

import com.twitter.bijection.Injection

import com.twitter.scalding.{TypedPipe, ValuePipe}

/\*\*

\* A matrix trait for representing a matrix backed by TypedPipe

\*

\* @tparam R Type for rows

\* @tparam C Type for columns

\* @tparam V Type for elements of the matrix

\*/

abstract class TypedPipeMatrix[R, C, @specialized(Double, Int, Float, Long, Short) V] {

implicit val semigroupV: Semigroup[V]

implicit val numericV: Numeric[V]

implicit val rowOrd: Ordering[R]

implicit val colOrd: Ordering[C]

implicit val rowInj: Injection[R, Array[Byte]]

implicit val colInj: Injection[C, Array[Byte]]

// num of non-zero elements in the matrix

val nnz: ValuePipe[Long]

// list of unique rowIds in the matrix

val uniqueRowIds: TypedPipe[R]

// list of unique unique in the matrix

val uniqueColIds: TypedPipe[C]

// get a specific row of the matrix

def getRow(rowId: R): TypedPipe[(C, V)]

// get a specific column of the matrix

def getCol(colId: C): TypedPipe[(R, V)]

// get the value of an element

def get(rowId: R, colId: C): ValuePipe[V]

// number of unique rowIds

lazy val numUniqueRows: ValuePipe[Long] = {

this.uniqueRowIds.aggregate(Aggregator.size)

}

// number of unique unique

lazy val numUniqueCols: ValuePipe[Long] = {

this.uniqueColIds.aggregate(Aggregator.size)

}

}