KINC v0.1 Specification

Knowledge Independent Network Construction

Joshua Burns*, Stephen Ficklin*

November 18, 2015

*Dept. of Horticulture, Washington State University

Contents

1 Introduction **3**

2 Abstract Classes **4**
- 2.1 KINCData . . . 4
- 2.2 KINCAnalytic . . . 4

3 KINCData Classes **5**
- 3.1 Expression . . . 5
  - 3.1.1 Properties . . . 5
  - 3.1.2 Constructor . . . 5
  - 3.1.3 Virtual Functions . . . 5
  - 3.1.4 File Structure . . . 5
- 3.2 Correlation . . . 5
  - 3.2.1 EXP Binary File Format . . . 6
- 3.3 Network . . . 6
  - 3.3.1 File Structure . . . 6
- 3.4 Annotation . . . 6
  - 3.4.1 File Structure . . . 6

4 KINCAnalytic Classes **8**
- 4.1 Pearson . . . 8
- 4.2 Spearman . . . 8

1 Introduction

KINC is designed for use in construction of biological networks, specifically, gene co-expression networks. KINC performs three major steps: 1) construction of a similarity matrix of pair-wise expression correlations, 2) thresholding of the similarity matrix to form an adjaceny matrix, 3) export of the adjaceny matrix to form a tab-delimited network file.

This document provides an overview for the data structures and file formats used by KINC.

Can we describe here how the command-line for KINC will work and how use of the classes described below translates into command-line arguments?

2 Abstract Classes

The following abstract classes serve as the base for all structures within the kinc program. All other classes should inherit from these. This design will allow for dynamic addition of user implemented plugins in the future, because the functions of the abstract classes are clearly defined.

2.1 KINCData

Th KINCData class is responsible for reading and writing of KINC data files. Classes that inherit from the KINCData class are responsible for implementing the generic functions exposed by this class. This includes reading, writing, merging, exporing, indexing and querying. Typically, child classes provide importers that read commonly used file formats into their own binary file format and exporters to convert back to those same file formats.

2.2 KINCAnalytic

The KINCAnalytic class is responsible for taking in one or more KINCData objects and employ an algorithm such as a statistical test to produce one or more new KINCData objects. A KINCAnalytic

3 KINCData Classes

3.1 Expression

The Expression class is responsible for manging gene expression-level data.

3.1.1 Properties

Do we need any properties?

3.1.2 Constructor

KINCData(int argc, char *argv[])

We need to design how the functions of the class will receive arguments. will we have a constructor that receives, parses and responds to errors for all functions? Or should each function be responsible for checking it's own arguments. I know we can't do that in the abstract class, but we need to accomdate the behavior we settle on in our design so plugins are consistent.

3.1.3 Virtual Functions

The following functions should be implemented by any plugin that creates classes that inherits the KINCData class.

virtual void import() = 0

This function reads a tab-delimited file. Each line of this file represents the gene expression levels of a single gene, transcript or probeset. Each tab-separated value in a single line indicates the gene expression level for each sample. The expression level of a samples must be in the same order for every line. The first line of the file may contain a tab-delimited list of sample names, and a file may contain as many samples and genes as desired.

virtual void export() = 0

virtual void query() = 0

virtual void merge() = 0

3.1.4 File Structure

1. Special Identifier that identifies this file as an expression file.
2. User defined name for expression data.
3. History of this data, where it came from.
4. Number of genes, then number of samples per gene.
5. List of all genes.
6. 2 Dimensional list of all gene samples, per gene.

3.2 Correlation

This is responsible for storing correlation data between genes.

3.2.1 Correlation Binary File Format

The following describes the format of the KINC correlation file. All multi-byte numbers are little-endian, regardless of the machine endianness.

Field	Description	Type	Value
magic	The magic number that identifies this file as a correlation file.	char[5]	kcor\1
historyLen	The length of the header.	int32_t	
nameLen	The length of the dataset name.	int32_t	
name	The user defined name for this correlation dataset.	char[nameLen]	
historyLen	The length of history information.	int32_t	
history	A string that describes the provenance of this file.	char[historyLen]	

I like this type of table for describing the file format. I borrowed it from the BAM file specification

1. Special Identifier that identifies this file as a correlation file.
2. User defined name for correlation data.
3. History of this data, where it came from.
4. Number of genes, number of correlations per gene pair.
5. List of correlation methods used.
5. List of all genes.
6. Special lists of all gene correlations sorted by their correlation value. Each entry will have a correlation value, then a number representing how many gene correlations are this value, then the list of gene correlations. For each type of correlation used a list is made.
7. Special 2 Dimensional list of all gene correlations, special being a diagonal matrix instead of square.

3.3 Network

This is responsible for storing network data between genes.

3.3.1 File Structure

1. Special Identifier that identifies this file as a network file.
2. User defined name for network data.
3. History of this data, where it came from.
4. Number of genes.
5. List of all genes.
6. Special 2 Dimensional list of all gene edges, special being a diagonal matrix instead of square.

3.4 Annotation

This is responsible for storing additional information for genes.

3.4.1 File Structure

1. Special Identifier that identifies this file as an annotation file.
2. User defined name for annotation data.
4. Number of genes, number of annotations.
5. List of all genes.
6. List of all annotation types.
7. 2 Dimensional list of all gene annotations, per gene.

4 KINCAnalytic Classes

4.1 Pearson

This takes an Expression BioData object and produces a Correlation BioData object. It uses the Pearson correlation statistical method for giving correlation values.

4.2 Spearman

This takes an Expression BioData object and produces a Correlation BioData object. It uses the Spearman correlation statistical method for giving correlation values.