BioLockJ User Manual

College of Computing and Informatics
Department of Bioinformatics
Fodor Lab

Change Log

#	Date	Description
1.0	07/27/2017	Initial document BioLockJ v1.0

File Name

BioLockJ_User_Manual.pdf

Table of Contents

1. Pipeline Overview

BioLockJ is a light-weight, extensible, metagenomics pipeline designed to improve the speed, accuracy, and reproducibility of 16s amplicon and whole genome sequencing (WGS) data analysis. BioLockJ runs on any Linux system (and by extension OSX) but is most powerful in a high performance computing environment by utilizing its parallel processing capabilities.

Pipeline execution is guided by a single BioLockJ configuration file which can be used to reproduce your analysis and serves to document all runtime parameters.

Note: BioLockJ properties listed in Section 2.4 will appear in italics throughout this text.

Primary Inputs

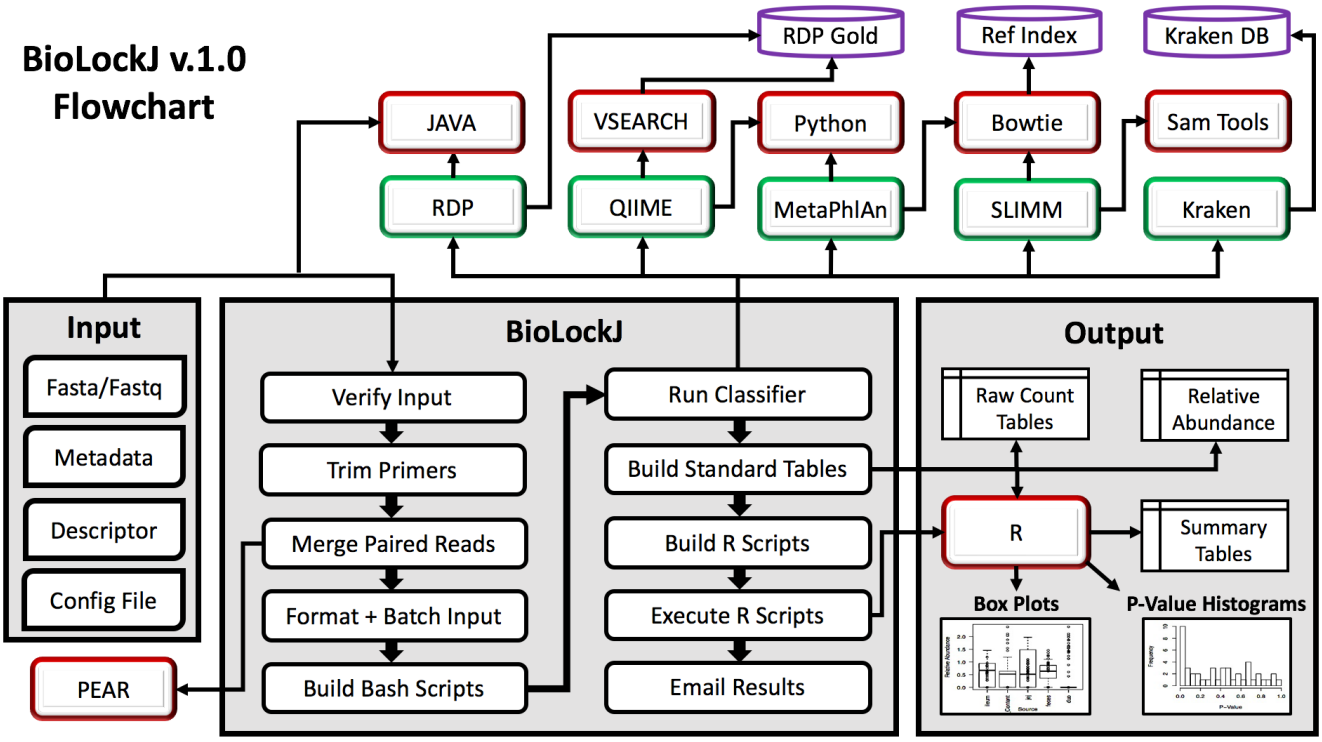
1. *input.dirs* Fast-A/Fast-Q sequence files
2. *metadata.file* Path to metadata file
3. *metadata.descriptor* Path to descriptor file

BioLockJ writes and executes bash shell scripts that call command line bioinformatics tools based on user input, provided via the BioLockJ configuration file. Generated scripts are reliable, organized, efficient, and reproducible.

Execution Summary

1. Formats sequences to meet classifier specifications
2. Trim primers, merge paired reads, and rarefy as needed
3. Classifies sequences using RDP, QIIME, Kraken, MetaPhlAn, or SLIMM
4. Generates raw count and relative abundance tables
5. Builds statistical models to find significant OTUs correlated with metadata
6. Generates summary tables and PDF reports with histograms and bar charts
7. Emails a summary report upon completion

1.1 System Diagram



1.2 Installation

- BioLockJ is deployed as a single JAR file, which can be executed by JAVA
  1. Download GitHub BioLockJ project from github.com/mikesioda/BioLockJ
  2. Install software dependencies

1.3 Software Dependencies

- Required software: Java, R, R packages (Kendall & Coin)
- BioLockJ can be deployed with as few as one classifier, the rest are optional
- Unused classifiers and their dependencies do not need to be installed

BioLockJ Software Dependencies

#	Program	Version	Description
1	Java	1.8	Required - java.com
2	Python	2.7.12	Required by QIIME and MetaPhlAn - python.org
3	Bowtie2	2.3.2	Required by MetaPhlAn and SLIMM bowtie-bio.sourceforge.net/bowtie2/index.shtml
4	SAMtools	1.4	Required by SLIMM - htslib.org
5	Vsearch	2.4.3	Required for chimera removal in QIIME. github.com/torognes/vsearch
6	PEAR	0.9.8	Paired-End reAd merger, used by RDP & QIIME sco.h-its.org/exelixis/web/software/pear
7	RDP	2.11	16S Classifier: Ribosomal Database Project github.com/rdpstaff/classifier
8	QIIME	1.9.1	16S Classifier: Quantitative Insights Into Microbial Ecology qiime.org
9	Open MPI	1.10	Open Message Passing Interface, required by QIIME open-mpi.org
10	Kraken	0.10.5-beta	WGS Classifier - ccb.jhu.edu/software/kraken
11	MetaPhlAn2	2.0	WGS Classifier: Metagenomic Phylogenetic Analysis huttenhower.sph.harvard.edu/metaphlan2
12	SLIMM	0.2.2	WGS Classifier: Species Level Identification of Microbes from Metagenomes - github.com/seqan/slimm
13	GNU Awk	4.0.2	Convert Fastq to Fasta for QIIME - gnu.org/software/gawk
14	GNU Gzip	1.5	Decompress gzipped files for QIIME - gnu.org/software/gzip
15	R	3.2.3	Statistical modeling package - cran.r-project.org
16	Kendall	2.2	Kendall rank correlation p-values for continuous data types cran.r-project.org/web/packages/Kendall/index.html
17	Coin	1.2	Conditional Inference Procedures in a Permutation Test Framework: Computes exact p-value for Wilcox_test cran.r-project.org/web/packages/coin/index.html

1.4 Launching BioLockJ

- Runtime parameters are located in the BioLockJ configuration file
- To run BioLockJ, call java on BioLockJ.jar and pass the configuration file path

Java Command

nohup java -jar *${jar_path}*/BioLockJ.jar *${config_file_path}* > /dev/null 2>&1 &

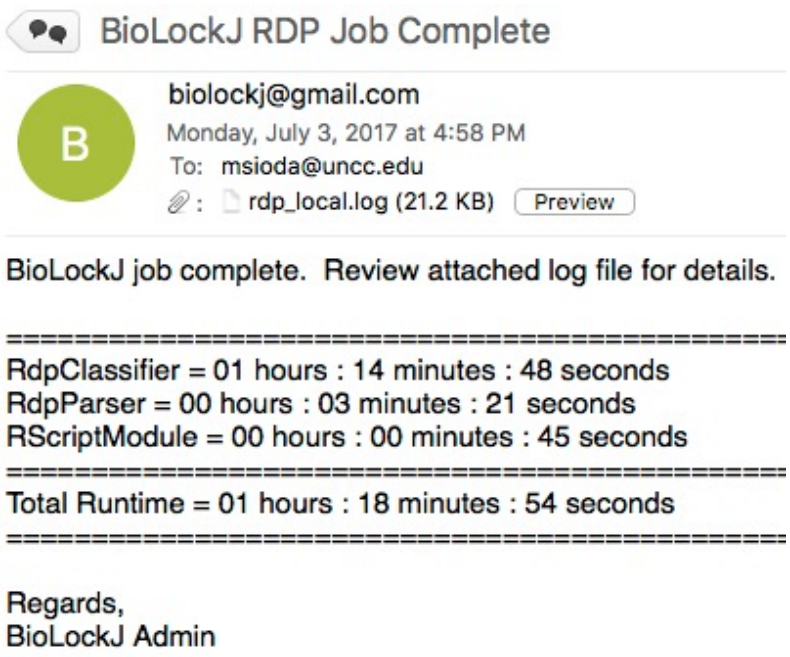
Command Part	Description
nohup	Continue execution after the terminal connection is closed
java	Java executable command
-jar	Required parameter to run a JAR
${jar_path}/BioLockJ.jar	Path to BioLockJ.jar
${config_file_path}	Path to configuration file
> /dev/null 2>&1	Discard terminal output – it is already output to ${LOG_FILE}
&	Run process in background to free terminal for other work

1.5 Failure Recovery

- Save time by recovering from failures rather than restarting the entire pipeline
- To determine root cause of failure:
  - Check log file for Java stack trace pointing to line number in Java class
  - Check failed module's "failures" directory for failure indicator files
- Fix the problem and resume pipeline by updating BioLockJ configuration file:
  - Set successfully completed module control property values = N
  - Set input.dirs = output directory of last successful module
  - Set metadata.file = path of current version if updated by pipeline
  - Set metadata.descriptor = path of current version if updated by pipeline
- Launch BioLockJ with the updated configuration

1.6 Email Notification

- BioLockJ sends *email.to* recipients a summary report & log file after pipeline execution
- Email body reports runtime for each module
- Email body reports script failures, if any
- Log4J attachment contains execution details based on log level (see Section 3.3.2)
- Option to attach Qsub output and error files if run in clustered environment


BioLockJ RDP Job Complete

B

biolockj@gmail.com
Monday, July 3, 2017 at 4:58 PM
To: msioda@uncc.edu
: rdp_local.log (21.2 KB) Preview

BioLockJ job complete. Review attached log file for details.

==========
RdpClassifier = 01 hours : 14 minutes : 48 seconds
RdpParser = 00 hours : 03 minutes : 21 seconds
RScriptModule = 00 hours : 00 minutes : 45 seconds
==========
Total Runtime = 01 hours : 18 minutes : 54 seconds
==========

Regards,
BioLockJ Admin

2. Input File Specification

2.1 Sequence Files

- Supported classifiers will accept 2 sequence file formats:
  1. Fast-A
  2. Fast-Q
- Files may be gzipped (with extension .gz)
- Files names must be unique – important to consider with multiple *input.dirs*
- Kraken & RDP sequence files may be multiplexed

2.2 Metadata

- Columns must be tab-delimited
- First column must hold the Sample ID (regardless of column header name)
- Blank cells are considered empty Strings
- Null values must be indicated using *metadata.nullValue*

2.3 Descriptor

- Columns must be tab-delimited
  - Attribute (Column 1): required field, contains name of metadata attribute
  - Type (Column 2): required field, options: Binary, Continuous, Categorical
  - Comments (Column 3): optional field, for user comments
- Metadata first column contains Sample ID, not to be included in descriptor file
- All metadata columns, other than the 1st, must be assigned an attribute type
- Type determines the statistical model implemented in R for each *report.attributes*:
  - Binary: Exact Wilcoxon Signed Rank Sum (Coin package)
  - Continuous: Kendall Tau Rank Correlation (Kendall package)
  - Categorical: One-Way ANOVA

Attribute	**Type**	**Comments**
attribute1	Binary	Metadata column must contain 1 – 2 unique values
attribute2	Continuous	Metadata column must contain numeric values
attribute3	Categorical	Metadata column must contain 2 or more unique values

2.4 BioLockJ Configuration

- The BioLockJ configuration file contains runtime parameters as name-value pairs:
- List properties are comma separated

  property1.name=property.value1
  property2.name=property.value2a,property.value2b,property.value2c
  …

Property	**Value**
project.name	Each pipeline execution generates a timestamped directory: project.name_yyyyMMdd_kkmmss
project.rootDir	Parent directory for BioLockJ project run directories
project.copyInputFiles	Options: Y/N. If Y, copy *input.dirs* into the project directory
project.deleteTempFiles	Options: Y/N. If Y, delete module temp dirs after execution
project.classifierType	Options: rdp, qiime, kraken, metaphlan, slimm
control.runOnCluster	Options: Y/N. If Y, cluster properties is required
control.trimSeqs	Options: Y/N. If Y, the SeqTrimmer module will execute
control.mergePairs	Options: Y/N. If Y, the PairedSeqMerger module will execute
control.rarefySeqs	Options: Y/N. If Y, the Rarefier module will execute
control.runClassifier	Options: Y/N. If Y, the ClassifierModule will execute
control.runParser	Options: Y/N. If Y, the ParserModule will execute
control.run_rScript	Options: Y/N. If Y, the RScriptBuilder will execute
input.dirs	Must contain files expected by the first module executed. Multiple directories must be comma-separated.
input.ignoreFiles	Files listed here will be ignored if found in *input.dirs*. Multiple files must be comma-separated.
input.demultiplex	Options: Y/N. If Y, sequence files include reads from multiple samples & sample IDs must be extracted from the sequence headers. RDP & Kraken classifiers only. If N, there is one sample per file and the file name must contain the sample ID.
input.pairedReads	Options: Y/N. If Y, file names must include *input.forwardFileSuffix* or *input.reverseFileSuffix*
input.forwardFileSuffix	File name suffix to indicate a forward read
input.reverseFileSuffix	File name suffix to indicate a reverse read
input.trimPrefix	For files named by Sample ID, provide the prefix preceding the ID to trim when extracting Sample ID. If *input.demultiplex*=Y, provide any characters in the sequence header preceding the ID. For fastq, typically “@”.

Property	**Value**
input.trimSuffix	For files named by Sample ID, provide the suffix after the ID, often this is just the file extension. Do not include read direction indicators listed in input.forwardFileSuffix/reverseFileSuffix. If *input.demultiplex*=Y, provide 1st character in the sequence header after ID; for fastq, typically ":" char
input.rarefyMinNumSeqs	Discard samples without min # of seqs
input.rarefyMaxNumSeqs	Randomly select max # of seqs for each sample
input.trimSeqPath	Path to file containing primers to trim File must contain only one sequence per line
cluster.batchCommand	The command to submit jobs on the cluster
cluster.params	Include in header of scripts submitted on cluster
cluster.validateParams	Options: Y/N. If Y, validate procs=*script.numThreads*
cluster.modules	List of modules to load before execution Adds "module load" command to bash scripts
script.exitOnError	Options: Y/N. If Y, program exits if any script failures occur, otherwise failures logged to failure directory
script.batchSize	Number of sequence files to process per script
script.chmodCommand	Command to grant script execute permissions
script.numThreads	Passed to number of threads parameter in classifier
metadata.file	Metadata file path, attributes referenced in R properties validated based on descriptor value
metadata.descriptor	Descriptor file path, defines all metadata columns
metadata.nullValue	Define how null values are represented in metadata
metadata.commentChar	Define how comments are indicated in metadata
report.numHits	Options: Y/N. If Y, and add #Hits to metadata
report.numReads	Options: Y/N. If Y, and add #Reads to metadata
report.fullTaxonomyNames	Options: Y/N. If Y, ParserModule will use full taxonomy names in output tables
report.addGenusToSpeciesName	Options: Y/N. If Y, ParserModule adds genus prefix to species name in output tables

Property	Value
report.useGenusFirstInitial	Options: Y/N. If Y, ParserModule adds genus 1st initial prefix to species name in output tables
report.attributes	R script adds statistical models for metadata attributes
report.minOtuCount	ParserModule ignores OTU counts below min count
report.emptySpaceDelim	Reports separate genus and species using this value
report.taxonomyLevels	Options: domain, phylum, class, order, family, genus, species. Generate reports for listed taxonomy levels
email.sendNotification	Options: Y/N. If Y, sent notification email
email.sendQsub	Options: Y/N. If Y, attach qsub output and error files
email.maxAttachmentSizeMB	Max size (in MB) for log file attachment
email.encryptedPassword	Encrypted password from *email.from* account. If BioLockJ is passed a 2nd parameter (in addition to the config file), the 2nd parameter should be the clear-text password. The password will be encrypted and stored in the prop file for future use. WARNING: Base64 encryption is only a trivial roadblock for malicious users. This functionality is intended merely to keep clear-text passwords out of the configuration files and should only be used with a disposable *email.from* account.
email.from	Notification emails sent from this account, provided *email.encryptedPassword* is valid
email.to	Comma-separated email recipients list
r.logNormal	Options: Y/N. If Y, use relative abundance table in R
r.logBase	Options: 10/e. If e, use natural log (base e), otherwise use log base 10
r.maxTitleSize=25	Report OTU names trimmed after max # characters
r.rareOtuThreshold	If >1, R will filter OTUs below # provided. If <1, R will treat # as percentage and ignore OTUs not found in that percentage of table rows in each taxa level
r.filterAttributes	Ordered list of attributes to report on in R script

Property	Value
r.filterOperators	Ordered list of logical operators to apply to *r.filterAttributes*
r.filterValues	Ordered list of values to compare with *r.filterAttributes*
r.filterNaAttributes	Rows with NA values for attributes ignored in R script
r.numHistogramBreaks	Number of breaks in P-value histograms output by R script
exe.classifier	Classifier executable command
exe.classifierParams	Optional classifier parameters, excluding parameters generated by BioLockJ (input files, output files, #threads)
exe.rScript	Executable RScript command
exe.java	Executable Java command
exe.python	Executable python command
exe.gzip	Executable gzip command
exe.awk	Executable awk command
exe.bowtie	Executable bowtie2 command
exe.bowtie_params	Optional bowtie2 parameters
exe.pear	Executable PEAR command
exe.pear_params	Optional PEAR parameters
exe.samtools	Executable samtools command
exe.vsearch	Executable vsearch command
exe.vsearchParams	Optional vsearch parameters
rdp.minThresholdScore	Required RDP minimum threshold score for valid OTUs
qiime.pickOtuScript	Options: pick_closed_reference_otus.py, pick_de_novo_otus.py, pick_open_reference_otus.py
qiime.alphaDiversityMetrics	Options listed online: scikit-bio.org/docs/latest/generated/skbio.diversity.alpha.html
qiime.formatMetadata	Options: Y/N. If Y, format metadata to build QIIME mapping
qiime.preprocessInput	Options: Y/N. If Y, decompress gzipped files and/or convert fastq files into fasta format, as required by QIIME
qiime.pickOtus	Options: Y/N. If Y, execute *qiime.pickOtuScript*

Property	Value
qiime.mergeOtuTables	Options: Y/N. If Y, merge OTU tables generated by batched scripts calling pick_closed_reference_otus.py
qiime.removeChimeras	Options: Y/N. If Y, remove chimeras after open or de novo OTU picking using *exe.vsearch*
kraken.db	Path to Kraken database
slimm.db	Path to SLIMM database
slimm.refGenomeIndex	Path the bowtie2 reference genome index

2.4.1 Example BioLockJ Configuration

```
project.name=twinStudy_tp2
project.rootDir=/research/microbiome/biolockj
project.copyInputFiles=N
project.deleteTempFiles=N
project.classifierType=RDP

control.runOnCluster=Y
control.trimSeqs=Y
control.mergePairs=Y
control.rarefySeqs=Y
control.runClassifier=Y
control.runParser=Y
control.run_rScript=Y

input.dirs=/datasets/16s/twinStudy/fw,/datasets/16s/twinStudy/rv
input.ignoreFiles=Cleandata.stat
input.demultiplex=N
input.pairedReads=Y
input.forwardFileSuffix=_R1
input.reverseFileSuffix=_R2
input.trimPrefix=timepoint2
input.trimSuffix=.fq
input.rarefyMinNumSeqs=10000
input.rarefyMaxNumSeqs=100000
input.trimSeqPath=/research/microbiome/primers/twinStudyPrimers.txt
```

```
cluster.batchCommand=qsub -q copperhead
cluster.params=#PBS -l procs=8,mem=32GB,walltime=12:00:00
cluster.validateParams=Y
cluster.modules=rdp/2.12
script.exitOnError=Y
script.batchSize=8
script.chmodCommand=chmod 774
script.numThreads=8

metadata.file=/research/microbiome/metadata/twinStudyMetadata.txt
metadata.descriptor=/research/microbiome/twinStudyDescriptor.txt
metadata.nullValue=NA
metadata.commentChar=##

report.numHits=Y
report.numReads=Y
report.fullTaxonomyNames=N
report.addGenusToSpeciesName=N
report.useGenusFirstInitial=Y
report.attributes=maritalStatus, sex, bmi, age
report.minOtuCount=2
report.emptySpaceDelim=.
report.taxonomyLevels=phylum,class,order,family,genus

email.sendNotification=Y
email.sendQsub=N
email.maxAttachmentSizeMB=5
email.encryptedPassword=SlrotqvCPGsFhWkKxtpwkQ==
email.from=biolockj@gmail.com
email.to=msioda@uncc.edu

r.logNormal=Y
r.logBase=e
r.maxTitleSize=25
r.rareOtuThreshold=0.25
r.filterAttributes=age
r.filterOperators=<
r.filterValues=17
r.filterNaAttributes=maritalStatus
r.numHistogramBreaks=20

exe.classifier=/apps/rdp_2.12/dist/classifier.jar
```

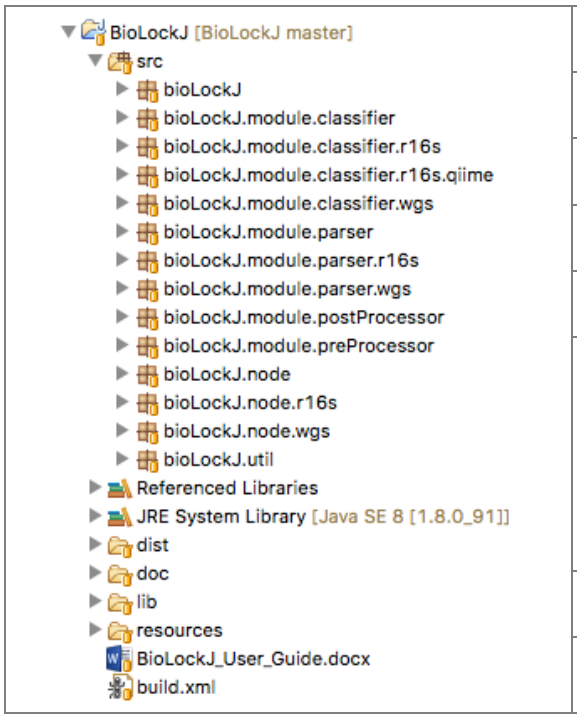
```
exe.classifierParams=/databases/silva128/rRNAClassifier.properties
exe.rScript=/apps/pkg/R-3.2.3/rhel7_u2-x86_64/gnu/bin/Rscript
exe.java=java
exe.python=python
exe.gzip=gzip
exe.awk=awk
exe.bowtie=bowtie2
exe.bowtie_params=no-unal, k 60
exe.pear=/apps/pear/pear-0.9.10-bin-64
exe.pear_params=t 150
exe.samtools=samtools
exe.vsearch=/apps/vsearch-2.4.3-linux-x86_64/bin/vsearch
exe.vsearchParams=db /databases/rdp_gold.fa

rdp.minThresholdScore=50

qiime.pickOtuScript=pick_closed_reference_otus.py
qiime.alphaDiversityMetrics=shannon,chao1,observed_species
qiime.formatMetadata=Y
qiime.preprocessInput=Y
qiime.pickOtus=Y
qiime.mergeOtuTables=Y
qiime.removeChimeras=N

kraken.db=/databases/kraken/all_bacteria_archaea_20170502

slimm.db=/apps/slimm/slimmDB_13K
slimm.refGenomeIndex=/databases/slimm/AB_13K_ref_genomes_bowtie2/AB_13K
```

3. System Architecture

- BioLockJ's modular design serves multiple purposes:
  1. Organize analysis by creating a logical separation of tasks and output file
  2. Facilitate failure recovery, easily identify where to restart pipeline after failure
  3. Promote pipeline extension, add modules to support new classifiers,

3.1 Java Project

- Screenshot from Eclipse Package Explorer (Java IDE)



BioLockJ	BioLockJ root directory
src	Java source code
dist	BioLockJ.jar ANT build script target
doc	Javadoc directory
lib	Required Java libraries (JAR files)
resources	➢ Config file templates ➢ Metadata files ➢ Descriptor files ➢ Primer files ➢ log4j.properties
BioLockJ User Manual	
build.xml	Apache ANT build script

3.1.1 Bundled JAR Files

- JAR files are located in the lib directory

JAR File	Java Classes	BioLockJ Invoking Method
commons-configuration-1.10	PropertiesConfiguration PropertiesConfigurationLayout	MailUtil.encryptAndStoreEmailPassword
commons-csv-1.4	CSVFormat CSVParser CSVRecord	MetadataUtil.processFile
commons-io-2.5	FileUtils IOFileFilter NameFileFilter TrueFileFilter WildcardFileFilter	Module.initInputFiles Module.setModuleInput ApplicationManager.copyInputDirs MailUtil.getAttachments
commons-lang-2.6	NestableException	Module.initInputFiles Module.setModuleInput ApplicationManager.copyInputDirs MailUtil.getAttachments
javax.mail	BodyPart InternetAddress Message MimeBodyPart MimeMessage MimeMultipart Multipart PasswordAuthentication Session Transport	MailUtil.sendEmailNotification
log4j-1.2.17 slf4j-api-1.7.22 slf4j-log4j12-1.7.22	Logger LoggerFactory	BioLockJ.initializeGlobalProps

3.1.2 Logging Framework

- Simple Logging Facade for Java (SLF4J) wraps the chosen logging framework
- The logging framework implemented for BioLockJ is Log4J
- Adjust log specificity by setting the log level in log4j.properties (default = INFO)
- System property ${LOG_FILE} is set based on configuration property: *project.name*
- Logger initialized in bioLockJ.ApplicationManager.buildNewProject method

3.1.3 Log4J Configuration

```
# Set log level and targets
log4j.rootLogger=INFO, file, stdout

# Configure command line output
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %-5p - %m%n

# Configure log file output
log4j.appender.file=org.apache.log4j.FileAppender
log4j.appender.file.File=${LOG_FILE}
log4j.appender.file.Append=false
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d %-5p - %m%n
```

3.1.4 Log Levels

Level	Log Level Description
DEBUG	All messages are logged, may result in large log files & impact performance
INFO	Informational, warning, and error messages are logged
WARN	Warning and error messages are logged
ERROR	Only error messages are logged

3.2 BioLockJ Module Overview

- Module execution occurs one at a time and in the order specified in the table below
- A module will run if it's control property is set to Y
- The first module configured to run uses *input.dirs* as input
- Additional modules use output from the previous module as input
- BioLockJ modules extend bioLockJ.Module.java

#	Module	Description	Control Property
1	SeqTrimmer	Remove primers from sequence files	*control.trimSeqs*
2	PairedSeqMerger	Merge paired reads	*control.mergePairs*
3	Rarefier	Discard samples with < min # seqs Randomly select max # seqs/sample	*control.rarefySeqs*
4	QiimePreProcessor	Decompress gzipped files Convert fastq to fasta Generate QIIME mapping file	*qiime.preprocessInput*
5a.1	ClosedRefClassifier	Batch reads for parallel processing Pick closed ref OTUs for QIIME	*qiime.pickOtus*
5a.2	MergeOtuTables	Merge OTU tables from 5a.1	*qiime.mergeOtuTables*
5b	OpenRefClassifier	Pick open ref OTUs for QIIME	*qiime.pickOtus*
5c	DeNovoClassifier	Pick de novo OTUs for QIIME	*qiime.pickOtus*
6	ClassifierModule	Build raw count and relative abundance tables	*control.runClassifier*
7	ParserModule	Merge metadata with classifier output to generate raw count and relative abundance tables for each level in: *report.taxonomyLevels*	*control.runParser*
8	RScriptBuilder	Build summary tables, p-value histograms, box-plots in R	*control.run_rScript*

***** Modules 4 – 5 are QIIME specific *****

3.2.1 Module Directory Structure

```
|--- project.rootDir                    Defined in configuration file
        |--- project.name_<ts>          Defined in configuration file & <yyyyMMdd_kkmmss>
                |--- #_module.name      # module (in the run order) & module name
                        |--- failures   Contains empty files indicating cause of script failures
                        |--- output     Module output (also input for next module)
                        |--- qsub       Contains output/error logs for each script run on cluster
                        |--- scripts    Contains module scripts & empty status indicator files
                        |--- temp       Contains intermediate output not passed next module
```

Directory	Usage
failures	Exists if module runs bash scripts
output	Exists for all modules
qsub	Exists if module bash scripts run on cluster
scripts	Exists if module runs bash or R scripts
temp	Exists if module stores intermediate output not passed to next module

Example Module Layout	Control Property
|--- projects	*project.rootDir=/projects*
|--- twinStudy_20170629_104521	*project.name=twinStudy*
|--- 0_SeqTrimmer	*control.trimSeqs=Y*
|--- output	
|--- 1_PairedSeqMerger	*control.mergePairs=Y*
|--- failures	
|--- output	
|--- temp	
|--- qsub	*control.runOnCluster=Y*
|--- scripts	
|--- 2_Rarefier	*control.rarefySeqs=Y*
|--- output	
|--- temp	
|--- 3_RdpClassifier	*control.runClassifier=Y*
|--- failures	
|--- output	
|--- qsub	*control.runOnCluster=Y*
|--- scripts	
|--- 4_RdpParser	*control.runParser=Y*
|--- output	
|--- temp	
|--- 5_RScriptBuilder	*control.run_rScript=Y*
|--- output	
|--- scripts	

3.2.2 Module Output

<table>
<tr><th>#</th><th>Module</th><th>Output Files</th></tr>
<tr><td>1</td><td>SeqTrimmer</td><td>Trimmed fasta/fastq sequence files</td></tr>
<tr><td>2</td><td>PairedSeqMerger</td><td>Merged fastq files</td></tr>
<tr><td>3</td><td>Rarefier</td><td>Rarefied fasta/fastq sequence files</td></tr>
<tr><td>4</td><td>QiimePreProcessor</td><td>QIIME only, Fast-A files & QIIME Mapping file</td></tr>
<tr><td>5a.1</td><td>ClosedRefClassifier</td><td>QIIME only, multiple otu_table.biom files</td></tr>
<tr><td>5a.2</td><td>MergeOtuTables</td><td>QIIME only, one otu_table.biom file</td></tr>
<tr><td>5b</td><td>OpenRefClassifier</td><td>QIIME only, one otu_table.biom file</td></tr>
<tr><td>5c</td><td>DeNovoClassifier</td><td>QIIME only, one otu_table.biom file</td></tr>
<tr><td>6</td><td>ClassifierModule</td><td>Classifier output:
<table>
<tr><td>RDP</td><td><sample_id>_reported.tsv</td></tr>
<tr><td>QIIME</td><td>otu_by_taxa_level/out_table_L<#>.txt
(OTU Levels #1 – #7 = domain – species)</td></tr>
<tr><td>Kraken</td><td><sample_id>_reported.tsv</td></tr>
<tr><td>MetaPhlAn</td><td><sample_id>_reported.tsv</td></tr>
<tr><td>SLIMM</td><td><sample_id>_<taxonomy_level>_reported.tsv</td></tr>
</table>
If report.numReads=Y, add numReads to:
• metadata.file
• metadata.descriptor</td></tr>
<tr><td>7</td><td>ParserModule</td><td>For each level defined in report.taxonomyLevels:
• <taxonomy_level>_RawCount_metaMerged.txt
• <taxonomy_level>_LogNormal_metaMerged.txt

If report. numHits =Y, add numHits to:
• metadata.file
• metadata.descriptor</td></tr>
<tr><td>8</td><td>RScriptBuilder</td><td>For each level defined in report.taxonomyLevels:
• boxplots_<taxonomy_level>.pdf
• meta_pValuesFor_<taxonomy_level>.txt</td></tr>
</table>

4. QIIME Modules

- Detailed QIIME script descriptions available online: qiime.org/scripts
- Module 4: Qiime Preprocessor prepares input for any OTU picking method
- Property *qiime.pickOtus* determines the OTU picking method (5a/5b/5c)
- Module 5a.1: Closed Reference Classifier batches input for parallel processing
- Module 5a.2: Merge OTU Tables collates results from Module 5a.1
- Module 5b: Open Reference Classifier picks closed reference OTUs and attempts to classify the remaining sequences via the de novo method (default UCLUST)
- Module 5c: De Novo Classifier picks OTUs via clustering algorithm (default UCLUST)
- Module 6: Qiime Classifier builds taxonomy reports with OTUs counts by sample

4.1 Qiime Preprocessor

- This module prepares input files for classification by QIIME OTU picking script
- Decompress gzipped files and convert fastq to fasta format, if needed
- Create QIIME mapping by adding and/or reordering *metadata.file* columns
- If adding fields to *metadata.file*, assign categorical data type in *metadata.descriptor*

Module #	4
Java Class	bioLockJ.module.classifier.r16s.qiime.QiimePreprocessor.java
QIIME Scripts	print_qiime_config.py — Print version information to qsub output file validate_mapping_file.py — Build and/or verify QIIME mapping file
Control Properties	*control.runClassifier* — Required value = Y *qiime.preprocessInput* — Required value = Y *qiime.formatMetadata* — If value = Y, build qiimeMapping.txt
Input	Fasta or Fastq forward reads, which may be gzipped (paired reads must be merged via PairedSeqMerger module)

<table>
<tr><td>Temp Directory</td><td>
<table>
<tr><th>File</th><th>Description</th></tr>
<tr><td>*.fasta or *.fastq</td><td>Decompressed gzipped sequence files</td></tr>
<tr><td>qiimeMapping.txt</td><td>If qiime.formatMetadata=Y, add QIIME mapping fields in metadata.file if needed</td></tr>
<tr><td>orderedMapping.txt</td><td>If qiime.formatMetadata=Y, reorder required columns in metadata.file if needed</td></tr>
</table>
</td></tr>
<tr><td>Output Directory</td><td>
<table>
<tr><th>File</th><th>Description</th></tr>
<tr><td>*.fasta</td><td>Primary output (Fast-A files)
QiimeClassifier module pick_*_otus.py input</td></tr>
<tr><td>mapping/*_corrected.txt</td><td>validate_mapping_file.py output</td></tr>
<tr><td>metadata.descriptor</td><td>If qiime.formatMetadata=Y, add new fields to descriptor file if needed</td></tr>
</table>
</td></tr>
</table>

4.2 Closed Reference Classifier

- This module batches input for parallel processing based on *script.batchSize*
- Each batch contains a subset of fasta files and a QIIME mapping (batchMapping.txt)
- Pick closed reference OTUs from a reference database for each batch
- Each batch outputs classifies reads by Sample ID into its own otu_table.biom file
- The MergeOtuTables module must run next to combine otu_table.biom files

<table>
<tr><td>Module #</td><td>5a.1</td></tr>
<tr><td>Java Class</td><td>bioLockJ.module.classifier.r16s.qiime.ClosedRefClassifier.java</td></tr>
<tr><td>QIIME Scripts</td><td>
<table>
<tr><td>add_qiime_labels.py</td><td>Build combined_seqs.fna</td></tr>
<tr><td>pick_closed_reference_otus.py</td><td>Build batch otu_table.biom files</td></tr>
</table>
</td></tr>
<tr><td>Control Properties</td><td>
<table>
<tr><td>control.runClassifier</td><td>Required value = Y</td></tr>
<tr><td>qiime.pickOtus</td><td>Required value = Y</td></tr>
<tr><td>qiime.pickOtuScript</td><td>Required value = pick_closed_reference_otus.py</td></tr>
</table>
</td></tr>
</table>

Input	Fast-A sequence files
Temp Directory	N/A
Output Directory	A *batch_#* directory for every *script.batchSize* # fasta files contains:

File	**Description**
otu_table.biom	pick_closed_reference_otus.py output MergeOtuTables module input
97_otus.tree	pick_closed_reference_otus.py output
batchMapping.txt	QIIME mapping for fasta/* files
combined_seqs.fna	add_qiime_labels.py output
fasta/*.fasta	*script.batchSize* # fasta files
log_*.txt	pick_closed_reference_otus.py log file
uclust_ref_picked_otus/*	pick_closed_reference_otus.py output

4.3 Merge OTU Tables

- This module combines otu_table.biom files output by Module 5a.1 ClosedRefClassifier

Module #	5a.2
Java Class	bioLockJ.module.classifier.r16s.qiime.MergeOtuTables.java
QIIME Scripts	merge_otu_tables.py — Merge ClosedRefClassifier otu_table.biom files
Control Properties	*control.runClassifier* — Required value = Y *qiime.mergeOtuTables* — Required value = Y *qiime.pickOtuScript* — Required value = pick_closed_reference_otus.py
Input	ClosedRefClassifier/output/batch_*/otu_table.biom files
Temp Directory	N/A

Output Directory	<table><tr><th>File</th><th>Description</th></tr><tr><td>**otu_table.biom**</td><td>merge_otu_tables.py output QiimeClassifier module input</td></tr></table>

4.4 Open Reference Classifier

- QIIME picks closed reference OTUs from a reference database (see Section 5.2)
- Unclassified reads are clustered with QIIME de novo method (see Section 5.5)
- If *qiime.removeChimeras=Y,* vsearch is used to find chimeric sequences
- QIIME script filter_otus_from_otu_table.py is used to remove chimeric sequences

Module #	5b
Java Class	bioLockJ.module.classifier.r16s.qiime.OpenRefClassifier.java
QIIME Scripts	<table><tr><td>add_qiime_labels.py</td><td>Build combined_seqs.fna</td></tr><tr><td>pick_open_reference_otus.py</td><td>Build otu_table_*.biom files</td></tr><tr><td>filter_otus_from_otu_table.py</td><td>Filter chimeras from otu_table_*.biom Build primary output: otu_table.biom</td></tr></table>
Control Properties	<table><tr><td>*control.runClassifier*</td><td>Required value = Y</td></tr><tr><td>*qiime.pickOtus*</td><td>Required value = Y</td></tr><tr><td>*qiime.pickOtuScript*</td><td>Required value = pick_open_reference_otus.py</td></tr><tr><td>*qiime.removeChimeras*</td><td>If value = Y, remove chimeras with vsearch</td></tr></table>
Input	Fast-A sequence files
Temp Directory	N/A

Output Directory	

File	Description
otu_table.biom	Primary output QiimeClassifier module input
chimeras.fasta	vsearch output filter_otus_from_otu_table.py input
combined_seqs.fna	add_qiime_labels.py output
final_otu_map*.txt	pick_open_reference_otus.py output
index.html	pick_open_reference_otus.py output
log_*.txt	pick_open_reference_otus.py log file
new_refseqs.fna	pick_open_reference_otus.py output
nochimeras.fasta	vsearch output
otu_table_mc*.biom	pick_open_reference_otus.py output filter_otus_from_otu_table.py input
pynast_aligned_seqs/*	pick_open_reference_otus.py output
rep_set.fna	pick_open_reference_otus.py output vsearch input
rep_set.tre	pick_open_reference_otus.py output
step*_otus/*	pick_open_reference_otus.py output
uclust_assigned_taxonomy/*	pick_open_reference_otus.py output

4.4.1 QIIME Open Reference [index.html]

- The Open Reference index.html report provides a summary of output files

Run summary data	
Run summary data	log_20170708144403.txt
Taxonomy assignments	
OTU taxonomic assignments	rep_set_tax_assignments.txt
OTU tables	
OTU table exluding OTUs with fewer than 2 sequences	otu_table_mc2.biom
OTU table exluding OTUs with fewer than 2 sequences and including OTU taxonomy assignments	otu_table_mc2_w_tax.biom
OTU table exluding OTUs with fewer than 2 sequences and sequences that fail to align with PyNAST and including OTU taxonomy assignments	otu_table_mc2_w_tax_no_pynast_failures.biom
Trees	
OTU phylogenetic tree	rep_set.tre
OTU maps	
Final map of OTU identifier to sequence identifers excluding OTUs with fewer than 2 sequences	final_otu_map_mc2.txt
Sequences	
OTU representative sequences	rep_set.fna
New reference sequences (i.e., OTU representative sequences plus input reference sequences)	new_refseqs.fna

4.5 De Novo Classifier

- QIIME uses the de novo OTU picking script to cluster reads and assign taxonomy
- If *qiime.removeChimeras=Y,* vsearch is used to find chimeric sequences
- QIIME script filter_otus_from_otu_table.py is used to remove chimeric sequences

Module #	5c
Java Class	bioLockJ.module.classifier.r16s.qiime.DeNovoClassifier.java
QIIME Scripts	add_qiime_labels.py — Build combined_seqs.fna pick_open_reference_otus.py — Build otu_table_*.biom files filter_otus_from_otu_table.py — Filter chimeras from otu_table_*.biom Build primary output: otu_table.biom
Control Properties	*control.runClassifier* — Required value = Y *qiime.pickOtus* — Required value = Y *qiime.pickOtuScript* — Required value = pick_de_novo_otus.py *qiime.removeChimeras* — If value = Y, remove chimeras with vsearch
Input	Fast-A sequence files
Temp Directory	N/A
Output Directory	Fast-A files must all be processed together to find de novo OTUs

File	**Description**
otu_table.biom	Primary output QiimeClassifier module input
chimeras.fasta	vsearch output filter_otus_from_otu_table.py input
combined_seqs.fna	add_qiime_labels.py output
log_*.txt	pick_de_novo_otus.py log file
nochimeras.fasta	vsearch output
pynast_aligned_seqs/*	pick_de_novo_otus.py output
rep_set/*	pick_de_novo_otus.py output vsearch input
rep_set.tre	pick_de_novo_otus.py output
uclust_assigned_taxonomy/*	pick_de_novo_otus.py output
uclust_picked_otus/*	pick_de_novo_otus.py output

4.6 QIIME Classifier

- QIIME Classifier builds taxonomy level reports by counting OTUs for each sample
- Alpha metrics can be included in R script by adding to *report.attributes*, if configured

Module #	6
Java Class	bioLockJ.module.classifier.r16s.QiimeClassifier.java
QIIME Scripts	summarize_taxa.py — Create otu_by_taxa_level/* alpha_diversity.py — Create alphaDiversity.txt add_alpha_to_mapping_file.py — Add alpha metrics to QIIME mapping
Control Properties	*control.runClassifier* — Required value = Y *project.classifierType* — Required value = QIIME *qiime.alphaDiversityMetrics* — If *qiime.alphaDiversityMetrics* configured add alpha metrics to qiimeMapping.txt and *metadata.descriptor*
Input	otu_table.biom
Temp Directory	N/A
Output Directory	**File** — **Description** **otu_by_taxa_level/*** — summarize_taxa.py output alphaDiversity.txt — alpha_diversity.py output otuSummary.txt — “biom summarize-table” output qiimeMapping.txt — add_alpha_to_mapping_file.py output *metadata.descriptor* — Add *qiime.alphaDiversityMetrics* fields to descriptor file, if any

5. Extending the Pipeline

- There are several options to extend the pipeline to meet your project needs:
  - Add complex filters or multi-variate statistical models to the R Script
  - Run datasets through multiple classifiers to compare accuracy and performance
  - Add new Java modules to support your favorite classifier

5.1 Enhancing the R Script

- R Script Module identifies significant OTUs for individual metadata *report.attributes*
- Generate new reports by updating report.r with complex filters and new models

5.2 Comparing Classifiers

- Update *project.classifierType* & add the classifier specific properties to switch classifiers
  - Compare significant OTUs output by RScriptBuilder
  - Compare raw count and relative abundance tables output by ParserModule
  - Compare overall performance with the BioLockJ runtime summary

5.3 Adding New Classifiers

- Add support for additional classifiers by extending 3 abstract Java classes:
  1. ClassifierModule.java
  2. ParserModule.java
  3. OtuNode.java

5.3.1 Extending ClassifierModule.java

- Validate new classifier properties in checkDependencies()
- Implement abstract methods that build bash scripts for new classifier:

```
protected abstract List<List<String>> buildScript( final List<File> files )
protected abstract List<List<String>> buildScriptForPairedReads( final List<File> files )
```

- Each inner List<String> holds a group of statements used to classify one sequence file

5.3.2 Extending ParserModule.java

- Implement abstract method to build BioLockJ OTU Nodes:

```
protected void createOtuNodes()
```

- Parse classifier output and instantiate new OtuNode subclass for each line
- Call addOtuNode(id, node) to map Sample ID to OTU count

5.3.3 Extending OtuNode.java

- Constructor accepts a String value representing one line of classifier output
- Parse line for OTU count

Appendix A: Contact Information

BioLockJ Administrator:	biolockj@gmail.com
Project URL:	github.com/mikesioda/BioLockJ
Principal Investigator:	Dr. Anthony Fodor <afodor@uncc.edu>
Lead Developer:	Michael Sioda <msioda@uncc.edu>