bioKepler: A Comprehensive Bioinformatics Scientific Workflow Module for Distributed Analysis of Large-Scale Biological Data

Ilkay Altintas¹, Jianwu Wang², Daniel Crawl¹, Shweta Purawat¹

¹ San Diego Supercomputer Center, UC San Diego
² UMBC

WorDS.sdsc.edu
A Toolbox with Many Tools

• Data
  • Search, database access, IO operations, streaming data in real-time...
• Compute
  • Data-parallel patterns, external execution, ...
• Network operations
• Provenance and fault tolerance

Need expertise to identify which tool to use when and how!
Require computation models to schedule and optimize execution!
Workflows are Used in These Diverse Scenarios in Biological Sciences

Data

- Sequencers
- Sensor networks
- Medical imaging

Data Generation

- Often for data reduction
- In real-time or offline

Data Acquisition

Many forms
- Data-intensive
- HPC
- Local Exploratory

Data Analysis

- From analysis to searchable results
- Standardization
- Auto generation of methods and materials

Data Publication

Workflows foster collaborations!

- Flexibility and synergy
- Optimization of resources
- Increasing reuse
- Standards compliance
CAMERA Example:

Using Scientific Workflows and Related Provenance for Collaborative Metagenomics Research

Community Cyberinfrastructure for Advanced Microbial Ecology Research and Analysis (CAMERA)

http://camera.calit2.net
CAMERA is a Collaborative Environment

Data Cart
Multiple Available Mixed collections of CAMERA Data (e.g. projects, samples)

User Workspace
Single workspace with access to all data and results (private and shared)

Group Workspace
Share specified User Workspace data with

Data Discovery
GIS and Advanced query options

Data Analysis
Workflow based analysis
Workflows are a Central Part of CAMERA

• **CAMERA-supported**
  – 28 existing workflows

• **Workflows under development**
  – Fragment Recruitment Viewer
  – Next Generation Sequencing
  – VIROME Pipeline
  – Standalone bioinformatics tools
  – National Center for Genome Research
  – Joint Genome Institute

• **User built**
  – Currently running in a sandbox
  – Will be ported to a virtual cloud environment

More than 1500 workflow submissions monthly!

• **Inputs**: from local or CAMERA file systems; user-supplied parameters
• **Outputs**: sharable with a group of users and links to the semantic database

All can be reached through the CAMERA portal at: http://portal.camera.calit2.net
DATA ANALYSIS

CAMERA utilizes workflows to launch data analysis tools. Workflows are configurable analysis packages that can be applied to data within the CAMERA workspace or to data uploaded from the local system. CAMERA workflows include:

- Metagenomic data annotation and clustering (RAMMCAAP)
- BLAST tools (Click here for a complete list of CAMERA Reference Datasets)
- RNA and Orf Prediction
- Click here for a complete list of CAMERA workflows
- Job Submission Policy
- FASTA Validation Guidelines

To Launch a Workflow:

1. Select an analysis from the workflow menu and click the 'Start >' button at the bottom of the page.
2. Fill out the analysis parameters and click the 'Submit Workflow!' button at the bottom of the page. For additional information regarding the parameters, mouse-over the circular ‘?’ button.
3. Use the Results and Status page to view and share results. For BLAST, use the dedicated Blast Results viewer.

Upload User Workflows (Beta):

CAMERA provides a collaborative environment for analysis and data. As part of this environment, users can upload and share their own workflows/analysis with their colleagues or with the greater scientific community. Please note this new feature is in a BETA state and may have problems. Initially, this area is for those who already have an understanding of the Kepler workflow system.

Click here to upload a workflow. Note that you must create a group to associate with the user workflow.

To get started with workflow development, please go here
By submitting or running a workflow, you are agreeing to the Terms and Conditions

Queue Status (Last Update: 3/20 20:30:03 PDT)
- CAMERA Cluster Utilization: 95%
  Workflows / Jobs* Running: 4 / 281
  Workflows / Jobs* Queued: 4 / 2231
- CAMERA Compute Cloud: Enabled
  Workflows / Jobs* Running: 1 / 320

* Each workflow is composed of numerous jobs.
CAMERA Workflows

CAMERA supported Workflows

- TBLASTN
- Blast Kegg
- Metagenomic data annotation and clustering
- Assembly
- DNA clustering
- tRNA prediction by hmer
- rRNA prediction by blastn

This is the full RAMMCAP pipeline for analysis of metagenomic sequences. It accepts a FASTA file of raw reads. The pipeline identifies the tRNA, rRNA, and ORFs from the reads. It then performs clustering analysis on the reads and the ORFs. The ORFs are annotated against PFAM, TIGRFAM, and COG.

Start >
Execute Workflow: Metagenomic data annotation and clustering

This is the full RAMMCAP pipeline for analysis of metagenomic sequences. It accepts a FASTA file of raw reads. The pipeline identifies the tRNA, rRNA, and ORFs from the reads. It then performs clustering analysis on the reads and the ORFs. The ORFs are annotated against PFAM, TIGRFAM, and COG.

Default Parameters

Parameters

- Run read clustering
- Run ORF clustering
- Run annotation
- Select ORF prediction method
- Select rRNA prediction method
- Input FASTA file of reads

Submit Workflow!
Execute Workflow: Metagenomic data annotation and clustering

Parameters for ORF call by six reading frame translation, default value is "-l 30 -L 30 -t 11"

- l 30 - L 30 means the cutoff length of ORFs is 30 amino acid
- t 11 means translation table 11

Options:
- l minimal length oforf, default 20
- - L minimal length of orf between 2 stop codons, default 40.
- t translation table, default 1
- - b ORF begin option, default 2
  1: start at the beginning of DNA sequence or after previous stop codon
  2: start with the first ATG if there is a stop codon upstream
  We don’t know which ATG is the real start, but for prokaryotic DNA,
a fragment between a stop codon and the first ATG can not be part of reality.
  Therefore, - b 2 is recommended for prokaryotic
- - c ORF end option, default 1
  1: end at the end of DNA sequence or at a stop codon
  2: must end at a stop codon

ORF clustering second run

Read clustering

E-value cutoff for Pfam

E-value cutoff for Tigrfam

E-value cutoff for COG
CAMERA Job Status

Job Status

Check the status of your jobs, download results, and export results to the CAMERA 2.0 semantic database.

My Jobs

<table>
<thead>
<tr>
<th>Job Name</th>
<th>Workflow</th>
<th>Status</th>
<th>Results</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANME2_ann</td>
<td>Metagenomic data annotation and chDONE</td>
<td>View</td>
<td>Download</td>
<td></td>
</tr>
<tr>
<td>ANME1_ann</td>
<td>Metagenomic data annotation and chDONE</td>
<td>View</td>
<td>Download</td>
<td></td>
</tr>
<tr>
<td>Mary90_vs_allRef</td>
<td>MEGA Blast</td>
<td>Done</td>
<td>View</td>
<td>Download</td>
</tr>
<tr>
<td>blasto_test3</td>
<td>BLASTP</td>
<td>Done</td>
<td>View</td>
<td>Download</td>
</tr>
</tbody>
</table>

Shared Jobs

<table>
<thead>
<tr>
<th>Job Name</th>
<th>Workflow</th>
<th>Project</th>
<th>User</th>
<th>Date</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC5_1_leads Workflow 11/30/2010 01:27 PM</td>
<td>Metagenomic data</td>
<td>carbon cyclin jsteele</td>
<td>11/30/2010 01:43 PM</td>
<td>View</td>
<td>Download</td>
</tr>
<tr>
<td>small job</td>
<td>Metagenomic data</td>
<td>October EPIE yoyornan</td>
<td>09/10/2010 05:57 PM</td>
<td>View</td>
<td>Download</td>
</tr>
<tr>
<td>My Workflow 12/01/2009 09:52 AM</td>
<td>ELASTN</td>
<td>SAB Worksp mmoran</td>
<td>12/01/2009 09:53 AM</td>
<td>results</td>
<td></td>
</tr>
<tr>
<td>SAB 1500 sequences exact dupes</td>
<td>Metagenomic data</td>
<td>SAB Worksp mmoran</td>
<td>11/25/2009 11:38 AM</td>
<td>View</td>
<td>Download</td>
</tr>
</tbody>
</table>
CAMERA Workflow Results

### RAMMCAP Workflow Results

**Summary**

<table>
<thead>
<tr>
<th>Annotation Category</th>
<th>Total</th>
<th>Total (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COGs</td>
<td>23041</td>
<td>8.61</td>
</tr>
<tr>
<td>PFAMs</td>
<td>23114</td>
<td>8.63</td>
</tr>
<tr>
<td>TGRFAM</td>
<td>38038</td>
<td>14.21</td>
</tr>
<tr>
<td>tRNA</td>
<td>199</td>
<td>0.07</td>
</tr>
<tr>
<td>rRNA</td>
<td>205898</td>
<td>76.92</td>
</tr>
</tbody>
</table>

- **Workflow Name:** My Workflow 03/08/2010 11:52 PM DICE
- **Workflow Type:** Metagenomic data annotation and clustering
- **Workflow Run ID:** 1268121202419
- **Workflow version:** 1.0

**Download File**

Will download all workflow output files and images in a zip file.

---

**SDSC**

SAN DIEGO SUPERCOMPUTER CENTER

**WOrDS**

Center
Pushing the boundaries of existing infrastructure and workflow system capabilities
New Requirements from the User Community

- **Increase reuse**
  - best development practices by the scientific community
  - other bio packages

- **Increase programmability by end users**
  - users with various skill levels
  - to formulate actual domain specific workflows

- **Increase resource utilization**
  - optimize execution across available computing resources
  - in an efficient, transparent and intuitive manner

- **Make analysis a part of the end-to-end scientific model from data generation to publication**
RAMMCAP – Rapid Clustering and Functional Annotation for Metagenomic Sequences

Annotation features:
• tRNA prediction (tRNAscan)
• rRNA prediction (meta_RNA, BLAST)
• ORF call (ORF_finder, Metagene)
• RPS-BLAST against COG etc
• HMMER against Pfam / Tigrfam

- Clustering of reads
- Multi-step clustering of ORFs
- GO assignment
- EC number assignment
A number of bioinformatics tools are used in RAMMNCAP

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST</td>
<td>Scalable parallel database search with blastn, blastp, tblastn, blastx, tblastx</td>
</tr>
<tr>
<td>MegaBLAST</td>
<td>Fast database search with MegaBLAST</td>
</tr>
<tr>
<td>Diversity</td>
<td>Diversity analysis for viral metagenome</td>
</tr>
<tr>
<td>QC</td>
<td>Quality control for 454 raw reads</td>
</tr>
<tr>
<td>CD-HIT-454</td>
<td>Identify artificial duplicates from 454 reads</td>
</tr>
<tr>
<td>RAMMNCAP</td>
<td>Metagenome annotation</td>
</tr>
<tr>
<td></td>
<td>- rRNA, tRNA, ORF prediction</td>
</tr>
<tr>
<td></td>
<td>- reads and ORF clustering</td>
</tr>
<tr>
<td></td>
<td>- reads and ORF information</td>
</tr>
<tr>
<td></td>
<td>- family and function annotation (Pfam, TIGRfam, COG)</td>
</tr>
<tr>
<td></td>
<td>- Gene Ontology and Enzyme Classification annotation</td>
</tr>
<tr>
<td></td>
<td>- Combined annotation summary</td>
</tr>
<tr>
<td>FRV</td>
<td>Fragment Recruitment Viewer</td>
</tr>
<tr>
<td>Assembly</td>
<td>Consensus-based meta-assembler for 454 reads</td>
</tr>
<tr>
<td>KEGG</td>
<td>Pathway annotation by search KEGG database with blastp</td>
</tr>
<tr>
<td>RDP binning</td>
<td>Taxonomy binning of rRNA sequences using RDP classifier</td>
</tr>
<tr>
<td>BLAST binning</td>
<td>Taxonomy binning by querying ref. rRNA DB using blastn</td>
</tr>
<tr>
<td>tRNA</td>
<td>Identification of tRNAs from fragments using tRNA-scan</td>
</tr>
<tr>
<td>Meta-RNA</td>
<td>Identification of rRNAs from fragments using HMM</td>
</tr>
<tr>
<td>BLAST-RNA</td>
<td>Identification of rRNAs by querying ref. rRNA DB using blastn</td>
</tr>
<tr>
<td>ORF_finder</td>
<td>ORF call by six reading frame translation</td>
</tr>
<tr>
<td>Metagene</td>
<td>ORF call by Metagene</td>
</tr>
<tr>
<td>FragGeneScan</td>
<td>ORF call with FragGeneScan from 454 reads</td>
</tr>
<tr>
<td>Pfam</td>
<td>Protein family annotation against Pfam using HMMER</td>
</tr>
<tr>
<td>TIGRfam</td>
<td>Protein family annotation against TIGRfam using HMMER</td>
</tr>
<tr>
<td>COG</td>
<td>Protein family annotation against NCBI COG using rps-blast</td>
</tr>
<tr>
<td>KOG</td>
<td>Protein family annotation against NCBI KOG using rps-blast</td>
</tr>
<tr>
<td>PRK</td>
<td>Protein family annotation against NCBI PRK using rps-blast</td>
</tr>
<tr>
<td>CD-HIT-EST</td>
<td>Clustering of reads</td>
</tr>
<tr>
<td>CD-HIT</td>
<td>Clustering of ORFs</td>
</tr>
<tr>
<td>H-CD-HIT</td>
<td>Multiple level clustering of ORFs into ORF family</td>
</tr>
</tbody>
</table>
Original implementation of the annotation workflow in Kepler

A green box is called a ‘actor’, which performs a task.

This special actor represents an annotation component, such as BLAST search.

Workflow parameters, which can be specified by users in portal, are passed to workflow components.

Data flow is divided.
Each actor was a wrapper to a web service!
RAMMCAP – Rapid Clustering and Functional Annotation for Metagenomic Sequences

Data size
- KB
- MB
- GB
- TB

CPU time
- Minute
- Hour
- Day
- Year

Memory
- GB
- TB

Parallel
- No need
- No
- Multi threading
- MPI
- Map Reduce

QC
cd-hit
metagene
tRNA
hmmer
blast
Another cases – RNA-seq / genomic / metagenomic

1. Raw reads
   - Reads QC
   - Reads QC

2. HQ reads
   - Assemble
     - Velvet, SOAPdenovo, Abyss, Oases, Trinity
     - BWA
     - Bowtie
     - BLAST

3. Alignments
   - Contigs
     - Further analysis

Data size
- KB
- MB
- GB
- TB

CPU time
- Minute
- Hour
- Day
- Month

Memory
- GB
- 100 GB

Parallel
- No need
- No
- Multi threading
- MPI
- Map Reduce
bioKepler implementation:
Using bioActors instead of wrapper actors

Wrapper Actors
• Need implementation of underlying computational tools

bioActors
• Reusable
• Multiple execution modes
• Build-in parallel execution capabilities
A coordinated ecosystem of biological and technological packages for bioinformatics!
The bioKepler Approach

• **Parallel** Computation Framework
  – Use Distributed Data-Parallel (DDP) frameworks, e.g., MapReduce, and other parallelization methods to execute subworkflows

• **bioActors**
  – Configurable and reusable higher-order components for bioinformatics and computational biology

• **Transparent** support for different execution engines and computational environments

• Deployment on **diverse** environments
Reuse, Programmability, Execution

- Funded by NSF ABI & CI Reuse programs - Altintas (PI) and Li (Co-PI)
- Development of a comprehensive bioinformatics scientific workflow module for distributed analysis of large-scale biological data

Big improvement on usability and programmability by end users!

Kepler supports
- Workflows
- Other third party programming tools, e.g., R, Matlab, KNIME
- Extensible task and data parallelization
- Service orientation
- Execution on multiple engines, e.g., SDF, SGE, Hadoop

bioKepler
CloudBioLinux
Galaxy
Bio-Linux

Kepler
- CORE
- Distributed Data Parallel
- Provenance
- Reporting
- Run Manager
- ...
bioKepler’s Conceptual Framework

Bioinformatics Tools:
- Mapping
- Clustering
- Assembly

Customize & Integrate

Bioinformatician

Data-Parallel Execution Patterns:
- Map-Reduce
- Master-Slave
- All-Pairs

Provenance:
- Data Lineage
- Execution History

Reporting:
- Report Designer
- PDF Generation

Run Manager:
- Search
- Tag

Fault-Tolerance:
- Alternatives
- Error Handling

Deploy & Execute

BioKepler

bioActors:
- BLAST
- HMMER
- CD-HIT

Workflow

Director

Scheduler

Executable Workflow Plan

Execution Engine

Deploy & Execute

Data Transfer:
- Amazon EC2
- Triton Resource
- FutureGrid
- Sun Grid Engine

Compute

Data:
- Ensembl
- Genbank
- CAMERA

SAN DIEGO SUPERCOMPUTER CENTER
bioActors

• Set of steps to execute a bioinformatics tool locally or in an external environment
  – Locally executable
  – Parallelized external execution
• Customizable by the user based on external packages
  – Tools imported from CloudBioLinux
• Tools are evaluated on their computational requirements
Transparent Execution includes Parallelization Solutions in Distributed Environments

• *Traditional parallel programming interfaces*
  – Examples: MPI and OpenMP
  – Hard to implement
  – Original sequential tools cannot be reused

• *Parallel job execution*
  – Examples: SGE and Condor
  – Original sequential tools can be reused
  – Create small jobs by splitting data or tasks
  – Hard to achieve data locality for each job

• *Data parallel job execution*
  – Examples: Hadoop and Stratosphere
  – Original sequential tools can be reused
  – Support customized and automatic data partition and distribution
  – Support data locality for each job through special distributed file system, HDFS
Distributed Data-Parallel bioActors

• Set of steps to execute a bioinformatics tool in DDP environment
• Customized from the ExecutionChoice actor
• Includes:
  – Data-parallel patterns, e.g., Map, Reduce, Cross, All-Pairs, etc., to specify data grouping
  – I/O to interface with storage
  – Data format specifying how to split and join
1. Search

2a. Choose Specific

2b. Choose Generic

2b. Create Sub-Workflow

3. Add to Workflow

4a. Execute

4b. Add to Larger Workflow

4c. Save in Library

User: Workflow Developer

bioActor Library

A1 A2 ... An

DDP Generic

DDP Blast

DDP Director

Workflow

Results

bioActor Library

DDP bioActor Usage Model

User: Workflow Developer
Status of bioActors

500+ bioActors are listed under current bioKepler release, ~40 of them are parallelized.

This workflow implements a computational procedure for automated identification of cell populations from multidimensional flow cytometry data with the FLOCK algorithm. It contains three major steps including Transformation (FCSTrans), Filtering (DAG), and Clustering (FLOCK). The workflow transfers user input data from front-end web portal to the backend workflow cloud virtual machine or to the cluster, carries computational analysis in background for multiple input files on Cloud VM or the SDSC Gordon Supercomputer, and uploads result back to the web portal. The workflow configures or by-pass filter stage depending user filter choice. The Dynamic Dataflow (DDF) director manages workflow execution.

See http://flowgate.jcvi.org

Authors: Shweta Purawat, Jianwu Wang, Ilkay Altintas, Robert Sinkovits @ SDSC
        Yu Qian, Rick Stanton, Hyunsoo Kim, Richard Scheuermann @ JCVI

April-25-2014
Example bioActors

- **Alignment**: BLAST, BLAT
- **Profile-Sequence Alignment**: PSI-BLAST
- **Hidden Markov Model**: HMMER
- **Mapping**: Bowtie, BWA, Samtools
- **Multiple Alignment**: ClustalW, Muscle
- **Clustering**: CD-HIT, Blastclust
- **Gene Prediction**: Glimmer, Genescan, Fraggenescan
- **tRNA prediction**: tRNA-scan, Meta-RNA
- **Phylogeny**: FastTree, RAxML
Example Workflows
Current Release

Downloadable as a package at: http://www.biokepler.org/releases

- A bioKepler VM executable on Amazon EC2, FutureGrid and SDSC Cloud
  - Builds upon CloudBioLinux including Bio-Linux and Galaxy

- A bioActor template that can be customized for different execution choices
  - e.g., local vs. Map/Reduce on a specific environment

- Example usecases
Demo and Questions

WorDS Director: Ilkay Altintas, Ph.D.
Email: altintas@sdsc.edu