SMRT® Link Troubleshooting Guide (v7.0.1)

Introduction

This document discusses common SMRT Link v7.0.1 issues and how to troubleshoot them.

SMRT Link is the web-based end-to-end workflow manager for the Sequel®/Sequel II Systems. It includes software applications for designing and monitoring sequencing runs, and analyzing and managing sequence data. The applications include:

- **Sample Setup:** Calculate binding and annealing reactions for preparing DNA samples for use on the Sequel/Sequel II System.
- **Run Design**: Design runs and create and/or import sample sheets which become available on the Sequel/ Sequel II System.
- **Run QC**: Monitor run progress, status and quality metrics.
- **Data Management**: Create Projects and Data Sets; manage access permissions for Projects and users; generate QC reports for Data Sets; view, import, export, or delete sequence, reference, and barcode files.
- **SMRT Analysis**: Perform multiple types of secondary analysis, including sequence alignment, variant detection, *de novo* assembly, structural variant calling, and RNA analysis.

Getting Support

Pacific Biosciences:

- http://www.pacb.com/support/software-downloads/
  PacBio web site includes SMRT Link software downloads, release notes, and documentation.
- http://www.pacb.com/products-and-services/analytical-software/devnet/
  PacBio Developer's Network web site includes tutorials and Data Sets.
- http://www.pacb.com/support/technical-support/
  PacBio Technical Support web site provides direct support for PacBio customers and Service Providers. Email **support@pacb.com** or call 1.877.920.PACB (7222). Current customers can submit an inquiry using our customer portal.

Third-Party Online Communities:

SeqAnswers.com (http://seqanswers.com/forums/forumdisplay.php?f=39) includes a PacBio forum that is actively monitored by the PacBio user community as well as some PacBio scientists and engineers.

Google Groups Mailing Lists:

- **SMRT_isoseq**: https://groups.google.com/forum/#!forum/smrt_isoseq
  Transcript isoform analysis with PacBio Iso-Seq® analysis and/or third-party tools.
- **SMRT_SV**: https://groups.google.com/forum/#!forum/smrt_sv
  Analysis of structural variants.
- **SMRT_kinetics**: https://groups.google.com/forum/#!forum/smrt_kinetics
  Base modification analysis.
- **SMRT_denovo**: https://groups.google.com/forum/#!forum/smrt_denovo
  De novo genome assembly.

General Troubleshooting

SMRT Link Log Files

Installation/Upgrade

These files are useful for investigating and troubleshooting issues that occur during installation or upgrade:

```
$SMRT_ROOT/userdata/log/installer/install.log
$SMRT_ROOT/userdata/log/installer/upgrade.log
$SMRT_ROOT/userdata/log/installer/reconfig.log
```

SMRT Link Services

This file is useful for viewing error messages during routine operations:

```
$SMRT_ROOT/userdata/log/smrtlink-analysisservices-gui/secondary-smrt-server.log
```

SMRT Link Java Console

These files are useful for investigating major system failures, such as memory overflow:

```
$SMRT_ROOT/current/bundles/smrtlink-analysisservices-gui/current/private/pacbio/smrtlink-
analysisservices-gui/tomcat_current/logs
```

WSO2

These files are useful for investigating issues with user rights and roles, such as authentication problems:

```
$SMRT_ROOT/userdata/log/smrtlink-analysisservices-gui/wso2/wso2carbon.log
$SMRT_ROOT/userdata/log/smrtlink-analysisservices-gui/wso2/wso2-apigw-errors.log
```

SMRT Analysis Job - Main Pipeline Logs

These files are useful for investigating any SMRT Analysis job failures; they are located in the `jobs_root` directory, for example `$SMRT_ROOT/userdata/jobs_root/004/004000/logs`.

```
└── logs
   ├── master.log
   └── pbsmrtpipe.log (Less verbose than master.log.)
```

SMRT Analysis Job - Task logs

These files are useful for investigating task-specific failures. Within the task directory is task-level output from individual task executions. Example: `$SMRT_ROOT/userdata/jobs_root/004/004000/tasks/pbcore-tools.tasks.filterdataset-0/`.

```
└── tasks
    └── pbalign.tasks.pbalign_ccs-1
        ├── cluster.stderr
        ├── cluster.stdout
        ├── stderr
        └── stdout
```

SMRT Analysis Job - Standard Error and OUT logs

Logs in the job top-level directory contain information returned from the shell or `pbscala`. These files are useful for investigating job failures at the SMRT Link level, for example database errors. The `*job.std*` files contain

STDERR/STDOUT from the main job execution script. Note that some of these files may be present for other job types, such as Data Set import or merging.

```
├── job.stderr
├── job.stdout
├── pbscala-job.stderr
└── pbscala-job.stdout
```

Sending Log Files to Technical Support

Troubleshooting information can be sent to Pacific Biosciences' Technical Support multiple ways. The following two methods **require** a connection to the PacBio Event Server and Update Server.

- From the SMRT Link menu: **About > Troubleshooting Information > Send**.
- From a SMRT Link "Failed" analysis Results page: Click **Send Log Files**.

If there is **no** connection to the PacBio Event Server, run the following command to generate a .tgz file and email the file to **support@pacb.com** to file a case:

- `$SMRT_ROOT/admin/bin/tsreport-install --bundle`

SMRT Link Configuration Files

smrtlink.config

This is the master configuration file generated by the installer; it has the following format:

```
# smrtlink config

# 'install' settings
install__user='secondarytest';
install__group='Domain Users';
install__sluuid='__USE_DEFAULT__';               # Current val: '1e401734-a193-4863-bbae-
2e1294e77401'

# 'system' settings
system__memtotal='__USE_DEFAULT__';              # Current val: '70866960384'

# 'smrtlink' settings
smrtlink__dnsname='smrtlink-alpha.nanofluidics.com';
smrtlink__gui_port='8080';
smrtlink__services_port='8081';
smrtlink__gui_minmem='__USE_DEFAULT__';          # Current val: '3392'
smrtlink__gui_maxmem='__USE_DEFAULT__';          # Current val: '3392'
smrtlink__services_minmem='__USE_DEFAULT__';     # Current val: '16896'
smrtlink__services_maxmem='__USE_DEFAULT__';     # Current val: '16896'
smrtlink__mail_host='__USE_DEFAULT__';           # Current val: ''
smrtlink__mail_port='__USE_DEFAULT__';           # Current val: '25'
smrtlink__mail_user='__USE_DEFAULT__';           # Current val: ''
smrtlink__mail_password='__USE_DEFAULT__';       # Current val: ''
smrtlink__extended_cell_use_enable='__USE_DEFAULT__';  # Current val: 'false'
```

To update the system configuration, PacBio recommends that you modify this file and then run `$SMRT_ROOT/admin/bin/smrt_reconfig` to generate `smrtlink.config`, `smrtlink-system-config.json` and `pbsmrtpipe_00.json.`

Note: If you edit any of the `smrtlink.config`, `smrtlink-system-config.json` or `pbsmrtpipe_00.json` files manually, they will be **overwritten** by the next update or `smrt_reconfig` call.

smrtlink-system-config.json

Use this file to view all the configurations used by SMRT Link services. As mentioned earlier, this is **automatically** generated and local edits may be overwritten by future updates. However it is more readable and useful for debugging services than the installer configuration file.

```
{
  "smrtflow": {
    "server": {
      "port": 9091,
      "manifestFile": "/opt/pacbio/smrtlink/install/smrtlink-release_7.0.1.66975/etc/pacbio-
manifest.json",
      "eventUrl": "https://smrtlink-eve.pacbcloud.com:8083",
      "dnsName": "smrtlink-release.nanofluidics.com",
      "bundleDir": "/opt/pacbio/smrtlink/install/smrtlink-release_7.0.1.66975/bundles/smrtlink-
analysisservices-gui/current/private/pacbio/smrtlink-analysisservices-gui/resources/pacbio-
bundles"
    },
    "engine": {
      "maxWorkers": 16,
      "jobRootDir": "/opt/pacbio/smrtlink/userdata/jobs_root",
      "pbsmrtpipePresets": [
          "/opt/pacbio/smrtlink/userdata/generated/config/computecfg_00/pbsmrtpipe_00.json"
      ]
    },
    "db": {
      "properties": {
        "databaseName": "smrtlinkdb",
        "user": "smrtlink_user",
        "password": "password",
        "portNumber": 9095,
        "serverName": "localhost"
      }
    }
  },
  "pacBioSystem": {
    "tomcatPort": 9090,
    "tomcatMemory": 1024,
    "smrtLinkServerMemoryMin": 16896,
    "smrtLinkServerMemoryMax": 16896,
    "smrtViewPort": 9094,
    "tmpDir": "/opt/pacbio/smrtlink/userdata/tmp_dir",
    "logDir": "/opt/pacbio/smrtlink/userdata/log/smrtlink-analysisservices-gui",
    "pgDataDir": "/opt/pacbio/smrtlink/userdata/db_datadir",
    "enableCellReuse": false,
    "remoteBundleUrl": "http://smrtlink-update.pacbcloud.com:8084",
    "smrtLinkSystemRoot": "/opt/pacbio/smrtlink",
    "smrtLinkSystemId": "2b4ba7ae-28a8-4a79-a255-c07cd34ea069",
    "mailHost": "smtp.nanofluidics.com",
    "mailPort": 25,
    "mailUser": null,
    "mailPassword": null
  },
  "comment": "Created: Tue Jul  9 09:55:08 PDT 2019"
}
```

pbsmrtpipe_00.json

Use this file to view the configuration presets for running `pbsmrtpipe`, or when a SMRT Analysis job fails. In SMRT Link v7.0, there is **no** option to configure multiple JMS submission commands - one JMS submission command may be selected from a drop-down list when setting up the SMRT Analysis job. As such, there may be multiple `pbsmrtpipe_##.json` files within `computecfg_##` directories, where the ## matches on each.

By default, `smrtlink-system-config.json` points to `computecfg_00` and therefore the `pbsmrtpipe_00.json` configuration file.

Example: `$SMRT_ROOT/smrtlink/userdata/generated/config/computecfg_00/pbsmrtpipe_00.json:`

```
{
   "pipelineId":   "installer.user.config.pbsmrtpipe_00",
   "presetId":     "installer.user.config.pbsmrtpipe_00",
   "name":         "SMRT Analysis Compute Configuration",
   "description":  "SMRT Analysis Compute Configuration",
   "options": {
     "pbsmrtpipe.options.debug_mode":           false,
     "pbsmrtpipe.options.max_nproc":            24,
     "pbsmrtpipe.options.tmp_dir":              "/opt/pacbio/smrtlink/userdata/tmp_dir",
     "pbsmrtpipe.options.chunk_mode":           true,
     "pbsmrtpipe.options.max_total_nproc":      1000,
     "pbsmrtpipe.options.max_nchunks":          24,
     "pbsmrtpipe.options.distributed_mode":     true,
     "pbsmrtpipe.options.exit_on_failure":      true,
     "pbsmrtpipe.options.cluster_manager":      "/opt/pacbio/smrtlink/userdata/generated/config/
computecfg_00",
     "pbsmrtpipe.options.max_nworkers":         100,
     "pbsmrtpipe.options.progress_status_url": ""
   },
   "taskOptions": {},
 "_comment": "Created by installprompter: Tue Jul  9 09:55:07 PDT 2019"
```

SMRT Link Administration

SMRT Link Recovery Installation

Use the following steps to recover a SMRT Link installation that is not working correctly or has suffered major corruption, such as files deleted, moved, and so on. The most recent database backup can be used to restore job history and LDAP settings to the new database.

1. Set `$SMRT_USER, $SMRT_ROOT`:
   ```
   SMRT_ROOT=/opt/pacbio/smrtlink
   SMRT_USER=smrtanalysis
   su - ${SMRT_USER}
   ```

2. Stop the running services. Terminate processes if necessary:
   ```
   ${SMRT_ROOT}/admin/bin/services-stop
   ```

3. Backup the existing SMRT Link Installation directory:
   ```
   mv ${SMRT_ROOT} ${SMRT_ROOT}.bak
   ```

   This directory can be removed once the new installation has been verified and failure analysis completed.

4. Install SMRT Link: This installation command **preserves** the existing configuration from the backed-up installation. Make sure to install the **same** version of SMRT Link.

   ```
   smrtlink_7.0.1.66975.run \
   --rootdir ${SMRT_ROOT} \
   --configfile ${SMRT_ROOT}.bak/userdata/config/smrtlink.config \
   --batch
   ```

5. Restore the SMRT Link Database from backup:
   ```
   ${SMRT_ROOT}/admin/bin/dbhelper --restore smrtlinkdb \
   --restore-file ${SMRT_ROOT}.bak/userdata/db_datadir/backups/upgrade/latest_smrtlinkdb.sql
   ```

6. Start the services:
   ```
   ${SMRT_ROOT}/admin/bin/services-start
   ```

SMRT Link Processes Sometimes Exceed Soft User Limits

Symptom: Under normal usage, SMRT Link can spawn processes exceeding the user resource limits. The resulting errors can take a number of forms, such as the following:

```
-bash: fork: retry: Resource temporarily unavailable
su - smrtanalysis : cannot set userid: Resource temporarily unavailable.
java.lang.OutOfMemoryError: unable to create new native thread
```

Diagnosis: A snapshot of SMRT Link thread usage can be determined by using:

```
SMRT_USER="smrtanalysis"
ps -Lf -u ${SMRT_USER} | wc -l
```

Compare this with the current soft resource limit settings for `max user processes` and `open files` respectively:

```
ulimit -u -n
```

Solution: The soft resource limit can be increased using:

```
ulimit -u 8192
ulimit -n 8192
```

We have found that a limit of 8192 for `max user processes` and `open files` seems to work in most cases.

```
$ ulimit -u -n
max user processes              (-u) 8192
open files                      (-n) 8192
```

Symptom: The hard resource limit, which is the upper bound of the range a user can set for itself, is sometimes set lower than what is recommended for SMRT Link, preventing the soft resource limit from being set. Following is the output from `ulimit`:

```
$ ulimit -n 8192
-bash: ulimit: open files: cannot modify limit: Operation not permitted
```

Diagnosis: In this case, the hard resource limit must be increased. Check the current limits:

```
ulimit -Hu -Hn
```

Solution: To change the hard resource limits, edit `/etc/security/limits.conf`, and add the following lines:

```
smrtanalysis - nproc 8192
smrtanalysis - nofile 8192
```

Log out of any user shells for the `$SMRT_USER` and log back in. Check the new `ulimit` settings using `ulimit -u -n`.

Stop, then restart SMRT Link services:

```
$SMRT_ROOT/admin/bin/services-stop
ps -ef | grep smrt
# kill any reported processes still running.
$SMRT_ROOT/admin/bin/services-start
```

Troubleshooting Analysis Services

Confirm that the services were started by running `services-start`:

```
$SMRT_ROOT/admin/bin/services-start

   Checking services state...
   Checking for port conflicts...
   Checking system...
     Checking mount points...

   Starting SMRT View Server...
     SMRT View Server started successfully..

   Starting SMRT Link Services...
     Running apply-config...
     Checking services status...
       Checking SMRT Link Analysis services is not running...
       Checking SMRT Link GUI webserver is not running...
       Checking WSO2 API Manager daemon is not running...
     Checking system limits...
     Checking ports...
     Starting database...
     Starting all services...
       Starting SMRT Link Analysis services...
         Waiting for SMRT Link Analysis services to start...
       Starting SMRT Link GUI webserver...
       Starting WSO2 API Manager...
     Connecting to WSO2 API Manager (may take a few minutes)...
     Configuring WSO2 API Manager...
       Creating roles...
       Setting user roles...
       Adding pbicsuser user...
         User 'pbicsuser' already exists
       Configuring SMRT Link APIs...
       Configuring API Manager admin roles...
       Configuring API Manager user roles...
     All SMRT Link Services started successfully.

   Waiting for services to start...

   Checking SMRT Link status...
   Checking SMRT View status...

   Status summary:
       SMRT Link status:  ok
       SMRT View status:  ok

   Importing canneddata and barcodes...

   Services started successfully.
```

The last line of output should indicate `Services started successfully`.

If logged into the host running SMRT Link, check that the services are up and running:

```
$SMRT_ROOT/admin/bin/services-status

Checking SMRT Link status...
Checking SMRT View status...

Status summary:
    SMRT Link status:  ok
    SMRT View status:  ok
```

To check that the services can be reached from the command line of the SMRT Link server:

```
curl -s http://localhost:9091/status | python2 -m json.tool
{
    "id": "smrtlink_analysis",
    "message": "Services have been up for 5 minutes and 30.517 seconds.",
    "uptime": 330517,
    "user": "smrtanalysis",
    "uuid": "7162bc10-afaf-4b36-9ccd-a3263352fb2d",
    "version": "1.2.0+66974.ead9565"
}
```

Alternately, a more detailed service status can be displayed using `pbservice` from the SMRT Link server command line:

```
$SMRT_ROOT/smrtcmds/bin/pbservice status --host localhost --port 9091
ID                UUID                                 Version             Message
smrtlink_analysis 7162bc10-afaf-4b36-9ccd-a3263352fb2d 1.2.0+66974.ead9565 Services have been up
for 9 minutes and 43.122 seconds.

Pbservice 1.2.0+ead9565 IS compatible with Server 1.2.0+66974.ead9565

DataSet Summary (active datasets) :

SubreadSets             : 3
HdfSubreadSets          : 0
ReferenceSets           : 4
GmapReferenceSets       : 0
BarcodeSets             : 7
AlignmentSets           : 4
ConsensusAlignmentSets  : 0
ConsensusReadSets       : 0
ContigSets              : 2
TranscriptSets          : 0
TranscriptAlignmentSets : 0

System Job Summary by job type:

Import DataSet                    : 15
Merge DataSet                     : 0
Analysis                          : 7
Convert Fasta to ReferenceSet     : 0
Convert Fasta to BarcodeSet       : 0
```

Listing ports opened for listening on SMRT Link host:

```
$ netstat -nplt | grep "java\|postgres"
tcp     0      0 0.0.0.0:5672        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:9611        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:9999        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:9711        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:61423       0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:8243        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 10.0.0.1:10711      0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:8280        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:7611        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 127.0.0.1:10397     0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:7711        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:8672        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:9090        0.0.0.0:*         LISTEN      45301/java
tcp     0      0 0.0.0.0:9443        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 0.0.0.0:9763        0.0.0.0:*         LISTEN      45507/java
tcp     0      0 127.0.0.1:9091      0.0.0.0:*         LISTEN      45066/java
tcp     0      0 0.0.0.0:9094        0.0.0.0:*         LISTEN      44930/java
tcp     0      0 0.0.0.0:11111       0.0.0.0:*         LISTEN      45507/java
tcp     0      0 127.0.0.1:9095      0.0.0.0:*         LISTEN      45058/postgres
```

Upgrade: Ensuring SMRT Analysis Services are Stopped

Sometimes, the `$SMRT_ROOT/admin/bin/services-stop` script may leave residual processes related to SMRT Link. Following are a few things you can check to ensure that this has not happened, and what to do if services remain.

1. Always ensure that you have attempted to stop the services using the normal script. (You may safely run the script repeatedly.)

   ```
   $SMRT_ROOT/admin/bin/services-stop
   ```

2. Check to see if any SMRT Link related processes remain:

   ```
   ps -ef | grep "smrt"
   ```

If any SMRT Link processes remain, the command will return more than just the `grep` command. With all processes successfully stopped, the output will **only** include the `grep` command.

```
$ ps -ef | grep "smrt"

smrtana+  77970      1  0 Aug09 ?        00:00:28 /home/smrtanalysis/v5-test/smrtlink/install/
smrtlink-release_5.0.0.6792/bundles/smrtlink-analysisservices-gui/install/smrtlink-analysisservices-
gui-release_5.0.0.6752/private/thirdparty/postgresql/postgresql_9.6.1/bin/postgres -D /home/
smrtanalysis/v5-test/smrtlink/userdata/db_datadir/dbstore -p 55560 -h localhost

smrtana+  78251      1  0 Aug09 ?        00:00:00 sh /home/smrtanalysis/v5-test/smrtlink/install/
smrtlink-release_5.0.0.6792/bundles/smrtlink-analysisservices-gui/install/smrtlink-analysisservices-
gui-release_5.0.0.6752/private/pacbio/smrtlink-analysisservices-gui/wso2am-2.0.0/bin/wso2server.sh

smrtana+ 123086  77970  0 09:08 ?        00:00:00 postgres: smrtlink_user smrtlinkdb 127.0.0.1(2138)
idle

smrtana+ 123088  77970  0 09:08 ?        00:00:00 postgres: smrtlink_user smrtlinkdb 127.0.0.1(2139)
idle

smrtana+ 123090  77970  0 09:08 ?        00:00:00 postgres: smrtlink_user smrtlinkdb 127.0.0.1(2140)
idle

smrtana+ 123349 123317  0 09:18 pts/0    00:00:00 grep smrt
```

If you receive any output besides your `grep` process, first attempt to shut down the processes cleanly with Signal 15. This can be done with either `pkill` or `kill`. (The `-15` is the Signal 15, which will attempt to shut the process down cleanly. The `-f` in both `pkill` and `pgrep` ensures that the entire command-line is matched against the string `smrt.`)

`pkill -15 -f "smrt"` or `kill -15 $(pgrep -f "smrt")`

Check again to see if any SMRT Link processes remain:

```
ps -ef  | grep "smrt"
```

If any processes are returned (besides the `grep` process again), repeat the above step. If processes still remain, send a Signal 9 to terminate those processes:

`pkill -9 -f “smrt”` or `kill 9 $(pgrep -f "smrt")`

Check again to see if any SMRT Link processes remain:

```
ps -ef  | grep "smrt"
```

If you still see SMRT Link processes from the output, those processes may be hung. At this point the best course of action is a reboot.

Please be aware that using `grep` to search for the string `smrt` will result in all processes returned if the chosen user name contains the string `smrt`, such as the recommended `smrtanalysis`. The example below accounts for that particular case, so adjust as necessary. Though, the `pgrep` and `pkill` examples do **not** match against username.

```
ps -ef | grep -E "^smrt.*smrt"
```

Configuration: Advanced Distributed Computing Configuration

PacBio supports distributed job submission in SMRT Link using the following job schedulers:

- GridEngine: SGE, OGE, UGE (**Note**: GridEngine variants are for the most part interoperable.)
- PBS Pro, PBS, Torque
- LSF, OpenLava
- SLURM

Configuring Job Distribution

Job distribution is configured interactively at installation. Configuration files are generated at installation and reconfiguration, and in most cases do not need additional modification.

> To review and/or modify configuration settings for SMRT Link job distribution at any time (this will stop the SMRT Link services and cause any actively running jobs to fail):
>
> ```
> $SMRT_ROOT/admin/bin/smrt_reconfig
> ```

New configuration settings will be automatically applied following a restart of SMRT Link Services:

```
    $SMRT_ROOT/admin/bin/services-stop
    $SMRT_ROOT/admin/bin/services-start
```

Switching Distributed Computing On or Off in SMRT Link

The recommended way to enable/disable distributed computing is to run `$SMRT_ROOT/admin/bin/smrt_reconfig` and select `NONE` for the `jms_type`.

Checking JMS Configuration

Use the following command to see the currently-configured job submission command:

```
    $SMRT_ROOT/admin/bin/runjmscmd --start --test
```

Output:

```
noexec: LOGNAME='smrtanalysis' PATH='/usr/bin:/usr/bin:/bin' SGE_CELL='default' SGE_ROOT='/usr/share/gridengine' \
TERM='linux' USER='smrtanalysis' /usr/bin/qsub -S /bin/bash -sync y -V -q default -N J112233.runjmscmd_test_jobname \
-o /path/to/stdout.log -e /path/to/stderr.log -pe smp 1 'tmpcmd.sh --arg1 --arg2 "string arg with spaces"'
```

Customizable JMS Configuration File

To specify resource request lists for `mem_free` and `h_rt` to SGE `qsub` (for example, to define `qsub` options in the variable `SGE_STARTARGS`), add the following line to `$SMRT_ROOT/smrtlink/userdata/user_jmsenv/user.jmsenv.ish`:

```
    export SGE_STARTARGS="-l mem_free=2G, h_rt=120:0:0"
```

To make changes to other JMS environment variables:

```
JMS_TYPE="SGE"
export SGE_ROOT="/opt/gridengine";
export SGE_CELL="default";
QUEUE="high_mem";
PE="mpi";
```

Restarting SMRT Link Services is **not** needed.

Generated JMS Configuration Files

The following configuration files are generated by the installer. Do **not** edit them as they will **not** persist between upgrades. Customizations should be added in `$SMRT_ROOT/smrtlink/userdata/user_jmsenv/user.jmsenv.ish`.

This file contains JMS environment variables:

```
$SMRT_ROOT/smrtlink/userdata/generated/config/computecfg_00/jmsenv_00.ish
```

This file is a command-line template for submitting jobs:

```
$SMRT_ROOT/smrtlink/userdata/generated/config/computecfg_00/start_00.tmpl
```

This file is a command-line template for stopping jobs:

```
$SMRT_ROOT/smrtlink/userdata/generated/config/computecfg_00/stop_00.tmpl
```

SLURM - Revert to Using salloc/srun Instead of sbatch

As of SMRT Link v7.0.1 and later, the default is to use `sbatch` instead of `salloc/srun`. The last few releases of SLURM included a `--wait` switch for `sbatch,` which makes it the blocking call it is expected to be. But if there is some reason to use `salloc/srun` instead, set the following variables in the `user.jmsenv.ish` file:

```
SLURM_USE_SALLOCSRUN=true
SLURM_PRESTART_ARGS=""
SLRUM_START_ARGS=""
```

The `SLURM_PRESTART_ARGS` and `SLURM_START_ARGS` variables are included for precautionary reasons. If there are arguments specified in either **not** actually intended for `salloc` or `srun`, respectively, then those arguments should be cleared out to avoid conflict. If legitimate `salloc` or `srun` arguments are needed, then specify those values rather than setting zero-length strings.

Moving the SMRT Link Jobs Directory to a New Location

This procedure is for moving the SMRT Link jobs directory from the default location to an alternate location.

- Make sure **all** currently-running analyses are completed before performing the following steps.
- Perform the steps while logged in as the `$SMRT_USER`.

1. Stop SMRT Link Services:
   ```
   ${SMRT_ROOT}/admin/bin/services-stop
   ```

2. Move SMRT Link jobs to the new file system location:
   ```
   NEW_JOBS_ROOT=/path/to/new/jobs_root/
   mkdir -p ${NEW_JOBS_ROOT}
   mv ${SMRT_ROOT}/userdata/jobs_root.default/* ${NEW_JOBS_ROOT}
   ```

3. Redirect the symlink to the new location:

```
ln -sfn ${NEW_JOBS_ROOT} ${SMRT_ROOT}/userdata/jobs_root
```

4. Start SMRT Link Services:

```
${SMRT_ROOT}/admin/bin/services-start
```

Run QC

Recovering from a Failed Data Set Import

Symptom: Automatic import or multiple manual import attempts failed - the Data Sets **cannot** be seen in Data Management or Run QC. The failed import job was deleted by the user.

Diagnosis:

1. Restart SMRT Link Services and verify that Data Sets cannot be seen in Data Management or Run QC despite multiple services-start attempts.
2. Run `pbservice get-dataset` and inspect the job status. The job summary field should show `INACTIVE/DELETED`.

```
 *DATASET SUMMARY*
              id: 40
            uuid: ba28e17f-f6e0-4dab-9bb7-980add05f04f
            name: HG04217_LIB4_30pM-Cell4
      numRecords: 565322
     totalLength: 4960024160
           jobId: 39
             md5: fa9ab0cb4992e7abd510a8f8afc5a0d3
       createdAt: 2017-09-15T21:48:19.762Z
       updatedAt: 2017-09-21T23:16:04.012Z
            tags: subreadset
            path: /igm/runs/IGM_Seq03/sequel/r54201_20170914_145022/4_D01/
m54201_170915_064139.subreadset.xml
        (INACTIVE/DELETED)

    *JOB SUMMARY* (INACTIVE/DELETED)
                id: 39
              uuid: f04981e4-f8c9-4298-bef3-1f712a201e89
              name: Job import-dataset
             state: SUCCESSFUL
        project id: 1
         jobTypeId: import-dataset
         is active: false
         createdAt: 2017-09-15T21:44:30.001Z
         updatedAt: 2017-09-20T20:59:47.339Z
          run time: 429318 sec
        SL version: 5.0.1.9585
        created by: admin
           comment: Importing DataSet
              path: /igm/projects1/smrtlink_userdata/jobs_root/000/000039
```

Solution:

1. Run the following `dataset create` command on the `subreadset.xml` file.
   Note that this will update the `UniqueId` for `MetaType="PacBio.DataSet.SubreadSet"` in the `new.subreadset.xml` file that is generated. All other aspects will remain identical.

   ```
   $SMRT_ROOT/smrtcmds/bin/dataset create new.subreadset.xml /path/to/orig.subreadset.xml
   ```

2. Import using `$SMRT_ROOT/smrtcmds/bin/pbservice import-dataset`.

   Usage:
   ```
   pbservice import-dataset --host localhost --port 9091 --debug /path/to/dataset.xml
   ```

Run QC Empty for a Run

Symptom: Run QC metrics are empty and no plots can be viewed.

Diagnosis: Go to the SMRT Link Data Management page and verify that a Data Set for that run does **not** exist.

Solution:

1. Manually import the data:

- Graphical User Interface option: Go to **SMRT Link Data Management > View or Import Sequence Data > Import > Sequel Sequence Data**.

- Command-line option:

```
$SMRT_ROOT/smrtcmds/bin/pbservice import-dataset --host localhost --port 9091 \
--debug /path/to/dataset.xml
$SMRT_ROOT/smrtcmds/bin/pbservice import-dataset --host localhost --port 9091 \
--debug /opt/smrtlink/ userdata/data_root/r54001_000101_000000  (import multiple cells)
```

2. Verify that the import was successful:

- Graphical User Interface option: Go to the SMRT Link Run QC page and verify that metrics are no longer empty for the run of interest.

- Command-line option:

```
curl http://localhost:9091/smrt-link/datasets/subreads?showAll=true
```

Root Cause: Run QC can be empty if a prior import attempt failed due to any of the following reasons:

- Intermittent network connectivity failure
- Intermittent `smrtlink` services down

Data Management

Relative Path Dataset XML Quick Fix

Use the following workaround for relative-path dataset XMLs which cause analysis application errors such as the following from CCS2:

```
terminate called after throwing an instance of 'std::runtime_error'
what(): could not open BAM file: .../00002/tasks/pbccs.tasks.ccs-0/
m54001_160218_185027.subreads.bam
```

Compare the `ResourceId` attribute in the following examples.

Relative-Path Dataset XML:

```
<pbbase:ExternalResource
    Description="Points to the subreads bam file."
    MetaType="PacBio.SubreadFile.SubreadBamFile"
    Name="subreads bam"
    ResourceId="m54001_160322_221501.subreads.bam"
    TimeStampedName="pacbio_subreadfile_subreadbamfile-160323_03424927"
    UniqueId="53d3ff91-0ff4-4250-8850-2011e2d22031"
    Version="7.0.0">
```

Absolute-Path Dataset XML:

```
<pbbase:ExternalResource
    Description="Points to the subreads bam file."
    MetaType="PacBio.SubreadFile.SubreadBamFile"
    Name="subreads bam"
    ResourceId="/data/runs/54001/r54001_20160322_220424/1_A01/m54001_160322_221501.subreads.bam"
    TimeStampedName="pacbio_subreadfile_subreadbamfile-160323_03424927"
    UniqueId="53d3ff91-0ff4-4250-8850-2011e2d22031"
    Version="7.0.0">
```

Relative-Path Dataset XML Quick Fix commands:

This command will convert relative-path dataset XMLs to absolute-path dataset XMLs:

```
$SMRT_ROOT/smrtcmds/bin/dataset create --relative new.subreadset.xml orig.subreadset.xml
```

Subread Data Set from Instrument is not Displayed in SMRT Link After Import

Symptom: Subread Data Sets (the `subreadset.xml` file) are **not** visible in the SMRT Link graphical user interface following import.

Diagnosis: Rerun the import:

```
$SMRT_ROOT/current/bundles/smrttools/smrtcmds/bin/pbservice import-dataset --host localhost
--port 9091 --debug /path/to/new.subreadset.xml
```

A message indicating successful import, or reporting that the Data Set is already found should be displayed.

Next, check that the web services return the Data Set by type - this is how the SMRT Link graphical user interface gets its list:

```
curl http://localhost:9091/smrt-link/datasets/subreads
```

If SMRT Link does not display the Data Set, then try the query by UUID (found in the SubreadSet XML `UniqueId` attribute.)

```
curl http://localhost:9091/smrt-link/datasets/<UniqueId>
```

If the Data Set can be found by UUID, but **not** by type `subreads`, then confirm the existence of a line defining the run name in the `subreadset.xml` file:

```
grep 'pbmeta:Name' /path/to/subreadset.xml
```

If this line does **not** exist, then follow the steps below to edit the dataset XML.

Solution: If the dataset XML is not importing correctly and the `<pbmeta:Name></pbmeta:Name>` tag is missing from the file, backup and edit the file, then follow these steps to edit and reimport the dataset XML.

Backup the `subreadset.xml` file:

```
cp /path/to/subreadset.xml /path/to/subreadset.xml.bak
```

Make the following edits to `/path/to/subreadset.xml`:

1. Change the dataset XML UUID using the PacBio-provided `dataset` tool:
   ```
   $SMRT_ROOT/smrtcmds/bin/dataset newuuid /path/to/subreadset.xml
   ```
2. Add a `<pbmeta:Name></pbmeta:Name>` tag within `<pbmeta:RunDetails></pbmeta:RunDetails>`.

   Example:
   ```
   <pbmeta:RunDetails>
           <pbmeta:Name>RUN_NAME</pbmeta:Name>
           <pbmeta:TimeStampedName>r54099_20160304_202916</pbmeta:TimeStampedName>
           <pbmeta:CreatedBy>String</pbmeta:CreatedBy>
           <pbmeta:WhenCreated>0001-01-01T00:00:00</pbmeta:WhenCreated>
           <pbmeta:WhenStarted>2016-03-04T20:35:18.118914Z</pbmeta:WhenStarted>
   </pbmeta:RunDetails>
   ```

The dataset XML can then be imported:

```
$SMRT_ROOT/current/bundles/smrttools/smrtcmds/bin/pbservice import-dataset --host localhost
--port 9091 --debug /path/to/subreadset.xml
```

SMRT Analysis

HGAP Fails with Insufficient Memory Allocation Error

Diagnosis: HGAP fails on block size errors on the `daligner` step. The `analysis.log` file may contain one of the following errors:

- `"Error: Insufficient memory allocation (16.0Gb, reduce block size or increase allocation" in the analysis.log`
- `Warning: Block size too big, index occupies more than 1/4 of desired memory allocation (16.0Gb)`

Solution: Based on the genome size and complexity, this may involve fine tuning the `falcon` configuration override parameters.

Try reducing the block size (`-s`) option by pasting the following options for `falcon` configuration override in SMRT Link's HGAP 4 **Advanced Analysis Parameters** dialog:

```
pa_DBsplit_option = -x500 -s100; ovlp_DBsplit_option = -x500 -s100
```

Note that the defaults values in HGAP are `-x500 -s200`.

Iso-Seq Reads Flagged as 3'--3' Reads in Classification Step

Symptoms:

- The majority of Iso-Seq reads are failing in the classification step.
- Lower number of transcripts from `Isoseq3`.
- The total number of CCS reads look good, but the resulting number of reads with 5' and 3' primer is much lower.

Diagnosis:

1. View the `lima` summary for unusually high number of undesired 3p–3p artifacts. Example:

```
Lima Full length transcript report for Cell 2:

ZMWs input (A) : 349047
ZMWs above all thresholds (B) : 10708 (3%)
ZMWs below any threshold (C) : 338339 (97%)

ZMW marginals for (C):
Below min length : 1143 (0%)
Below min score : 0 (0%)
Below min end score : 60490 (18%)
Below min passes : 191 (0%)
Below min score lead : 0 (0%)
Below min ref span : 33932 (10%)
Without adapter : 191 (0%)
Undesired 5p--5p pairs : 6200 (2%)
Undesired 3p--3p pairs : 322001 (95%)
Undesired no hit : 191 (0%)

ZMWs for (B):
With different barcodes : 10708 (100%)
Coefficient of correlation : 0%

ZMWs for (A):
Allow diff barcode pair : 348856 (100%)
Allow same barcode pair : 348856 (100%)
```

```
Reads for (B):
Above length : 10708 (100%)
Below length : 0 (0%)
```

2. Inspect the CCS reads. Visually examine the CCS reads for the presence of a 3' ATGGGG overhang of the 5' primer. The CCS reads from a proper Iso-Seq library should contain 5' primer→ATGGGG overhang→ cDNA sequence→polyA tail→3' primer. You would observe that the 5-6 bp ATGGGG overhang is missing for the 3' end of the 5' primer sequence:

   AAGCAGTGGTATCAACGCAGAGTAC**ATGGG**

   Instead, polyA/T tracks on both sides of a “transcript" will be found, indicative of random priming.

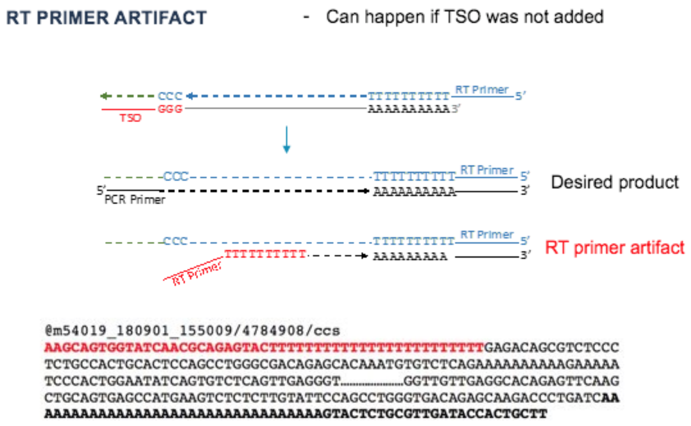
   The Iso-Seq primers are:

   ```
   >primer_5p
   AAGCAGTGGTATCAACGCAGAGTACATGGGG

   >primer_3p
   AAGCAGTGGTATCAACGCAGAGTAC

   >primer_3p (OR one can use rev_comp of 5p)
   GTACTCTGCGTTGATACCACTGCTT
   ```

   5' primer is expected to have the overhang ATGGGG and if the CCS reads are missing the ATGGGG overhang, it is an indication of problems in sample preparation. We suspect that this could happen due to a degraded Template Switching Oligonucleotides (TSO) batch or lack of TSO.



```
@m54019_180901_155009/4784908/ccs
AAGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTGAGACAGCGTCTCCC
TCTGCCACTGCACTCCAGCCTGGGCGACAGAGCACAAATGTGTCTCAGAAAAAAAAAAGAAAAA
TCCCACTGGAATATCAGTGTCTCAGTTGAGGT..................GGTTGTTGAGGCACAGAGTTCAAG
CTGCAGTGAGCCATGAAGTCTCTCTTGTATTCCAGCCTGGGTGACAGAGCAAGACCCTGATCAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGTACTCTGCGTTGATACCACTGCTT
```

   The 5’ and 3’ primers are reverse-complement and ATGGGG is the only distinguishing feature between the 5' and 3' primers, without which the reads are flagged as 3’—3’ by `lima`. Earlier versions of Iso-Seq, which were not as strict, will pass these, but the current Iso-Seq will **fail** them.

Solution/Workaround

We suspect that this could happen due to a degraded Template Switching Oligonucleotides (TSO) batch or lack of TSO. Please contact your local FAS or call PacBio Technical Support.

- Prepare the Iso-Seq library using fresh TSO or a new kit.
- See pages 4-5 of the document “Procedure & Checklist - Iso-Seq Template Preparation for Sequel Systems".

For Research Use Only. Not for use in diagnostic procedures. © Copyright 2018 - 2019, Pacific Biosciences of California, Inc. All rights reserved. Information in this document is subject to change without notice. Pacific Biosciences assumes no responsibility for any errors or omissions in this document. Certain notices, terms, conditions and/or use restrictions may pertain to your use of Pacific Biosciences products and/or third party products. Please refer to the applicable Pacific Biosciences Terms and Conditions of Sale and to the applicable license terms at http://www.pacb.com/legal-and-trademarks/product-license-and-use-restrictions/.

Pacific Biosciences, the Pacific Biosciences logo, PacBio, SMRT, SMRTbell, Iso-Seq and Sequel are trademarks of Pacific Biosciences. BluePippin and SageELF are trademarks of Sage Science, Inc. NGS-go and NGSengine are trademarks of GenDx. FEMTO Pulse and Fragment Analyzer are trademarks of Agilent Technologies Inc. All other trademarks are the sole property of their respective owners.

P/N 101-397-000 Version 03 (August 2019)