SMRT® Link Troubleshooting Guide (v8.0)

Introduction

This document discusses common SMRT Link v8.0 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
```

Cromwell

These files are useful for investigating issues with the Cromwell pipeline engine:

```
$SMRT_ROOT/userdata/log/cromwell/cromwell.stdout
$SMRT_ROOT/userdata/log/cromwell/cromwell.stderr
```

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, logs for job ID `1234567` would be found in the directory `$SMRT_ROOT/userdata/jobs_root/0001/0001234/0001234567/logs`.

```
└── logs
  ├── workflow.bea64a7f-29c9-4ba3-bb8e-335e4d861466.log
  └── sl_align_subreads/ (Points to pbcromwell workflow logs)
```

SMRT Analysis Job - Task logs

These files are useful for investigating task-specific failures. Within the `cromwell-job` directory is task-level output from individual task executions.

Example: Below is a simple example of a `pbcromwell` pipeline task, relative to the imaginary job directory `$SMRT_ROOT/userdata/jobs_root/0001/0001234/0001234567/cromwell-job`.

```
└── cromwell-job
    └── call-dataset_filter
        ├── execution
        │   ├── script
        │   ├── script.submit
        │   ├── stderr
        │   ├── stderr.submit
        │   ├── stdout
        │   └── stdout.submit
        └── inputs
            └── 1234567890
                └── 0dd95f6e-0a50-4a06-8a56-4b921460dc95.subreadset.xml
```

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, such as database errors. The `pbscala-job.std*` files contain STDERR/STDOUT from the main job execution script.

Note: Some of these files may be present for other job types, such as Data Set import or merging.

```
├── 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
```

The full file name of the tarball will be printed to `stdout`, but the most recently generated file will also be linked to from the symbolic link `$SMRT_ROOT/userdata/tsreport/data/ts-install.tgz`.

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 run `$SMRT_ROOT/admin/bin/smrt_reconfig` to rerun the configuration prompts and generate `smrtlink.config`, `smrtlink-system-config.json`, `cromwell.conf` and `cromwell_00.json`. When running `smrt_reconfig`, the existing settings display as defaults. To reset to the factory defaults, enter the special value `_USE_DEFAULT_` at any prompt.

Note: PacBio advises **against** editing any of the `smrtlink.config`, `smrtlink-system-config.json`, `cromwell.conf` or `cromwell_00.json` files manually, as 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, you may find that this file is more readable and useful for debugging services than the installer configuration file.

```
$ cat /opt/pacbio/smrtlink/userdata/generated/config/smrtlink-system-config.json
{
  "smrtflow": {
    "server": {
      "port": 9091,
      "manifestFile": "/opt/pacbio/smrtlink/smrtlink/install/smrtlink-release_8.0.0.80529/etc/
pacbio-manifest.json",
      "eventUrl": "https://smrtlink-eve.pacbcloud.com:8083",
      "dnsName": "vm-ts-node01.nanofluidics.com",
      "bundleDir": "/opt/pacbio/smrtlink/smrtlink/install/smrtlink-release_8.0.0.80529/bundles/
smrtlink-analysisservices-gui/current/private/pacbio/smrtlink-analysisservices-gui/resources/
pacbio-bundles"
    },
    "engine": {
      "maxWorkers": 8,
      "jobRootDir": "/opt/pacbio/smrtlink/smrtlink/userdata/jobs_root",
      "pbsmrtpipePresets": [
          "/opt/pacbio/smrtlink/smrtlink/userdata/generated/config/computecfg_00/
cromwell_00.json”
      ]
    },
    "db": {
      "properties": {
        "databaseName": "smrtlinkdb",
        "user": "smrtlink_user",
        "password": "password",
        "portNumber": 9095,
        "serverName": "localhost"
      }
    },
    "cromwell": {
      "host": "localhost",
      "port": 9096
    }
  },
  "pacBioSystem": {
    "tomcatPort": 9090,
    "tomcatMemory": 1024,
    "smrtLinkServerMemoryMin": 4608,
    "smrtLinkServerMemoryMax": 4608,
    "tmpDir": "/opt/pacbio/smrtlink/smrtlink/userdata/tmp_dir",
    "logDir": "/opt/pacbio/smrtlink/smrtlink/userdata/log/smrtlink-analysisservices-gui",
    "pgDataDir": "/localdisk/scratch/smrtlink/userdata/db_datadir",
    "enableCellReuse": false,
    "remoteBundleUrl": "http://smrtlink-update.pacbcloud.com:8084",
    "smrtLinkSystemRoot": "/localdisk/scratch/smrtlink",
    "smrtLinkSystemId": "2718d61c-d9ea-428d-8ae2-f348c0e7e1f3",
    "mailHost": null,
    "mailPort": 25,
    "mailUser": null,
    "mailPassword": null
  },
  "comment": "Created: Tue Oct 15 15:28:11 PDT 2019"
}
```

cromwell_00.json

Use this file to view the configuration presets for running `pbcromwell`, or when a SMRT Analysis job fails. SMRT Link v8.0 includes the 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 `cromwell_##.json` files within `computecfg_##` directories, where the `##` matches each configuration file.

Note: Do **not** edit this file manually.

By default, `smrtlink-system-config.json` points to `computecfg_00` and therefore the `cromwell_00.json` configuration file.

Example: `$SMRT_ROOT/smrtlink/userdata/generated/config/computecfg_00/cromwell_00.json`:

```
{
    "pipelineId":   "installer.user.config.cromwell_00",
    "presetId":     "installer.user.config.cromwell_00",
    "name":         "SGE",
    "description":  "SGE",
    "options": {
        "cromwell.workflow_options.nproc":         7,
        "cromwell.workflow_options.max_nchunks":   88,
        "cromwell.workflow_options.log_level":     "INFO",
        "cromwell.engine_options.backend":         "computecfg_00",
        "cromwell.engine_options.read_from_cache": false,
        "cromwell.engine_options.write_to_cache":  true
    },
    "taskOptions": {},
    "_comment": "Created by installprompter: Thu Oct 10 10:50:42 PDT 2019"
}
```

cromwell.conf

This is the configuration file for the `Cromwell` pipeline engine. Besides configuring various limitations and other `pbcromwell` settings, it also describes how the JMS submission, deletion, and status commands should be called. **Note**: Do **not** edit this file manually.

As seen below, SMRT Link calls the `runjmscmd` script for all three JMS commands. The `runjmscmd` script, in turn, will source the relevant `jmsenv_##.ish` configuration file, where the variable values are then used to generate these commands.

Note: The `start.tmpl` and `stop.tmpl` files are **no longer** used in SMRT Link v8.0 and later.

Example: `$SMRT_ROOT/smrtlink/userdata/generated/config/cromwell.conf`**:**

```
include required(classpath("application"))
system {
  max-concurrent-workflows = 32
}
webservice {
  port = 9096
  interface = localhost
}
call-caching {
  enabled = true
    invalidate-bad-cache-results = true
}
workflow-options {
  workflow-log-temporary = false
}

database {
  profile = "slick.jdbc.PostgresProfile$"
  db {
    driver = "org.postgresql.Driver"
    url = "jdbc:postgresql://localhost:9095/cromwell"
    user = "smrtlink_user"
    password = "password"
    port = 9095
    connectionTimeout = 5000
  }
}

backend {
  # Always keep the default at "Local" and specify any other backend
  # via the computecfg model, in the cromwell_NN.json preset json file.
  default = "Local"
  providers {
    Local {
      actor-factory = "cromwell.backend.impl.sfs.config.ConfigBackendLifecycleActorFactory"
      config {
          concurrent-job-limit = 10
        filesystems {
          local {
            caching {
              duplication-strategy: [
              "soft-link"
              ]
            }
            localization: [
              "soft-link",
            ]
          }
        }
      }
    }
    computecfg_00 {
      actor-factory = "cromwell.backend.impl.sfs.config.ConfigBackendLifecycleActorFactory"
      config {
        concurrent-job-limit = 500
        exit-code-timeout-seconds = 1800
        script-epilogue = ""

        # Define runtime attributes, settable on a per-workflow or
        # per-task basis
        runtime-attributes = """
            Int cpu = 1
            Float? memory_gb
            Float? mempercpu_gb
        """

        submit = """
        /opt/pacbio/smrtlink/install/smrtlink-release_8.0.0.80529/admin/bin/runjmscmd \
```

```
            --start \
            --computecfg-id "computecfg_00" \
            --computecfg-name "SMRT Analysis Compute Configuration" \
            --computecfg-description "SMRT Analysis Compute Configuration" \
            --jobname ${job_name} \
            --stdoutfile "${out}" \
            --stderrfile "${err}" \
            --nproc "${cpu}" \
            --cmd "${script}"
        """

        # command for killing/aborting
        kill = """
        /opt/pacbio/smrtlink/install/smrtlink-release_8.0.0.80529/admin/bin/runjmscmd \
            --stop \
            --computecfg-id "computecfg_00" \
            --computecfg-name "SMRT Analysis Compute Configuration" \
            --computecfg-description "SMRT Analysis Compute Configuration" \
            --jobid ${job_id}
        """

        # Command used at restart to check if a job is alive
        check-alive = """
        /opt/pacbio/smrtlink/install/smrtlink-release_8.0.0.80529/admin/bin/runjmscmd \
            --status \
            --computecfg-id "computecfg_00" \
            --computecfg-name "SMRT Analysis Compute Configuration" \
            --computecfg-description "SMRT Analysis Compute Configuration" \
            --jobid ${job_id}
        """

        # How to search the submit output for a job_id
        job-id-regex = "(\\d+)\\.\\w+"

        filesystems {
          local {
            localization: [
             "soft-link","hard-link","copy"
            ]
            caching {
              duplication-strategy: [
                "soft-link", "hard-link", "copy"
              ]
              caching-strategy: "file"
            }
          }
        }
      }
    }
  }
}
```

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} ${SMRT_ROOT}.bak
   ```
4. If the `db_datadir` symbolic link points outside of `$SMRT_ROOT`, also backup the existing `db_datadir` destination:
   ```
   mv "$(realpath -e ${SMRT_ROOT}/userdata/db_datadir)"{,.bak}
   ```

   Note: The `.bak` directories may optionally be removed once the new installation has been verified and failure analysis completed.
5. 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_8.0.0.80529.run \
   --rootdir ${SMRT_ROOT} \
   --configfile ${SMRT_ROOT}.bak/userdata/config/smrtlink.config \
   --batch
   ```
6. Restore the SMRT Link Database from a previous backup:
   ```
   $SMRT_ROOT/admin/bin/dbhelper --restore smrtlinkdb --restore-file /opt/pacbio/smrtlink/
   userdata/db_datadir/backups/upgrade/latest_smrtlinkdb.sql
   ```

   Note: Do **not** restore the `wso2am` or `cromwell` databases.
7. Start the services:
   ```
   ${SMRT_ROOT}/admin/bin/services-start
   ```

SMRT Link Database Backup

- To backup the SMRT Link database, run `$SMRT_ROOT/admin/bin/dbhelper –backup.`
- To backup regularly, set up a `cron` job to run the above command on a weekly basis. (This is **highly Recommended**.)

Note: SMRT Link v8.0 **no longer** automatically performs a weekly database backup.

Restoring SSL Certificates after Recovery Installation

1. Determine the fully qualified domain name (FQDN) of the SMRT Link installation. If different from the previous installation, make sure that the FQDN is listed on the SSL certificate or it will not be valid. Optionally set it to a variable.
   ```
   FQDN="$(hostname --fqdn)"
   ```

2. Locate the Java Keystore from the previous installation and either copy it to a convenient location, or set the current location to a variable for later use.
```
KEYSTORE="${SMRT_ROOT}.bak/current/bundles/smrtlink-analysisservies-gui/current/
private/pacbio/smrtlink-analysisservices-gui/wso2am-2.0.0/repository/resources/
security/smrtlink_example_com.jks"
```

3. Set the passphrase to a variable by parsing WSO2's `carbon.xml` configuration, as shown below.
```
KEYPW="$(grep KeyPassword ${SMRT_ROOT}.bak/current/bundles/smrtlink-analysisservices-
gui/current/private/pacbio/smrtlink-analysisservices-gui/wso2am-2.0.0/repository/conf/
carbon.xml | sed -e 's|^[[:space:]]*||' -e 's|.*$||')"
```

4. Stop the SMRT Link services if they are still running. (Make sure there are no active SMRT Link jobs, or they will need to be rerun.)
```
$SMRT_ROOT/admin/bin/services-stop
```

5. Rerun the `install_ssl_cert.sh` script in the new SMRT Link installation, feeding it the information from the steps above as arguments. The arguments **must** appear in the order shown.
```
$SMRT_ROOT/admin/bin/install_ssl_cert.sh $FQDN $KEYSTORE $TRUSTSTORE $KEYPW
```

6. Restart the SMRT Link services to have it use the restored certificate.
```
$SMRT_ROOT/admin/bin/services-start
```

7. Once you have determined that the custom SSL certificate is working as expected, unset the variables (or at least the KEYPW, to be safe.)
```
unset FQDN KEYSTORE TRUSTSTORE KEYPW
```

Note: Only the `$SMRT_USER` of the previous installation will have the permissions necessary to access the files containing the pass phrase.

Restoring LDAP Settings prior to Recovery Installation

The settings for the LDAP configuration can be found in the file below. Only the `$SMRT_USER` will have the permissions to access this file.

```
${SMRT_ROOT}.bak/current/bundles/smrtlink-analysisservices-gui/current/private/pacbio/
smrtlink-analysisservices-gui/wso2am-2.0.0/repository/deployment/server/userstores/
smrtlink_example_com.xml
```

Note: This file does **not** provide the password for the BIND DistinguishedName account required for WSO2 to synchronize with the LDAP server.

Importing an OpenSSL Formatted Key and Certificate into the Java Keystore

This conversion method may be useful if a PKCS #7 or PKCS #12 certificate is unavailable for use with a Java Keystore. There are several ways to achieve this end result; this is one of the simpler options.

1. Using the `openssl` command (available in every Linux instance by default), export the key, certificate, and CA-Certificate into a PKCS #12 bundle.
```
openssl pkcs12 -export -in /path/to/certificate.crt -inkey /path/to/keyfile.key -chain
-CAfile /path/to/ca-file.crt -name "smrtlink.example.com" -out /path/to/pkcs12-
bundle.p12
```

   – Be sure the key has already been encrypted and the encryption pass phrase matches the export pass phrase that will be set when prompted after running the command below.

– If you see the message "Error unable to get issuer certificate getting chain", then concatenate the CA's certificate chain with your own CA signed certificate. The resulting combined certificate chain file may then be passed to `-CAfile`. In the example below, `/path/to/ca-file.crt` would be replaced by `/path/to/ca-file-chain.pem` instead.

```
cat /etc/ssl/cert.pem /path/to/certificate.crt > /path/to/ca-file-chain.pem
```

2. Import the resulting PKCS #12 file into a new Java Keystore:

```
keytool -importkeystore -deststorepass $KEYPW -destkeystore /path/to/
smrtlink_example_com.jks -srckeystore /path/to/pkcs12-bundle.p12 -srcstoretype pkcs12 -
deststoretype pkcs12
```

3. Determine the current alias for the imported PKCS #12 bundle and change it to `server`:

```
keytool -list -v -keystore /path/to/smrtlink_example_com.jks
keytool -changealias -alias smrtlink_example_com -destalias server -keystore /path/to/
smrtlink_example_com.jks
```

4. Proceed with Step 4 on page 15 of the **SMRT Link Software Installation's** SSL Certificate procedures.

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
```

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, as the root user, 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...

    Waiting for closing ports...

    Starting SMRT Link Services...
      Running apply-config...
      Checking services status...
        Checking SMRT Link Analysis services is not running...
        Checking SMRT Link Cromwell Server 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 Cromwell Server...
          Waiting for SMRT Link Cromwell Server to start...
        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...

    Status summary:
    SMRT Link status:  ok
    Services started successfully.
```

The last line of output should state `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...

    Status summary:
        SMRT Link 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, display a more detailed service status 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 the 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.

Note that using `grep` to search for the string `smrt` will result in **all** processes being 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: (**Note**: 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 --noexec
```

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"'
```

Checking JMS Functionality

Use the following command to have `runjmscmd` generate a very simple script that is submitted to the cluster with a request for only one slot:

```
$SMRT_ROOT/admin/bin/runjmscmd --start --test --exec --gen --wait
```

Output:

```
LOGNAME='smrtanalysis' PATH='/bin:/usr/bin:/bin' SGE_CELL='default' SGE_ROOT='/usr/share/gridengine'
TERM='screen-256color' USER='smrtanalysis' /bin/qsub -S /bin/bash -q sequel-farm -N J123.jmstestcmd -
o /opt/pacbio/smrtlink/userdata/jobs_root.default/jmstest/jmstestcmd.stdout -e /opt/pacbio/smrtlink/
userdata/jobs_root.default/jmstest/jmstestcmd.stderr -pe smp 1 /opt/pacbio/smrtlink/install/
smrtlink-release_8.0.0.80529/admin/bin/../../../../userdata/jobs_root/ jmstest/jmstestcmd.sh
Your job 1392693 ("J123.jmstestcmd") has been submitted
Job submit command completed successfully.
Waiting for job completion...
Job completed successfully (29 seconds).
```

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 **only** be added to the file `$SMRT_ROOT/smrtlink/userdata/user_jmsenv/user.jmsenv.ish`, or to the appropriate per-compute configuration variant.

This file contains the default JMS configuration's environment variables:

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

This is the Cromwell configuration, used for submitting and stopping jobs, among other things:

```
$SMRT_ROOT/smrtlink/userdata/generated/config/cromwell.conf
```

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`. As of SLURM 16.05.0, a `--wait` switch was included for `sbatch`, making it a blocking call. With the introduction of `pbcromwell` in SMRT Link v8.0, blocking calls are **no longer** used, 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. This clears any arguments specifically intended for `salloc` and `srun`, respectively.

Troubleshooting LSF Issues: Chunking is broken

Symptom: Analysis using LSF distributed computing fails.

SMRT Link uses `call-guess_optimal_max_nchunks` to calculate the desired `nchunks` value, and then writes the result to the `stdout` log file.

IBM LSF **also** writes log output to this file and parsing of the `nchunks` value is currently broken as a result.

Solution:

1. Contact PacBio Technical Support for a patch file.
2. Replace the file `$SMRT_ROOT/current/bundles/smrttools/current/private/pacbio/pbpipeline-resources/wdl.zip`.
3. Restart the SMRT Link services: `$SMRT_ROOT/admin/bin/services-start`

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
   ```

Changing the Default Administrative Password in WSO2

Symptom: SMRT Link services fail to start after the WSO2 Admin password is changed.

Diagnosis - Confirm the following:

1. The WSO2 password for the Admin user was changed to a string containing special characters; that is characters **other** than alphanumeric, hyphens, or underscores.
2. The password was changed using the `${SMRT_ROOT}/admin/bin/set-wso2-creds` command.
3. Output from the `services-start` command looks like the following:

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

Waiting for closing ports...

Starting SMRT Link Services...
  Running apply-config...
  Checking services status...
    Checking SMRT Link Analysis services is not running...
```

```
    Checking SMRT Link Cromwell Server 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 Cromwell Server...
      Waiting for SMRT Link Cromwell Server to start...
    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...
Failed running CREATE_ROLES Access Denied. Authentication failed - Invalid credentials
Errors detected in starting SMRT Link services.

Errors encountered, shutting down all services...

Stopping Services...
  Stopping SMRT Link Analysis services...
  Stopping SMRT Link Cromwell Server...
  Stopping WSO2 API Manager...
    Waiting for WSO2 API Manager to stop......
  Stopping SMRT Link GUI webserver...
  Stopping database...
All services stopped.

services-start: Error! Services not started properly, all services stopped
```

Solution:

1. Change the Admin password using the WSO2 API Manager interface. **Note**: PacBio recommends that the password includes **only** alphanumeric characters, underscores, or dashes.
2. Stop services: `$SMRT_ROOT/admin/bin/services-stop`
3. Manually change the passwords in the `user-mgt.xml` and the `jndi.properties` files.
4. Run `${SMRT_ROOT}/admin/bin/set-wso2-creds --user 'admin' --password 'newpw'`
5. Start services: `$SMRT_ROOT/admin/bin/services-start`

pbcromwell Help Corrections

When you use the `pbcromwell --help` command, some of the examples displayed are incorrect. Following are the **corrected** examples.

Generate `cromwell.conf` with HPC settings:

```
$ pbcromwell configure --default-backend SGE --output-file cromwell.conf
```

Launch a PacBio workflow:

```
$ pbcromwell run pb_ccs -e /path/to/movie.subreadset.xml --nproc 8 --config /full/path/to/
cromwell.conf
```

Run the CCS workflow:

```
$ pbcromwell run pb_ccs -e <SUBREADS> --nproc 8 --config /full/path/to/cromwell.conf
```

Run the Iso-Seq workflow, including mapping to a reference, and execute on SGE:

```
$ pbcromwell run pb_isoseq3 -e <SUBREADS> -e <PRIMERS> -e <REFERENCE> --nproc 8 --config /
full/path/to/cromwell.conf
```

Run a user-defined workflow:

```
$ pbcromwell run my_workflow.wdl -i inputs.json -o options.json --config /full/path/to/
cromwell.conf
```

Print details about the named PacBio workflow, including input files and task options. **Note**: The prefix `cromwell.workflows.` is optional.

```
$ pbcromwell show-workflow-details pb_ccs
$ pbcromwell show-workflow-details cromwell.workflows.pb_ccs
```

Iso-Seq does not Display as an Analysis Option with Demultiplexed Data

Symptom: In SMRT Analysis v8.0 after demultiplexing, the Iso-Seq application is **not** available from the Analysis Application droplist.

Solution: There are two solutions for this issue; you can choose either one.

SMRT Link GUI:

1. When creating the analysis, select **Data Type = CCS Data** (ConsensusReadSet), which are directly generated from the CCS Step.
2. Select one or more Data Sets, then click **Next**.
3. From the Analysis Application droplist, select **Iso-Seq**.
4. Fill in parameters and start the Analysis.
   - Note that these CCS reads are **not** demultiplexed; demultiplexing is performed by the Iso-Seq Application.
   - The Cluster step will be performed to reads from **all** barcoded samples. If Sample A and Sample B contains the **same** isoform, one isoform cluster will be reported.
   - Note that you have done Step 1 CCS and Step 2 Demultiplex, and you will start from Step 3 Refine, and all the way until Step 4 Cluster complete.

From the Command-Line:

Take a demultiplexed CCS BAM file for each barcode as input, and follow the documentation here: https://github.com/PacificBiosciences/IsoSeq/blob/master/README_v3.2.md#step-3---refine

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 the Data Set 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 > Import Data > Data Type = Sequel Data (XML)**. Select the file, then click **Import**.
   - 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

Users with Bioinformatician Role Cannot View Run QC

Users with the Bioinformatician role **should** be able to view the Run QC module, but currently cannot. This issue will be fixed in the next release of SMRT Link.

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 converts 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. 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.

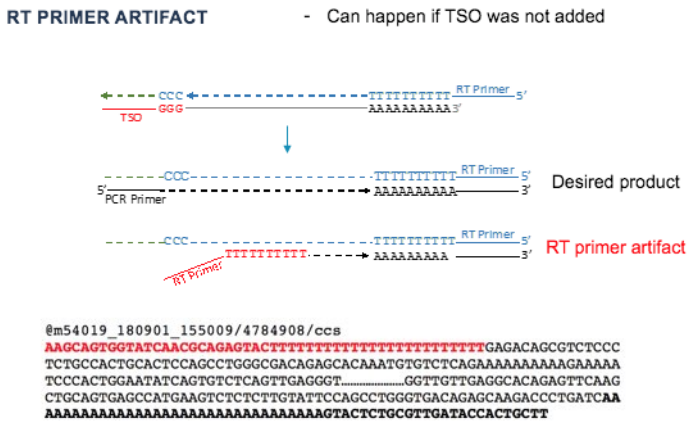
   The Iso-Seq primers are:

```
>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.



   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:

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**.

Microbial Assembly

Symptoms: Microbial Assembly analysis generates a microbial genome assembly with >5 contigs.

Solutions - Following are some suggestions to help improve the assembly, which can be tried individually or together.

1. To ensure sufficient coverage for assembly, set the **Genome Length** parameter to a value equal to (or slightly above) the actual genome size for the provided sample.

2. Increasing the coverage for assembly may help when the library quality is less than ideal.
   - In the **Create New Analysis** page, click **Advanced Parameters** and change the Coverage parameter from the default of 30 to a higher value, such as 40.

3. If the input subread length is relatively short (such as under 6000 bp) the minimum threshold for preassembled read length may need to be lowered. This can also help for longer insert libraries for which the quality is less than ideal.
   - In the **Create New Analysis** page, click **Advanced Parameters** and edit the Advanced options by adding the following line, which modifies the overlap filtering options by specifying a smaller `--min-len` value of 2000 bp (the default is 4000 bp):

     ```
     stage1.ovl_filter_opt = --max-diff 80 --max-cov 100 --min-cov 1 --bestn 20 --min-len
     2000 --gapFilt --minDepth 4;
     ```

4. In case the Microbial Assembly analysis is taking a long time to complete, try modifying the block size.
   - In the **Create New Analysis** page, click **Advanced Parameters** and edit the Advanced options by adding the following line, which changes the block size for the overlap process to a smaller value such as 200 Mb (the default is 1024 Mb):

     ```
     stage1.block_size = 200; stage2.block_size = 200;
     ```

     Note that the block sizes for both stages do **not** have to be identical, but in general, the same rule of thumb applies for both stages.

5. If the Microbial Assembly analysis stops due to an out-of-memory problem or overly high memory consumption in general, reducing the block size for the overlap process can help. Refer to the previous point for details on how to set the block size.

6. Optionally you can use the **Auto Analysis** option when designing the sequencing run to set up analysis for automatic execution once the sequence data is available on the SMRT Link server. For more information, see the document **SMRT Link User Guide (v8.0)**.

P/N 101-397-000 Version 04 (November 2019)