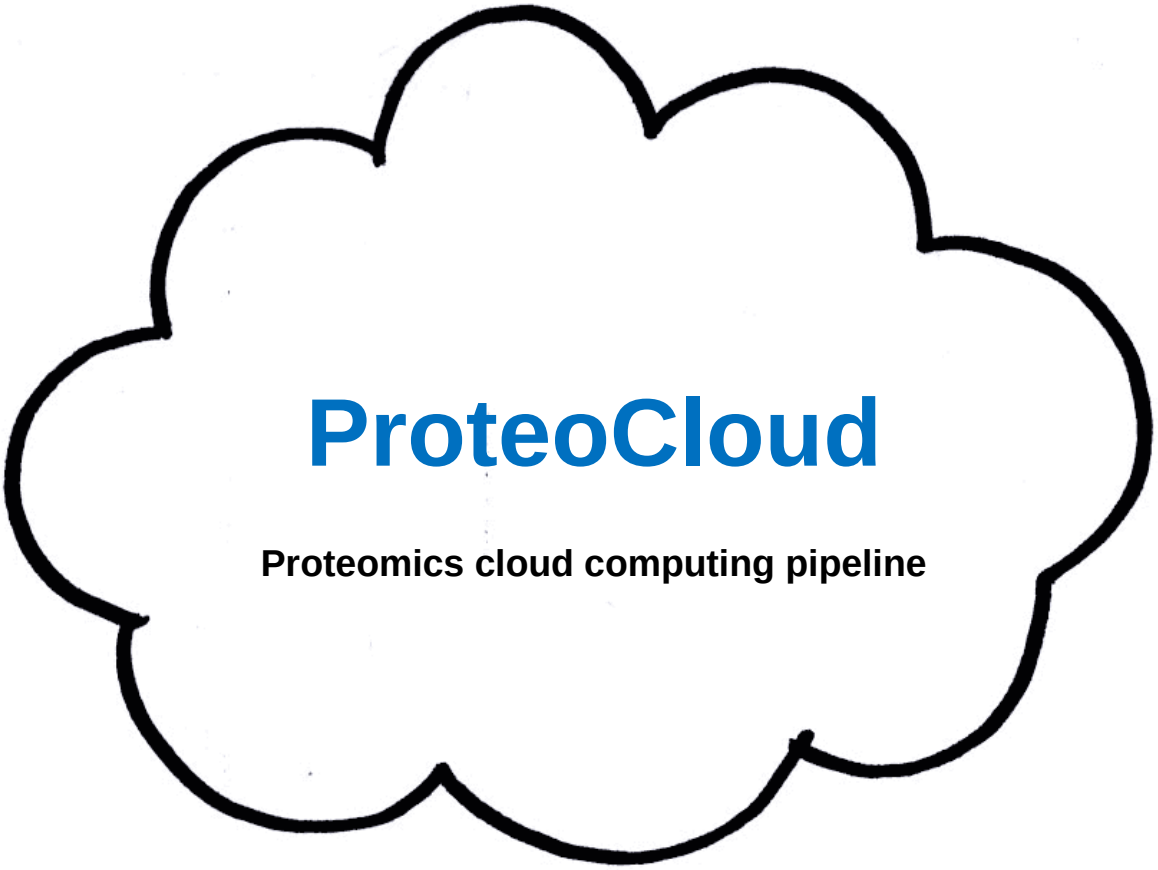

ProteoCloud

Proteomics cloud computing pipeline

Version 1.1

Documentation manual

by

Thilo Muth

Contents

Chapter 1

Introduction

ProteoCloud is a Java-based proteomics cloud computing pipeline system for peptide and protein identfications. It supports database searching and de novo. Requiring only MGF files as input the pipeline is able to identify peptides and proteins from tandem mass spectra. The search can be done on multiple servers and the results are stored on a centralized SQL server in the cloud.

The software has been developed by Thilo Muth (thilo.muth at gmx.de) and Prof. Lennart Martens (lennart.martens at gmail.com). Feel free to contact the corresponding authors whenever you have questions, comments or suggestions concerning **ProteoCloud**.

Chapter 2

Installing ProteoCloud

For installing ProteoCloud a proper java version should be installed and the proteocloud jar file (proteocloud.XYZ.jar) must be downloaded.

Downloading Java

Before installing ProteoCloud make sure you have Java 1.6 installed.
To check this, open a console/bash window and type:
java -version

If you haven't installed Java 1.6, you can find the download here:
http://www.java.com/download/

Downloading ProteoCloud

The latest version of ProteoCloud can be downloaded on:
http://proteocloud.googlecode.com/

Under the featured downloads you will find the latest version as jar file.
It is also possible to download the source files via SVN.

Chapter 3

Setup the ProteoCloud pipeline

- **The Controller Unit**

The Controller Unit represents the user interface and the master controller for ProteoCloud pipeline. Via the Controller Unit you can easily handle uploading of MS/MS files, launching the virtual server instances and the job managment for the different used search engines.

To start the Controller Unit, you need to unzip the downloaded file, fill the AWS credentials in the properties file (see below) and double-click on the ProteoCloud-XYZ.jar file.

- **Properties File**

The properties file **(proteocloud.properties)** can be found in the **conf** folder. In the properties file you need at least specify your AWS credentials (awsAccessKey, awsSecretKey, keypair and group) and the Amazon S3 bucket folder to have access to the cloud system.

For running searches in the cloud you need to specify the ProteoCloud AMI (listed on the ProteoCloud website) and the availability zone (default: us-east-1a). This can also be done in the application in the Project Settings dialog (see Chapter 4).

Further parameters can be set for the SQL database and project specific settings, such as taxon, precursor and fragment ion tolerances. More info on how to setup the SQL database server in the cloud, you can find in the following section.

- **SQL database**

In order to store the results from ProteoCloud searches, you need to provide a valid SQL database instance in the cloud. For the pipeline it is not necessary to install your own SQL database, but instead you can launch a pre-configured image (AMI) and assign a volume and an elastic IP to the booted instance. The instance comes with an Apache web server and a MySQL database.

For setting up the database it is recommended to use Elasticfox (a plugin for the Firefox-browser). In the following we show the various steps to take via this tools. Feel free to use the AWS management console or another API, then you have to do different steps as written in the following.

1. **Launch a SQL server instance:**

   In Elasticfox go to the **Instances** tab and select the public **ProteoCloud MySQL AMI** (see Proteocloud-website for the most recent version). By right-clicking on the entry, select *Launch instance(s) of this AMI* and follow the instructions to launch one instance of this AMI. You can find latest version and the identifier of the ProteoCloud MySQL AMI on the ProteoCloud website.

2. **Assign an elastic IP:**

   Navigate to the Elastic IPs tab in ElasticFox and click on the green plus (+) button. In order to associate the obtained elastic IP with an instance, right click on the IP and select *Associate Elastic IP with Instance*. In the following dialog select the previously launched MySQL server instance. Now you can reach the database under the specified elastic IP.

3. **Attach an EBS volume:**

   The last step is to attach the EBS (Elastic Block Storage) volume to launched MySQL server instance. This volume contains the file system used for the database and also a pre-configured structure of the database. Before attaching the volume

you need to create one from the ProteoCloud snapshot. The identifier of the snapshot (SNAP ID) you can find on the ProteoCloud website. On the Volumes and Snapshots tab, search for the ProteoCloud snapshop via the SNAP ID textfield on the bottom area (Saved Snapshots), right-click on the snapshot and select *Create new volume from this snapshop*. As volume size at least 100 GB is recommended. As availability zone take the default (us-east-1a). After the volume has been created you need to select it in the top area (Created Volumes), right-click on the volume and select *Attach this volume.* Select the Instance ID of the launched MySQL server instance, and write as device: **/dev/sdh**

You can the MySQL server by navigating to the following address:

http://[elastic IP]/phpmyadmin

➔ You need insert the previously obtained elastic IP in the URL

The standard access to the SQL database and PhpMyAdmin is the following:

Username: **proteocloud**

Password: **pcu2012**

Important:

Remember to change the access to the database, as the running Apache webserver will be accessible from the Internet. For this you need to login into the PhpMyAdmin application via your browser, add/change the user on the **Privileges** tab in PhpMyAdmin. Another possibility would be to change the database access via commandline (i.e. via SSH to the MySQL instance) and grant the privileges manually on the MySQL console:

```
GRANT ALL PRIVILEGES ON *.* TO username@'%' identified by "password";
```

After taking these configuration steps the ProteoCloud pipeline setup is done and the application can be used (see the following chapter).

Chapter 4

Using ProteoCloud

- **Project Settings**

When you start the application, you need to specify the different settings for the database (url, username + password), the project title and the parameters for the MS/MS identfication searches (taxon, fragment and precursori on tolerance). These settings are taken from the proteocloud.properties by default (see Chapter 3).

Figure 1 shows the Project Settings dialog. The same dialog can be found in the menu of the application window: → *Options* → *Project Settings [Alt + S]*

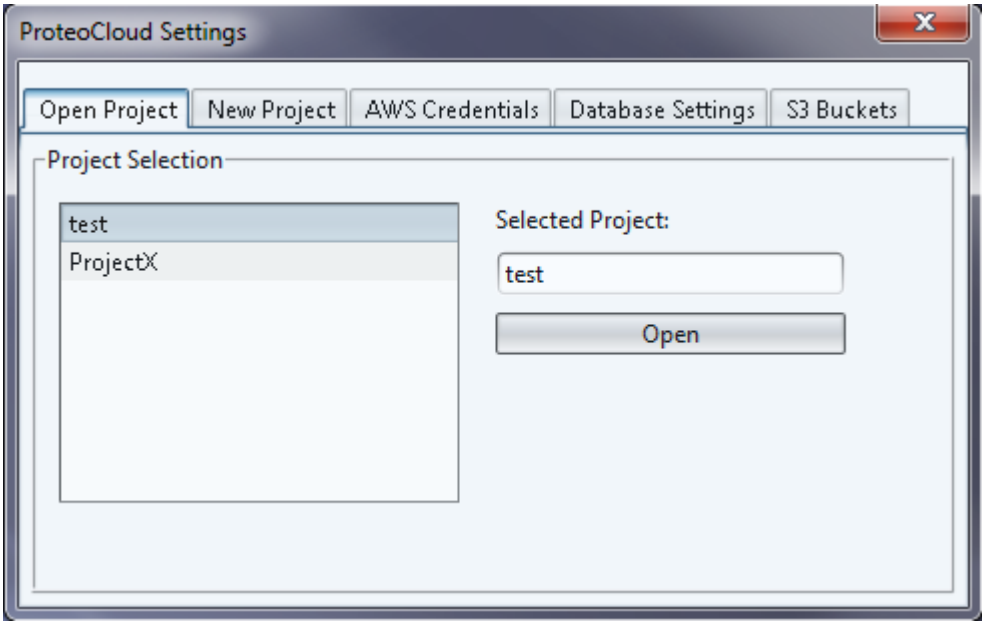

Figure 1: The Project Settings dialog

- **Input Files**

To choose the MGF files you want to process in the ProteoCloud pipeline you need to select the files on the top of the application general user interface (see Figure 2). After selecting the files you may change the chunk size. The size specifies the number of spectra that are used for on chunk. This has to be done to guarantee that a similar amount of data is provided to each of the running instances later on. 1000 MS/MS spectra constitute a conveient chunk size as default value. After specifying a new folder for the chunk files by means of the **Choose** button, you need to click on the **Chunk** button for splitting up the files. The file chunks are found in the folder previously specified. All that needs to be done is upload the file to the S3 Storage bucket. The uploaded files are listed in the Storage S3 Files table (see Figure 2).

- **Launch the instances**

On top of the master control section on the bottom of the main window the EC2 instances launch configuration can be found (see Figure 2). **Image ID** represents the identifier of the public ProteoCloud AMI. You can find the name of the latest ProteoCloud AMI listed on the ProteoCloud website. As availability zone the default is *us-east-1a.* The number of instances to boot needs to be specified and the instances can be booted via the **Launch** button. You can find a list of the booting and running instances on the **Instances** tab. (see Figure 4 and the text below.)

Important:
1000 MS/MS spectra consume roughly 1-2 hours compute time on one instance. E.g., if you want to search 10000 MS/MS (10 chunked MGF files) then you should consider 10 instances to boot. The instances are then searching in parallel as the provided spectra sets are distributed on the different machines.

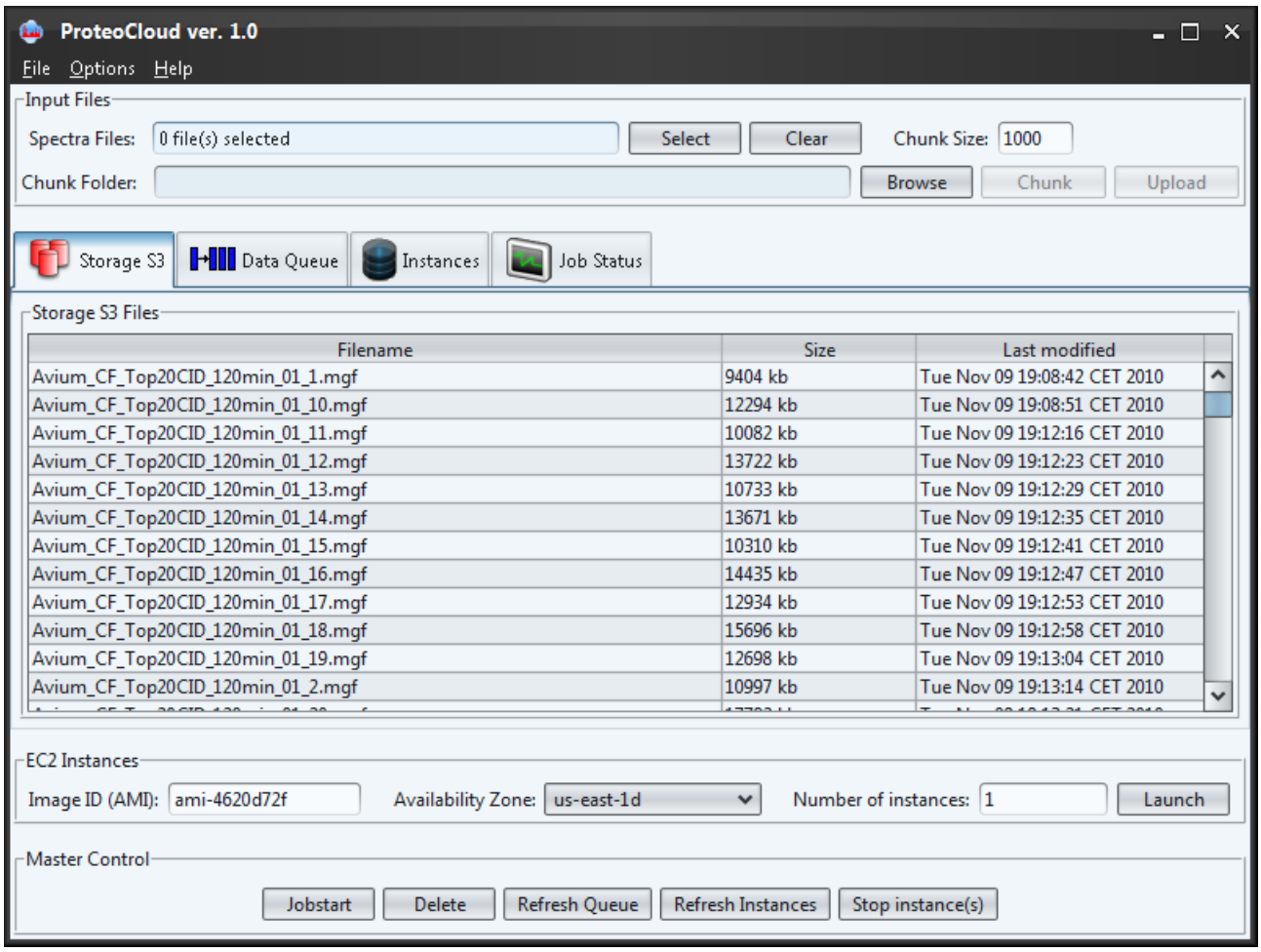

Figure 2: The general user interface and the Storage S3 view.

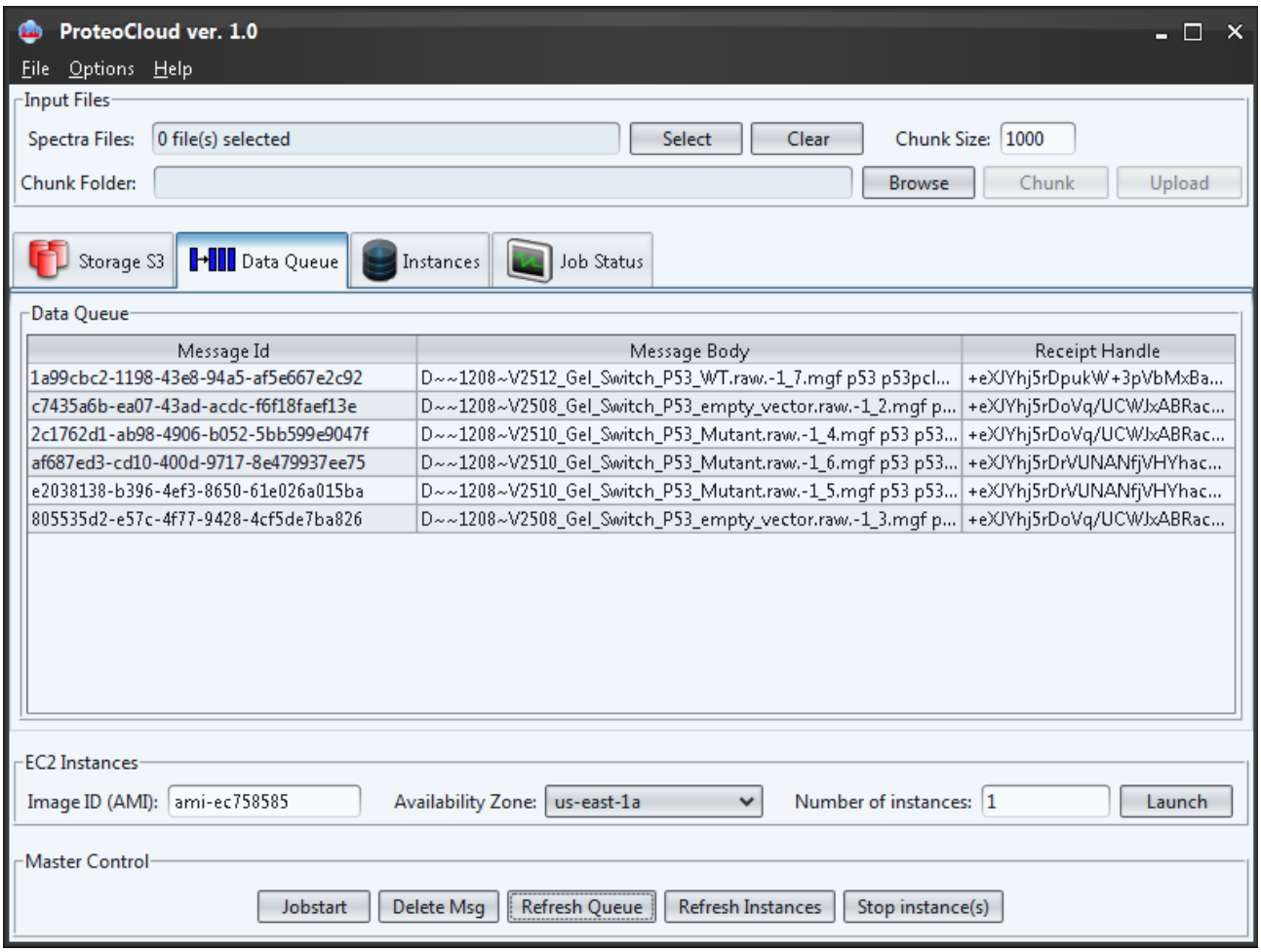

Figure 3: The Data Queue view

- **View of the files waiting in queue**

If you click on the **DataQueue** tab you can find the spectrum files which were added to the data queue and which are not yet processed (see Figure 3). These files wait for the instances to be taken. The column called **Message Body** contains the whole message string that was sent by the application when the user clicked on **Jobstart** previously.
By clicking on the **Refresh Queue** button the DataQueue view can be updated: Every time an instance processes a file, it will be removed from the data queue and the view table.
It is also possible to remove a file from the data queue by selecting its **Message Id** and hitting the **Delete** button.

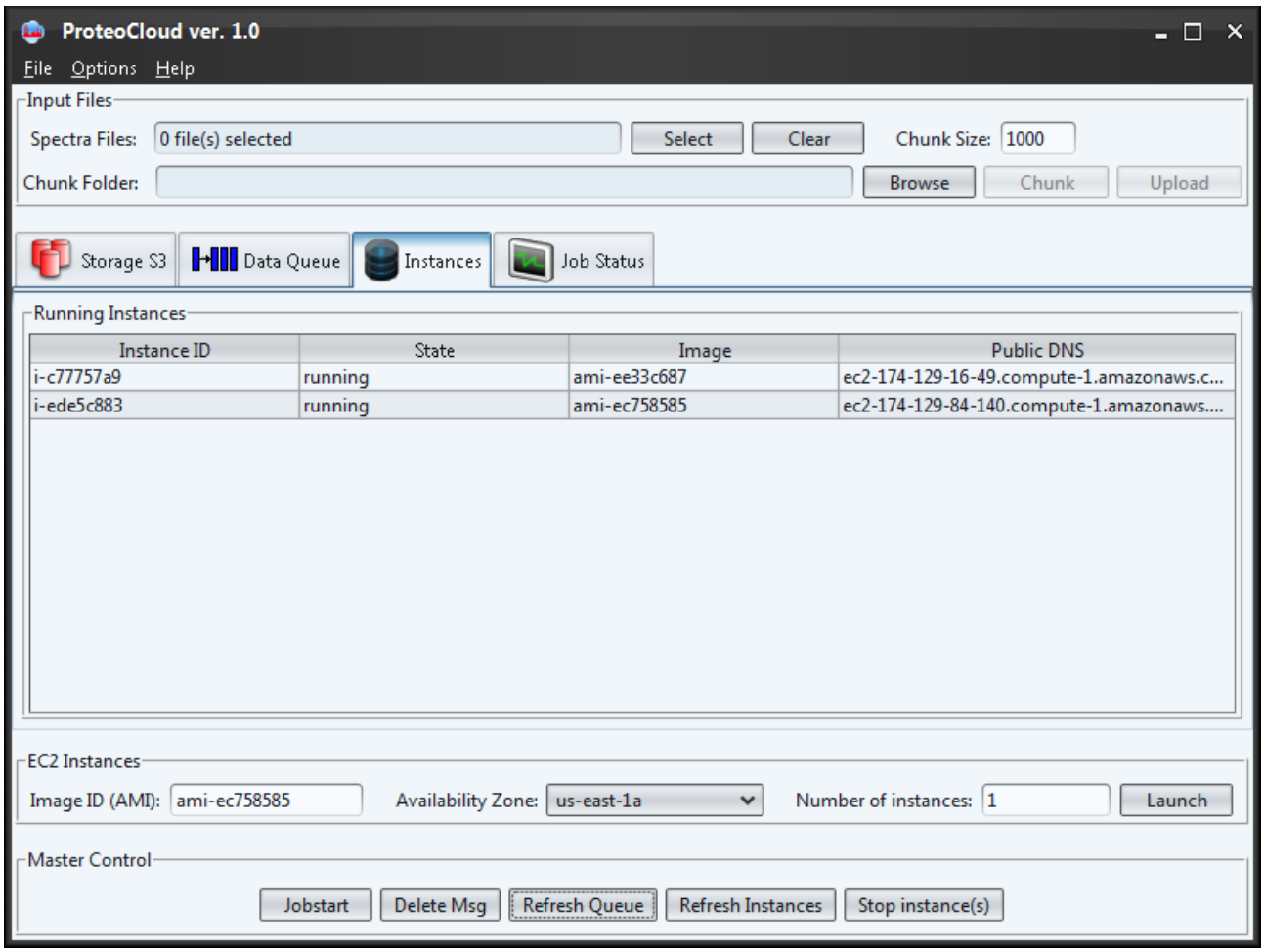

Figure 4: The Instances view

- **Instances**

By clicking on the **Instances** tab you can find find the running instances (see Figure 4). These files wait for the instances to be taken. The column called **Message Body** contains the whole message string that was sent by the application when the user clicked on **Jobstart** previously.

By clicking on the **Refresh Queue** button the DataQueue view can be updated: Every time an instance processes a file, it will be removed from the data queue and the view table.

It is also possible to remove a file from the data queue by selecting its **Message Id** and hitting the **Delete** button. The file then will not be processed from the instances anymore.

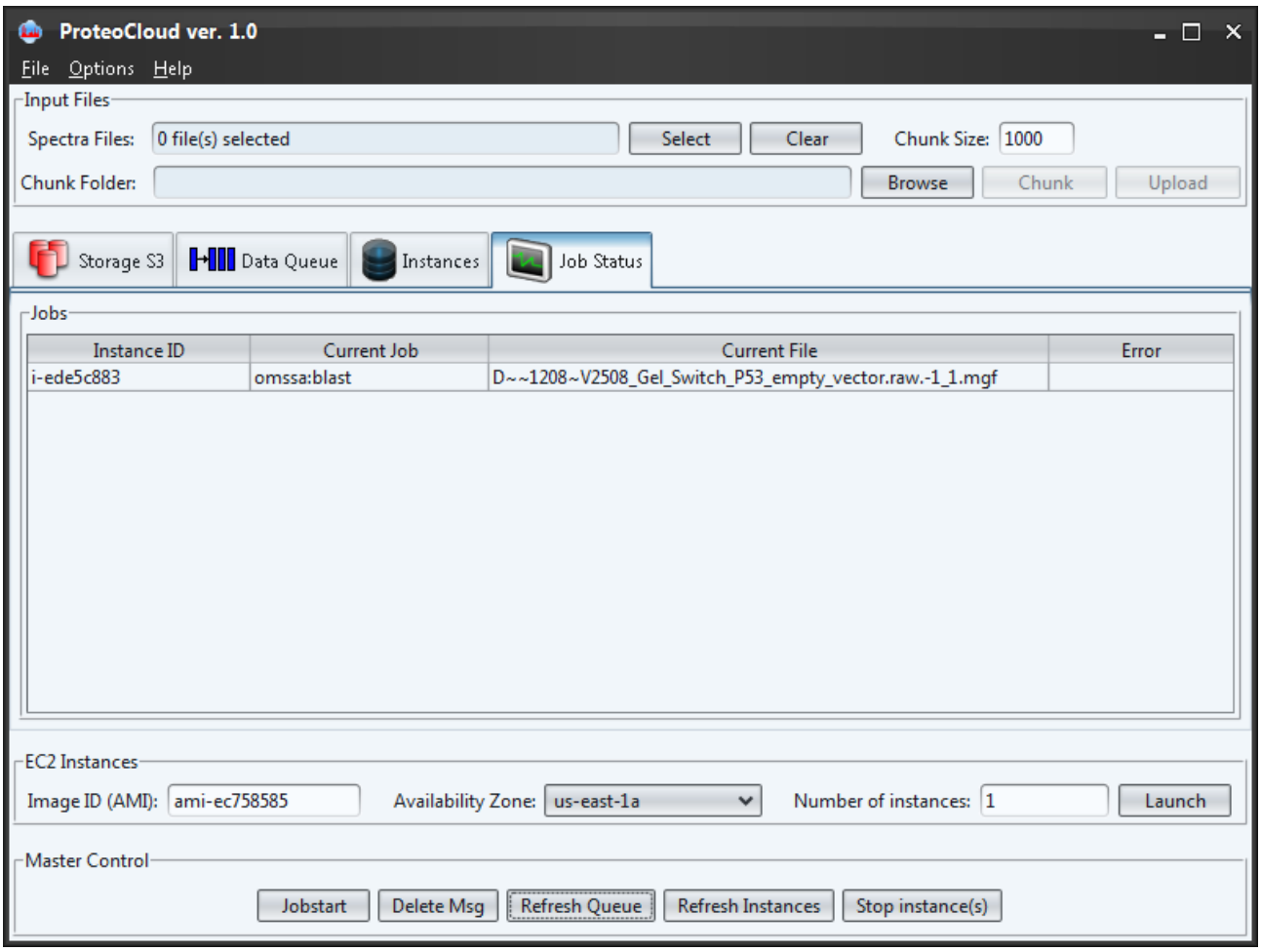

Figure 5: The Job Status view

- **Job Status**

By clicking on the **Job Status** tab (see Figure 5), you can find the current jobs that are running on the different server instances in the **Current Job** column. In the **Current File** column the processed spectrum file is diplayed. If any error occurs, it can be found in the **Error** column.

Once the instance has ended with the search job, it is signaled in the application as being finished in the **Current Job** column.

When all jobs are finished the instances can be shutdown via the **Stop instance(s)** button.

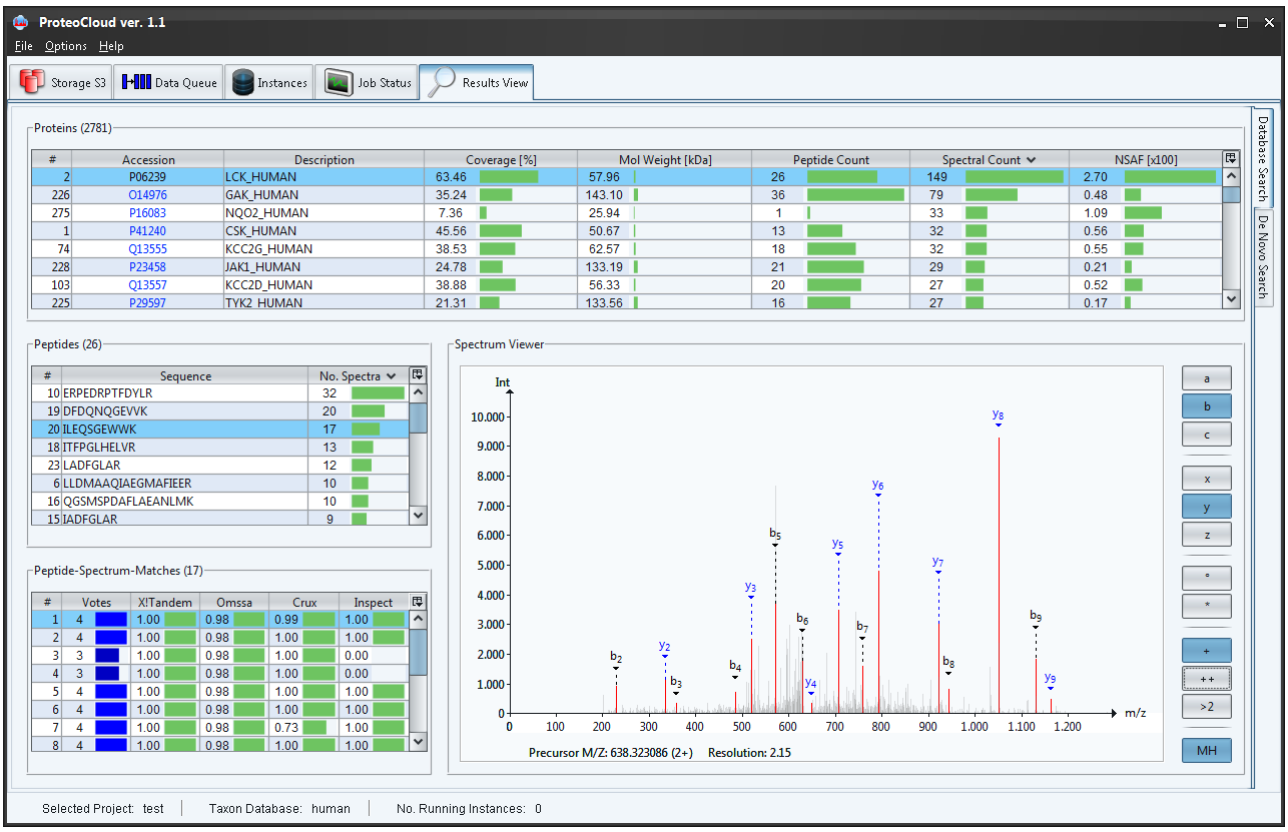

Figure 6: The results view

- **Results View**

By clicking on the **Result View** tab (see Figure 6), you can the protein result in the top display. By clicking into the table on one protein, the details for all related peptide identifications and peptide-to-spectrum matches (PSMs) are displayed. Additionally, the corresponding spectrum is shown where all details (e.g. fragment ions) can be observed.