“Parallel Protein Classification with IBM BigInsights”

Semester Thesis

Department of Computer Science
University of Applied Science Rapperswil

Spring Term 2013

Authors: Christof Büchi, Susanne Mathys
Advisor: Prof. Josef M. Joller
Project Partner: IBM Switzerland
External Co-Examiner: Romeo Kienzler
Abstract

*Big Data* is an expanding topic in information technology based on the huge collection of data which is available today on IT systems all over the world. Processing huge amounts of large files and analyzing unstructured data in real time could bring advantages for institutions or enterprise which store a large volume of generated data from their transactions.

Dealing with the rapid growth of data and analyzing it is crossing the boundaries of the given IT infrastructures. *Google* and *Yahoo!* have introduced their own way how to handle such datasets. A completely new architecture beyond well-known established tools and principles is required to store massive data efficiently in storage and process them with minimal overhead.

*Big Data* systems and frameworks such as *IBM BigInsights* with *Hadoop* provide a distributed fault-tolerant file system running on commodity hardware. They also allow writing custom applications in *Java* based on the *MapReduce* principle.

How difficult would it be to perform classification with a given single processing application on a *Big Data* system? During our research we wanted to show that it is as simple as setting up a cluster and running the tool out of a bash script that is used within a *Hadoop* streaming job. We took a look at the overhead of using such a complex framework for processing simple applications in a parallel manner. We also had a scope to the scale out characteristics of the cluster size.
Declaration

We, Christof Büchi and Susanne Mathys declare:

- This term project and the work presented in it is our own, original work.
- All the sources we consulted and cited are clearly attributed. We have acknowledged all main sources of help.

Rapperswil, May 31, 2013

Christof Büchi

Susanne Mathys
## Contents

1 Management Summary
   1.1 Problem Definition .................................................. 1
   1.2 Solution Proposal ..................................................... 1
   1.3 Sample Use Case ....................................................... 1
   1.4 Experiments ............................................................ 1
   1.5 Future Work ........................................................... 1

2 Introduction
   2.1 Objectives ............................................................. 2
   2.2 Motivation .............................................................. 2
   2.3 Ongoing situation ..................................................... 2
   2.4 Classification of proteins ........................................... 3
      2.4.1 Amino acids ....................................................... 3
      2.4.2 Proteins .......................................................... 3
      2.4.3 Protein sequence databases .................................. 4
   2.5 FASTA-format .......................................................... 5
      2.5.1 Classification ...................................................... 5
      2.5.2 Structural analysis .............................................. 5
      2.5.3 Sequence analysis .............................................. 5

3 System and Methods
   3.1 Hadoop ................................................................. 6
   3.2 Hadoop Streaming ..................................................... 6
   3.3 IBM BigInsights ........................................................ 6
   3.4 IBM BigSheets .......................................................... 7
   3.5 Classifiers .............................................................. 7
      3.5.1 RaptorX - structural classification ......................... 7
      3.5.2 InterProScan - sequence analysis ......................... 8
   3.6 MapReduce tasks ...................................................... 10

4 Configuration and setup
   4.1 Hadoop cluster with IBM BigInsights ............................. 11
   4.2 Cluster hardware configuration ................................... 12
   4.3 Preparing input dataset for HDFS ................................. 14
   4.4 Load data to HDFS .................................................... 14
   4.5 Running tasks with Hadoop Streaming ............................ 14
   4.6 Configuration parameters for Hadoop Streaming ............... 14
   4.7 Generating output for IBM BigSheets .............................. 15
   4.8 Scale out with nodes ............................................... 15

5 Results
   5.1 Facts ................................................................. 17
1 Management Summary

1.1 Problem Definition

In this project we will solve a large scale text classification problem using the massive parallel data processing infrastructure based on IBM InfoSphere BigInsights. Focus will not be on the classification algorithm itself but on ease of use in implementing such a system based on already existing classifiers and data sources to be integrated. Therefore we will focus on scalability and reuse.

1.2 Solution Proposal

At the beginning the data has to be retrieved and imported into the system. Depending on the data source, there are many options (e.g. BigInsights WebCrawler, Open Source Tools, HandWritten Crawler, ..) and therefore an appropriate solution has to be chosen and implemented. Depending on the type of data to be analyzed, the appropriate storage system has to be chosen (e.g. files in HDFS, HBASE, HIVE, ..). Based on Hadoop-Streaming, a library capable of integrating any UNIX command-line application able to read and write to Standard-Input/Standard-Output the integration of existing classification algorithms/software packages should be shown.

1.3 Sample Use Case

As data source, a publicly available protein database will be used. In order to copy a large test data set of protein sequences including their annotations into HDFS. Based on this protein sequence data a classifier will be used to classify these proteins into the following subclasses based on their amino acid sequence: alpha, beta, alpha + beta, alpha/beta, and zeta (irregular). Since the implementation of the classifier itself is not in the scope of this work, the classification performance will not be considered.

1.4 Experiments

The experimental evaluation will be performed in two dimensions: data set size and cluster size. Based on these recordings, the scale out coefficient of the system will be determined. It will be checked whether it holds the theoretical assumption of linear scale out, and if not, a bottleneck analysis will be performed.

1.5 Future Work

A further extension to the sample use-case could be enriching the protein information with the publication references. Based on the referenced abstracts and the already classified proteins, a text analytics plugin (SystemT) provided by IBM BigInsights could be used to classify these abstracts as well. The correlation between the protein classifier and the text classifier can also be drawn.
2 Introduction

2.1 Objectives

During our research we covered the following goals:

- Demonstrating the simple use of IBM BigInsights
- Running an existing executable on the IBM BigInsights cluster
- Achieving linear scale out behavior in experiments
- Determining overhead of using Hadoop framework for parallel processing

For the present use case, a command-line tool was used to run the Hadoop Streaming [8] framework in a cluster without considering the parallel implementation of the tool.

2.2 Motivation

In the past years science moved forwards thanks to improved technical capabilities. In the field of genetics and genomics large amounts of raw data [?] has been retrieved. To take benefit of this collection and draw some conclusions, all the data has to be analyzed for further insights. It is still difficult to handle massive datasets. A lot of time is invested to scale out known algorithms, such as classifying genoms and proteins. With our work, we want to show the simplicity of building a distributed system.

2.3 Ongoing situation

To process a big set of input data in parallel system bears some problems. First: The more different hardware components are used, the more hardware failures [18] are likely to occur. Mitigating these can be handled by software. Second: Big dataset have to be partitioned over several hosts for fault tolerance and raised I/O operations per second. Third: After parallel processing on different systems, a task is needed that aggregates the results of each system to one final result. In 2008 [6] Google introduced the MapReduce principle for processing data on clusters. The Hadoop [17] framework implements this principle. IBM released InfoSphere BigInsights which is based on Hadoop and brings some additional features [5] providing simplicity to manage Big Data.
2.4 Classification of proteins

There are numerous schemes to classify a protein in different groups. The challenge is that all implemented algorithms and tools do not provide as high a level of certainty and accuracy as classification of the proteins by human experts. On the other hand, humans have a high time consumption to complete such tasks.

2.4.1 Amino acids

Amino acids are essential elements of the organic system. Twenty three amino acids are known as proteinogenic. These amino acids are found in proteins. There is an official representation for amino acids which assigns a certain character to every amino acid. Table 2.1 shows the IUB/IUPAC standard codes.[14]

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Alanine</td>
</tr>
<tr>
<td>B</td>
<td>Aspartic acid or Asparagine</td>
</tr>
<tr>
<td>C</td>
<td>Cysteine</td>
</tr>
<tr>
<td>D</td>
<td>Aspartic acid</td>
</tr>
<tr>
<td>E</td>
<td>Glutamic acid</td>
</tr>
<tr>
<td>F</td>
<td>Phenylalanine</td>
</tr>
<tr>
<td>G</td>
<td>Glycine</td>
</tr>
<tr>
<td>H</td>
<td>Histidine</td>
</tr>
<tr>
<td>I</td>
<td>Isoleucine</td>
</tr>
<tr>
<td>K</td>
<td>Lysine</td>
</tr>
<tr>
<td>L</td>
<td>Leucine</td>
</tr>
<tr>
<td>M</td>
<td>Methionine</td>
</tr>
<tr>
<td>N</td>
<td>Asparagine</td>
</tr>
<tr>
<td>O</td>
<td>Pyrrolysine</td>
</tr>
<tr>
<td>P</td>
<td>Proline</td>
</tr>
<tr>
<td>Q</td>
<td>Glutamine</td>
</tr>
<tr>
<td>R</td>
<td>Arginine</td>
</tr>
<tr>
<td>S</td>
<td>Serine</td>
</tr>
<tr>
<td>T</td>
<td>Threonine</td>
</tr>
<tr>
<td>U</td>
<td>Selenocysteine</td>
</tr>
<tr>
<td>V</td>
<td>Valine</td>
</tr>
<tr>
<td>W</td>
<td>Tryptophan</td>
</tr>
<tr>
<td>Y</td>
<td>Tyrosine</td>
</tr>
<tr>
<td>Z</td>
<td>Glutamic acid or Glutamine</td>
</tr>
</tbody>
</table>

2.4.2 Proteins

Proteins are built as a chain of different amino acids. Some proteins are well-known and their functions are clearly identified. A certain protein is classified by predicting its properties and function. To predict a proteins’ function, an algorithm has to compare the sequence of amino acids with all sequences in a database of already known proteins.
2.4.3 Protein sequence databases

A number of databases which store information about proteins their amino acid sequences and properties.

The UniProt knowledge base supports two different databases:

- Swiss-Prot database [19], where all 539,616 entries are manually annotated and reviewed.
- TrEMBL database [20] with 32,153,798 entries, which are automatically annotated and are not beeing reviewed.

There are releases of the UniProt database every four weeks. It is possible to download the whole database as FASTA-file from UniProt’s FTP directory [4].
2.5 FASTA-format

The FASTA-format [7] is used to store a sequence of amino acids. There are two different kinds of records for each sequence:

- **Header line**
  Contains the description and identification of the sequence starting with the symbol “>” directly followed by an ID of the database and then followed by a blank and a description name.

- **Sequence line**
  One or more lines with the sequence of amino acids. Each line should not be longer than 80 characters. The amino acids are written in the IUB/IUPAC [14] standard code of amino acids.

| >sp|P69905|HBA_HUMAN Hemoglobin subunit alpha OS
| MVLSPADKTNVKAAGKGAHEGAEALERMFSLFPTTTKYPFHDLSHGSAQVKGHG
| KKVDAALTNAVYDMFNALSALSDLHANHLRVDTPVNFKLLSHCLLYTLAALPVEFTP
| AVHASLDKFLASVTLTSKYR

**Listing (2.1) FASTA-Format**

2.5.1 Classification

The spotlight is set on two different classifications: structure analysis and sequence analysis.

2.5.2 Structural analysis

In structural analysis every amino acid is assigned in a different structural category. We focus on the three classes: alpha-Helix(H), beta-Strand(E) and Loops(C). For structural analysis we take advantage of the RaptorX command line tool.

![Figure (2.1) Protein secondary structure [3]](image)

2.5.3 Sequence analysis

In sequence analysis, the given amino acid sequence will be compared or aligned to all known sequences and their properties. InterProScan will provide an ID and some references to the GO database, so that some information about the properties of the given sequence is returned.
3 System and Methods

3.1 Hadoop

Based on the idea of the MapReduce principle from Google [6], Yahoo! began to build an open-source alternative called Hadoop. Hadoop handles massive amounts of data in a distributed system. It allows to develop applications based on the Hadoop framework to run their own tasks, which are controlled by a central administration-node. It is built with a small amount of administration-nodes and many worker-nodes. The administration-nodes are used as a job-tracker and namenodes to distribute and manage tasks on the worker-nodes. If one worker-node fails, the task is marked as incomplete and will be sent to another worker-node. This allows a network with heterogeneous system as a cluster which carry on the same work. This calculation is normally done on one system and sequential for all sequences.

The Hadoop Framework is partitioned into different components: HDFS, MapReduce and common utilities. The HDFS provides an filesystem-layer over all distributed nodes. MapReduce allows one to write custom software and to run map reduce jobs. Common utilities are used to manage, start and stop a Hadoop cluster.

3.2 Hadoop Streaming

Hadoop provides a utility called Hadoop Streaming [8] to build map or reduce tasks in other languages than Java. It is possible to build and run a map and reduce task without programming and configuring Java Classes.

Hadoop Streaming asks for a executable or script which reads and writes to stdin and stdout. The given task is then built, executed and monitored directly on the cluster.

3.3 IBM BigInsights

IBM BigInsights contributes an easy to use webconsole, web-installer and many analytic and developer-related tools. It contains Apache Hadoop with associated products for a ready-to-use environment. Furthermore, it bundles a specific version of Hadoop and its components such as hive, flume, oozie, hbase and others within one component, which can be selected during the installation.

The following are the benefits [5] of using IBM BigInsights:

- Easy to use installer
- Realtime-view of cluster status and fine graded perspective on tasks and jobs
- Distribution of configuration-files over all nodes
- Scripts for cluster-management (adding node, healthcheck of cluster, complete cluster start and stop)
- Enhanced security with LDAP-access
• Flexible job scheduling mechanism
• Developer related functions such as eclipse plugin for linking references and automatic upload of code fragments
• Simple deployment and distribution of custom java applications on all nodes
• Already pre-deployed sample applications, such as wordcount

3.4 IBM BigSheets

IBM BigSheets allows one to use the output of the MapReduce task with a spreadsheet-like environment. For large output-files IBM BigSheets allows loading the data in a lazy-loading manner. It has many functions built-in such as counting rows, sorting output and building charts based on files inside HDFS based on MapReduce principle. IBM BigSheets is very useful where the data size is larger than normal spreadsheet tools can handle.

3.5 Classifiers

While researching the topic of classifying proteins, the following two classifiers were found: InterProScan and RaptorX. Both process FASTA-files with their own implemented algorithm. The tools were used within a Hadoop Streaming job to calculate the sequence of some proteins in a parallel manner.

3.5.1 RaptorX - structural classification

With the help of co-examiner Romeo Kienzler we established a contact to Jinbo Xu 1. Jinbo provided us the standalone RaptorX-Classifier, which we built into our Hadoop cluster. RaptorX is used to predict the secondary structure [15] of a protein based on a pre-calculated library. The following command shows the usage of this classifier:

```
./buildFeature -i query.seq -c 2
```

Listing (3.1) Raptor-Command

The output is a long list of information. The relevant information for our work are the secondary structures [15]. The structure is predicted from PSIPRED [15] based on the amino acid likelihood. The amino acids are grouped the following subclasses:

• H (alpha helix)
• E (beta strand)
• C (loops)

The following table 3.1 (page 8) shows the output of the classifier

---

1 Jinbo Xu, main-author of the RaptorX Classifier, http://ttic.uchicago.edu/~jinbo/
3.5.2 InterProScan - sequence analysis

*InterProScan* is the classifier provided by the European Bioinformatics Institute\(^1\). It predicts not only through one analysis, it also supports many different algorithms which can be used through a single interface. All algorithm could be used at the same time for an optimal prediction. Figure 3.1 shows all the supported algorithms on *InterProScan*.

![Figure (3.1)](image)

**Figure (3.1)** Supported algorithm on InterProScan [12]

Dr. Rémy Bruggmann\(^2\) recommended to use the *PfamA-26.0* algorithm which makes a prediction about the appropriate family and domain of the input-sequence. The following command shows the usage of *InterProScan*:

---

1 The European Bioinformatics Institute [http://www.ebi.ac.uk/](http://www.ebi.ac.uk/)
2 Dr. Rémy Bruggmann, Group Leader Bioinformatics University of Berne, [http://www.bioinformatics.unibe.ch](http://www.bioinformatics.unibe.ch)
The command outputs an XML file with description about the family and domain:

```xml
<hmmer3-match evalue="2.5E-28" score="98.4">
  <signature ac="PF00042" desc="Globin" name="Globin">
    <entry ac="IPR000971" desc="Globin" name="Globin" type="FAMILY">
      <go-xref db="GO" id="GO:0005506"/>
      <go-xref db="GO" id="GO:0020037"/>
    </entry>
    <models>
      <model ac="PF00042" desc="Globin" name="Globin"/>
    </models>
  </signature-

<s
locations>
  <hmmer3-location env-end="107" env-start="7" score="97.9" evalue="3.6E-28" hmm-start="1" hmm-end="108" hmm-length="0" start="7" end="107"/>
</locations>
</hmmer3-match>
```
3.6 MapReduce tasks

Hadoop Streaming allows us using already existing executable binaries. Hadoop Streaming only support read and write from stdin/stdout. Given the executables we used, only working with files for input and output data we had to encapsulate those tools within a bash script. The script runs after the mapping-process and covers following necessary actions:

- Reads the input stream and converts the one-line input (see section 4.3) to the necessary FASTA-format (see section 2.5)
- Reads out the sequence-id of the stream for sequence recognition and creates a temporary-file
- Calls the classifier with this temporary-file
- Gets the output of the classifier and filters the necessary information
- Passes the id with classified information to stdout.

For all jobs only one reduce task is used to reduce all different output files on the workernodes to only one output file in HDFS.

Referencing to Listing A.2 (appendix p. 25) for more information about our implementation of the map task for the classifier RaptorX.

Further consult the Listing A.4 of InterProScan bash script (appendix p. 27) for more details.
4 Configuration and setup

To run the use-case on a Hadoop cluster the following tasks and steps are necessary:

- Setting up a cluster with IBM BigInsights
- Downloading all sequences in one large input-file
- Preparing and modifying the input file for saving on HDFS
- Loading data into HDFS
- Configuring a MapReduce job with Hadoop Streaming and running it on the system.
- Formatting Hadoop Streaming output to a CSV-output file for import in IBM BigSheets
- Interpreting the generated data

4.1 Hadoop cluster with IBM BigInsights

For a basic Hadoop runtime-environment, a single node cluster is needed at a minimum. IBM BigInsights provides a single-node-cluster installation as a test-environment or as a complete installation including worker-nodes through the installation wizard. The wizard made the installation much easier.
4.2 Cluster hardware configuration

Thanks to IBM Switzerland we executed our experiments on a bladecenter hardware cluster.

Figure (4.1) Test environment: hardware cluster located at IBM Switzerland

To conduct performance analyzes, we chose ten worker nodes with an identical performance. Hadoop has a primary focus on heterogeneous systems, to enable it to operate on different commodity hardware. However, for our use-case that was less relevant, since we wanted to illustrate scale out performance and the resulting overhead.

The following table lists hardware specification and the used role inside the Hadoop cluster for each blade.
Table (4.1) The servers of our test environment

<table>
<thead>
<tr>
<th>Server</th>
<th>Model</th>
<th>Function</th>
<th>CPU</th>
<th>Memory</th>
<th>IP</th>
<th>FQDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blade1</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.101</td>
<td>Sa-biginsights-110-20-101-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade2</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.102</td>
<td>Sa-biginsights-110-20-102-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade3</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.103</td>
<td>Sa-biginsights-110-20-103-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade4</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.104</td>
<td>Sa-biginsights-110-20-104-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade5</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.105</td>
<td>Sa-biginsights-110-20-105-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade6</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.106</td>
<td>Sa-biginsights-110-20-106-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade7</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.107</td>
<td>Sa-biginsights-110-20-107-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade8</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.108</td>
<td>Sa-biginsights-110-20-108-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade9</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.109</td>
<td>Sa-biginsights-110-20-109-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade10</td>
<td>IBM HS21</td>
<td>task node, data node</td>
<td>4</td>
<td>12 GB</td>
<td>10.110.20.110</td>
<td>Sa-biginsights-110-20-110-rh5.tec.app.ibm.com</td>
</tr>
<tr>
<td>Blade14</td>
<td>IBM HS22V</td>
<td>job tracker, name node</td>
<td>16</td>
<td>30 GB</td>
<td>10.110.20.114</td>
<td>Sa-biginsights-110-20-114-rh5.tec.app.ibm.com</td>
</tr>
</tbody>
</table>
4.3 Preparing input dataset for HDFS

In HDFS all files are stored over different nodes. The default block-size in IBM BigInsights is configured with 128MB memory. To process an input dataset, Hadoop reads the file block by block and delivers this to the map task. There are different input-formats, per default the file content is provided line by line to the map task.

The input of the classifier has to be in FASTA-format [7]. However the FASTA-format consists of more than one line text-input for a protein sequence. This problem was solved by writing a short C++ program that reads FASTA-input file which was downloaded from UniProt [4] and wrote out one line for each sequence. Whereas the one-line record had to be split afterwards back to FASTA-format, we added as delimiter symbol ”%” after the header line to differ the headline with the sequence.

For more information about our program refer to the listing A.1 in the Appendix (page 25).

4.4 Load data to HDFS

To load some data into HDFS IBM BigInsights provides an upload function in the web console. This also can be done by simple line commands [10]:

```bash
hadoop fs -put localfile /user/hadoop/hadoopfile
```

Listing (4.1) HDFS put

4.5 Running tasks with Hadoop Streaming

Thanks to the Hadoop Streaming utility, a map task can be specified through any executable or script.

The following is an example of the command that was issued during performance tests:

```bash
hadoop jar /opt/ibm/biginsights/IHC/hadoop-streaming.jar -input /user/biadmin/fasta/input/db_dump_short -output output_10run -mapper /opt/scripts/raptorX-run.sh
```

Listing (4.2) Streaming Job

4.6 Configuration parameters for Hadoop Streaming

Within the Hadoop Streaming command it is feasible to adjust parameters of the Hadoop instance for the current running job.

In Table 4.2 (page 15) all parameters that were used by us are described.
4 Configuration and setup

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-D mapred.map.tasks=*</td>
<td>defining number of Map Tasks based on input sequences</td>
</tr>
<tr>
<td>-D mapred.task.timeout=*</td>
<td>for RaptorX set timeout to a value above 5 min</td>
</tr>
<tr>
<td>-D keep.failed.task.files=true</td>
<td>If true, the files for failed tasks will be kept</td>
</tr>
<tr>
<td>-D mapred.map.max.attempts=*</td>
<td>maximal attempt of restart a failed task(default 4)</td>
</tr>
<tr>
<td>-D stream.non.zero.exit.is.failure=false</td>
<td>ignore non zero return code of executable(default false)</td>
</tr>
</tbody>
</table>

4.7 Generating output for IBM BigSheets

*Hadoop* writes the result of the mapping task to it’s file-system. With *IBM BigInsights* these output files can be imported to *BigSheets* with a single click in the web console. Simply write output to stdout in CSV format and import the file from *HDFS* to *IBM BigSheets*. The CSV from our project is generated by writing the sequence id and the information gained from the output of the classifier.

4.8 Scale out with nodes

*IBM BigInsights* provides shell scripts for managing the amount of nodes on a cluster. The following command removes nodes from the cluster:

```
$ /opt/ibm/biginsights/bin/removenode.sh sa-biginsights-110-20-110-rh5.tec.app.ibm.com
```

Listing (4.3) remove node

To evade the default replication factor within *HDFS*, the force option on the command has to be issued:

```
$ /opt/ibm/biginsights/bin/removenode.sh hadoop -f sa-biginsights-110-20-109-rh5.tec.app.ibm.com
```

Listing (4.4) force remove node

Using the force option automatically decrements the value of replica of files in *HDFS*. (If the number of nodes fall off the defined property.) The property is named dfs.replication and the default value is three.

To change the replication factor on files, the following command can be used:

```
$ hadoop dfs -setrep -R -w 1 /user/biadmin/
```

Listing (4.5) change replication factor to one for directory /user/biadmin

Adding nodes is also performed with scripts:
Other management commands are starting and stopping the cluster:

```
./opt/ibm/biginsights/bin/start-all.sh
```
Listing (4.7) start the cluster

```
./opt/ibm/biginsights/bin/stop-all.sh
```
Listing (4.8) stop the cluster
5 Results

5.1 Facts

Below are the measurements of the experiments with the InterProScan tool

<table>
<thead>
<tr>
<th>sequences</th>
<th>native</th>
<th>1 node</th>
<th>5 nodes</th>
<th>10 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>117</td>
<td>133</td>
<td>138</td>
<td>137</td>
</tr>
<tr>
<td>5</td>
<td>585</td>
<td>617</td>
<td>136</td>
<td>139</td>
</tr>
<tr>
<td>10</td>
<td>1167</td>
<td>1217</td>
<td>253</td>
<td>251</td>
</tr>
<tr>
<td>40</td>
<td>4661</td>
<td>4842</td>
<td>1092</td>
<td>622</td>
</tr>
<tr>
<td>100</td>
<td>11673</td>
<td>12082</td>
<td>2541</td>
<td>1369</td>
</tr>
<tr>
<td>250</td>
<td>29146</td>
<td>30194</td>
<td>6449</td>
<td>3246</td>
</tr>
<tr>
<td>500</td>
<td>58314</td>
<td>62348</td>
<td>12828</td>
<td>6261</td>
</tr>
<tr>
<td>1000</td>
<td>116622</td>
<td>124537</td>
<td>25773</td>
<td>12497</td>
</tr>
</tbody>
</table>

Below are the measurements of the experiments with the RaptorX tool

<table>
<thead>
<tr>
<th>sequences</th>
<th>native</th>
<th>1 node</th>
<th>5 nodes</th>
<th>10 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>425</td>
<td>442</td>
<td>441</td>
<td>472</td>
</tr>
<tr>
<td>5</td>
<td>2128</td>
<td>2157</td>
<td>443</td>
<td>509</td>
</tr>
<tr>
<td>10</td>
<td>4304</td>
<td>4306</td>
<td>876</td>
<td>515</td>
</tr>
<tr>
<td>40</td>
<td>17026</td>
<td>17191</td>
<td>3459</td>
<td>1767</td>
</tr>
<tr>
<td>100</td>
<td>42569</td>
<td>44137</td>
<td>9065</td>
<td>4548</td>
</tr>
<tr>
<td>250</td>
<td>106427</td>
<td>110557</td>
<td>22409</td>
<td>11572</td>
</tr>
</tbody>
</table>
5.2 Cost of parallel processing

Figure 5.1 and 5.2 demonstrate the overhead of processing the sequences with InterProScan on our Hadoop cluster.

**Figure (5.1)** Mean-overhead of a InterProScan run with 100 various sequences

**Figure (5.2)** Mean-overhead of a InterProScan run with 250 various sequences
Figure 5.3 and 5.4 demonstrate the overhead of processing the sequences with RaptorX on our Hadoop cluster.

The values are calculated by following formula:

\[
\frac{\text{measured time}}{\text{amount of sequences} \times \text{cluster size}}
\]

Adding more nodes raises the CPU-power for handling more sequences in parallel. On the other hand managing more nodes will increase the average calculation time for one sequence, which results in a higher overhead per sequence.
5.3 Scale out

Scale out describes performance characteristics of adding more nodes to a cluster. This development is preferable linear, which means adding one more host to an already existing one node cluster results in double computing power. The additional needed power to distribute the tasks is called overhead and should be as less as possible.

As displayed in figure 5.5 the scale out behavior is almost linear. Adding more nodes results in linear less computation time. We normalized the measurement values with the formula:

\[
\text{normalized value} = \frac{\text{number of sequences}}{\text{measured time} / 60}
\]

The calculated values describes the throughput. The throughput itself describes the possible sequence calculation per minute.

As displayed in figure 5.6 the scale out behavior on our cluster with the usage of RaptorX for equal 100 sequences and equal 250 sequences is also almost linear.

\[\text{Figure (5.5) Throughput of InterProScan with 250 various sequences}\]

\[\text{Figure (5.6) Throughput of RaptorX}\]
5.4 Interpretation of the generated output files

5.4.1 InterProScan

As mentioned in section (System and Methods) InterProScan will provide us IDs of the InterProScan database and of the GO database. To give an overview of the used GO-IDs and IPR-numbers on the InterProScan output file, we ran the sample application “WordCount” on cluster. The following charts expose the occurrence of the GO-ID that InterProScan reported for 500 randomly chosen sequences:

![Figure (5.7) Occurrence of GO-IDs](image)

The GO-ID “GO:0005524” had the greatest occurrence with 62 counts. The entry is associated with the property “ATP binding”. The second most occurrence was the GO-ID “GO:0005622” with the property “intracellular”. 227 GO-IDs appeared only once. In Table B.1 (appendix p. 30) all GO-IDs with three or more occurrences of GO-IDs in the output file are listed. The following chart illustrate the frequency of the IPR-numbers that InterProScan reports from 500 sequences:
Five of the 500 sequences belong to the domain “RNA-binding S4”, which was reported through IPR-number IPR002942 and four of 500 sequences were assigned to the family “Ribosomal protein L5”, which is indicated by IPR-number IPR002132. The IPR-numbers are more specific and detailed entries than the GO-IDs. They provide a family or a domain to a certain protein. All IPR-numbers with two or more appearances in the output file of InterProScan run were listed in Table B.2 (appendix p. 33) and Table B.3 (appendix p. 34).

5.4.2 RaptorX

To interpret the output of the RaptorX tool, the frequency on the three different structures is counted and reported for each sequence. The loop structure appeared at minimum five times in all of the 247 different sequences of amino acids. There were 10 sequences without an alpha helix structure and 30 sequences without a beta strand structure. All details about the amount of these structures per sequence are listed in Table B.4 (appendix p. 35).
6 Discussion

6.1 Basic Hadoop

6.1.1 HDFS - replication and blocksize

An important idea behind the worker-node is locality of the data and the processing of the same data. It is important to process the data on the node where it resides. The locality is a big benefit for I/O-intensive applications. In best practices [11], the input data is split in 64 MB or 128 MB blocks. All of these blocks are replicated over three worker-nodes for fault tolerance.

In our case, the input size is much smaller than in usual use-cases: A 2000 sequence file uses about 1 MB of disk space and need 2.5 hours to process it with InterProScan on our 10node cluster.

In this situation Hadoop allows us the following options to tune I/O access time:

- Use a very small blocksize (around a few kilobytes) which results that our file is split into multiple blocks and those blocks could be local processed.

- Replicate our data to all nodes to guarantee that tasks would be scheduled on all nodes equal at the beginning of the Hadoop Streaming job.

We decided to not consider I/O access time. The time for the TCP connection to send a few characters and metadata in relation to the processing time was very small. The jobs were much more CPU-intensive then I/O-intensive.

6.1.2 CPU capacity utilization

During our research and experiments we had a lot of timeouts from the task-node. First we optimized our cluster for maximum performance. All four cores on our machines were used. The nodes had four tasks with 100 percent CPU usage. The task-tracker could not respond to heartbeats from the management node which resulted in the worker-node being marked as offline/unreachable.

In default-settings of IBM BigInsights, the formula is “quantity of cores - 1” for the quantity of maximum mapping tasks. We ran one task per node with three cores (from available four).

6.2 Hadoop Streaming framework

6.2.1 Streaming mnnerism

The Hadoop Streaming framework allows to execute an external executable as a MapReduce job. Negative aspects of this option are that the executable has to be distributed to all worker-nodes. For a small executable such as the UNIX cat-command, the parameter -file= exists which copies the file to the worker-node. In our case that was inappropriate because our tools (RaptorX and InterProScan) using libraries in background with many thousands of files (totally a site of approximately 40 gigabytes).

A speciality of the Streaming framework is to provide the input data as stdin and receive the output of the executable as stdout. To receive the data from input stream is unhandy for executable which
uses a file for their calculations. Both of our used classifiers needed a file, because they could process different algorithms in background which read the file many times. That made a virtual file also impossible, because a virtual file based on a stream could only be read once. At second reading the data would be different to the former reading. To meet our requirements we had to pack our executable in shell scripts. In the beginning of the script we processed the input stream and wrote it to a temporary file, which was then used by the executable, and then the output file piped to stdout.

6.2.2 The Streaming adaptability

Hadoop Streaming allowed us to use all available Hadoop job parameters for our task. We used a few in our jobs such as “-D mapred.map.tasks=500”. This was necessary because Hadoop Streaming has a default value of two map tasks. It is not the idea of Hadoop to generate large map tasks. For example: With a 500 sequence input file two nodes are calculating 250 sequence as one map task. Hadoop likes small tasks which can be redistributed to another node at failure. As a result, we ran at a minimum as many map tasks as sequences on our input file.

6.2.3 Failure handling

Hadoop has a primary viewpoint of failure handling. A failure does not mean a complete job-failure. It tells the management-node that the worker-node was unable to complete the given task and the management-node distributes the task to another worker-node. If a failure occurs four times in a row, the task is killed but the job continues. If a task contains bad input, the task will fail every time it is executed. Executing it on a different node has no other result. It will fail again and again. There are some possibilities to tell Hadoop that a non zero return code of the executable has not to be marked as failed attempt. One is to set the property “mapred.max.map.failures.percent”, which is unfortunately not supported by the Hadoop version (Hadoop 1.0.3) we used. The second option is to set the property “stream.non.zero.exit.is.failure=false” to ignore a non zero return code. We had to change this property dependent on the task. It was set to false only for the RaptorX runs. We also decremented the number of maximum task attempts “mapred.map.max.attempts” to a value of two because one bad input sequence could easily generate a useless server-run of about one hour when using RaptorX.

6.3 Conclusion

At the beginning, we were able to obtain initial success within short time. The good documentation of Hadoop Streaming let us take the first steps quickly. Without completely understanding the Hadoop ecosystem, we created some simple MapReduce jobs. Based on the information of the job tracker site, we gained insight into the complete task-handling and execution of MapReduce jobs. This disclosed the important facts about task trackers, datasets and the capabilities of the cluster. During our experiments, IBM Biginsights facilitated the distribution of cluster related settings. The benefit of Hadoop is a flexible configuration of parameters to fit the different needs of use cases. In summary, there are about 160 available properties to configure for the cluster and its dependencies. Some of the default configurations were inaccurate for our usage. Nevertheless, setting the correct parameters was difficult and had to be based on experience [1]. We think it will going to be a time consuming task to find best practice values for other use-cases. We conclude it is not a considered task when attempting to install and use a Big Data system like Hadoop. Although Hadoop is discussed in a lot communities [16] [9], there are only a few statements about best practices [13] [2]. We expect that the increasing use of Hadoop in real production environments will improve knowledge about best practices.
Appendix A
Program listings and bash scripts

Listing (A.1) C++ Programm FASTA2oneLine

```cpp
#include <iostream>
using namespace std;

int main() {
    std::string s{};
    int c = 0;
    while (getline(std::cin, s)) {
        if (s.front() == '>') {
            if (c != 0)
                std::cout << std::endl;
            else
                c = 1;
            std::cout << s << '%';
        } else {
            std::cout << s;
        }
    }
    return 0;
}
```

Listing (A.1) C++ Programm FASTA2oneLine

```bash
#!/bin/bash
# aufruf mit cat file | ./raptorX-run.sh >> output.csv

# 1.) read stdin
# 2.) for each sequenz save ID and store as $seqID, write ID to stdout as csv-detail-line
# 3.) create a file with the sequence in FASTA format in temp folder
# 4.) run raptorx, and filter output and writes out for csv-detail-line

#echo "seqID, H-value-count, E-value-count, L-value-count"
while read sequence
do
    # 2. Sequenz ID
    idx=`expr index "$sequence" " "`
    seqID=${sequence:1:$idx-2}
    seqID=${seqID//\|/-}
    echo -n "$seqID,"
    # 3. create FASTA file
    OIFS=$IFS
```
Appendix A Program listings and bash scripts

Listing (A.2) Bash-script RaptorX run

```bash
#!/bin/bash
#process last part of .tgt file from raptorX
#filters secondary structure information
#and counts the occurrence of H, E and L values and write it to STDOUT as csv-detail-line

swDO=''
countH=0
countE=0
countL=0

while read line
do
echo $line | grep -q "^/////////// Original SS3+SS8+ACC file"
if [ $? -eq 0 ]
  then
    swDO='1'
    read line
    read line
    read line
  fi
if [ "$swDO" = "1" ]
  then
    if [ "$line:0:1" != "#" ]
      then
        Hvalue=${line:0:5}
        Evalue=${line:6:5}
        Lvalue=${line:12:5}

        compare_result=`echo "\$Hvalue > $Evalue" | bc`
        if [ $compare_result -eq 1 ]
          then
            compare_result=`echo "\$Hvalue > $Lvalue" | bc`
```
if [ $compare_result -eq 1 ]
then
  countH=`expr $countH + 1`
else
  countL=`expr $countL + 1`
fi
else # Hvalue < Evalue
  compare_result=`echo "$Evalue > $Lvalue" | bc`
  if [ $compare_result -eq 1 ]
    then
      countE=`expr $countE + 1`
    else
      countL=`expr $countL + 1`
    fi
  else
    swDO=''
  fi
fi
done < $1
echo "$countH","$countE","$countL"

Listing (A.3) Bash-script RaptorX output filter script

#!/bin/bash
# 1.) read stdin
# 2.) for each sequenz save ID and store as $seqID, write ID to stdout as csv-detail-line
# 3.) create a file with the sequence in FASTA format in temp folder
# 4.) run interproscan, filter output and writes out for csv-detail-line
while read sequence
do
  # 2. Sequenz ID
  idx=`expr index "$sequence" " "`
  seqID=${sequence:1:$idx-2}
  seqID=$(echo "$seqID" | tr '|' '-')
  echo -n "$seqID,"
  # 3. create FASTA file
  OIFS=$IFS
  IFS=''
  arr2=$sequence
  count=0
  for x in $arr2
  do
    if [ $count -eq 0 ]
      then
        echo "$x" > /opt/temp/$seqID
      else
        echo "$x"| sed -e "s/\{60\}"/\n/g" >> /opt/temp/$seqID
      fi
    count=`expr $count + 1`
  done
  IFS=$OIFS
# 4. interproscan mit seqID file aufrufen

cd /opt/interproscan-5-RC5/

echo "$seqID" >> /opt/temp/interproscan_log

done

---

Listing (A.4) Bash-script InterProScan run

---

#!/bin/bash

# search all GO-IDs and IPR-IDs and write them to STDOUT

while read line
do
echo -n $line | grep -o -e "GO:[0-9]+" -e "IPR[0-9]+" | tr \n, done
echo -e "\n"

Listing (A.5) Bash-script InterProScan output filter script
Appendix B

Streaming job output files
<table>
<thead>
<tr>
<th>GO-ID</th>
<th>Occ.</th>
<th>Term name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0005524</td>
<td>62</td>
<td>ATP binding</td>
</tr>
<tr>
<td>GO:0005622</td>
<td>45</td>
<td>intracellular</td>
</tr>
<tr>
<td>GO:003735</td>
<td>42</td>
<td>structural constituent of ribosome</td>
</tr>
<tr>
<td>GO:0006412</td>
<td>42</td>
<td>translation</td>
</tr>
<tr>
<td>GO:005840</td>
<td>41</td>
<td>ribosome</td>
</tr>
<tr>
<td>GO:0016020</td>
<td>35</td>
<td>membrane</td>
</tr>
<tr>
<td>GO:005737</td>
<td>34</td>
<td>cytoplasm</td>
</tr>
<tr>
<td>GO:005114</td>
<td>33</td>
<td>oxidation-reduction process</td>
</tr>
<tr>
<td>GO:003677</td>
<td>26</td>
<td>DNA binding</td>
</tr>
<tr>
<td>GO:005975</td>
<td>22</td>
<td>carbohydrate metabolic process</td>
</tr>
<tr>
<td>GO:003824</td>
<td>19</td>
<td>catalytic activity</td>
</tr>
<tr>
<td>GO:0016021</td>
<td>19</td>
<td>integral to membrane</td>
</tr>
<tr>
<td>GO:003723</td>
<td>18</td>
<td>RNA binding</td>
</tr>
<tr>
<td>GO:0006351</td>
<td>17</td>
<td>transcription, DNA-dependent</td>
</tr>
<tr>
<td>GO:003899</td>
<td>16</td>
<td>DNA-directed RNA polymerase activity</td>
</tr>
<tr>
<td>GO:005525</td>
<td>16</td>
<td>GTP binding</td>
</tr>
<tr>
<td>GO:008152</td>
<td>16</td>
<td>metabolic process</td>
</tr>
<tr>
<td>GO:005515</td>
<td>14</td>
<td>protein binding</td>
</tr>
<tr>
<td>GO:006355</td>
<td>12</td>
<td>regulation of transcription, DNA-dependent</td>
</tr>
<tr>
<td>GO:000166</td>
<td>11</td>
<td>nucleotide binding</td>
</tr>
<tr>
<td>GO:006508</td>
<td>10</td>
<td>proteolysis</td>
</tr>
<tr>
<td>GO:004812</td>
<td>9</td>
<td>aminoacyl-tRNA ligase activity</td>
</tr>
<tr>
<td>GO:008270</td>
<td>9</td>
<td>zinc ion binding</td>
</tr>
<tr>
<td>GO:016491</td>
<td>9</td>
<td>oxidoreductase activity</td>
</tr>
<tr>
<td>GO:017038</td>
<td>9</td>
<td>protein import</td>
</tr>
<tr>
<td>GO:051536</td>
<td>9</td>
<td>iron-sulfur cluster binding</td>
</tr>
<tr>
<td>GO:003700</td>
<td>8</td>
<td>sequence-specific DNA binding transcription factor activity</td>
</tr>
<tr>
<td>GO:009058</td>
<td>8</td>
<td>biosynthetic process</td>
</tr>
<tr>
<td>GO:016868</td>
<td>8</td>
<td>intramolecular transferase activity, phosphotransferases</td>
</tr>
<tr>
<td>GO:006520</td>
<td>7</td>
<td>cellular amino acid metabolic process</td>
</tr>
<tr>
<td>GO:009055</td>
<td>7</td>
<td>electron carrier activity</td>
</tr>
<tr>
<td>GO:0016787</td>
<td>7</td>
<td>hydrolase activity</td>
</tr>
<tr>
<td>GO:0000287</td>
<td>6</td>
<td>magnesium ion binding</td>
</tr>
<tr>
<td>GO:0003924</td>
<td>6</td>
<td>GTPase activity</td>
</tr>
<tr>
<td>GO:0008137</td>
<td>6</td>
<td>NADH dehydrogenase (ubiquinone) activity</td>
</tr>
<tr>
<td>GO:0019843</td>
<td>6</td>
<td>rRNA binding</td>
</tr>
<tr>
<td>GO:0046872</td>
<td>6</td>
<td>metal ion binding</td>
</tr>
<tr>
<td>GO:0046983</td>
<td>6</td>
<td>protein dimerization activity</td>
</tr>
<tr>
<td>GO:0003676</td>
<td>5</td>
<td>nucleic acid binding</td>
</tr>
<tr>
<td>GO:003968</td>
<td>5</td>
<td>RNA-directed RNA polymerase activity</td>
</tr>
<tr>
<td>GO:0005198</td>
<td>5</td>
<td>structural molecule activity</td>
</tr>
<tr>
<td>GO:0005506</td>
<td>5</td>
<td>iron ion binding</td>
</tr>
<tr>
<td>GO:0005576</td>
<td>5</td>
<td>extracellular region</td>
</tr>
<tr>
<td>GO:0006164</td>
<td>5</td>
<td>purine nucleotide biosynthetic process</td>
</tr>
<tr>
<td>GO:0006415</td>
<td>5</td>
<td>translational termination</td>
</tr>
<tr>
<td>GO:0006418</td>
<td>5</td>
<td>tRNA aminoacylation for protein translation</td>
</tr>
<tr>
<td>GO:0006810</td>
<td>5</td>
<td>transport</td>
</tr>
<tr>
<td>GO:0007165</td>
<td>5</td>
<td>signal transduction</td>
</tr>
<tr>
<td>GO:0008033</td>
<td>5</td>
<td>tRNA processing</td>
</tr>
<tr>
<td>GO:0015078</td>
<td>5</td>
<td>hydrogen ion transmembrane transporter activity</td>
</tr>
<tr>
<td>GO:0016620</td>
<td>5</td>
<td>oxidoreductase activity, acting on the aldehyde or oxo group of donors, NAD or NADP as acceptor</td>
</tr>
<tr>
<td>GO:0016740</td>
<td>5</td>
<td>transferase activity</td>
</tr>
<tr>
<td>GO:0016874</td>
<td>5</td>
<td>ligase activity</td>
</tr>
<tr>
<td>GO:0024773</td>
<td>5</td>
<td>ATP synthesis coupled electron transport</td>
</tr>
<tr>
<td>GO:0043039</td>
<td>5</td>
<td>tRNA aminoclylation</td>
</tr>
<tr>
<td>GO:0050585</td>
<td>5</td>
<td>transmembrane transport</td>
</tr>
<tr>
<td>GO:00041197</td>
<td>4</td>
<td>cysteine-type endopeptidase activity</td>
</tr>
<tr>
<td>GO:004222</td>
<td>4</td>
<td>metalloendopeptidase activity</td>
</tr>
<tr>
<td>GO:0005215</td>
<td>4</td>
<td>transporter activity</td>
</tr>
<tr>
<td>GO:006396</td>
<td>4</td>
<td>metalloendopeptidase activity</td>
</tr>
<tr>
<td>GO:006457</td>
<td>4</td>
<td>protein folding</td>
</tr>
<tr>
<td>GO:010181</td>
<td>4</td>
<td>FMN binding</td>
</tr>
<tr>
<td>GO:0015986</td>
<td>4</td>
<td>ATP synthesis coupled proton transport</td>
</tr>
<tr>
<td>GO:0015991</td>
<td>4</td>
<td>ATP hydrolysis coupled proton transport</td>
</tr>
<tr>
<td>GO:0016114</td>
<td>4</td>
<td>terpenoid biosynthetic process</td>
</tr>
<tr>
<td>GO:0016743</td>
<td>4</td>
<td>carboxyl- or carbamoyltransferase activity</td>
</tr>
<tr>
<td>------------</td>
<td>---</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>GO:0016773</td>
<td>4</td>
<td>phosphotransferase activity, alcohol group as acceptor</td>
</tr>
<tr>
<td>GO:0016887</td>
<td>4</td>
<td>ATPase activity</td>
</tr>
<tr>
<td>GO:0043565</td>
<td>4</td>
<td>sequence-specific DNA binding</td>
</tr>
<tr>
<td>GO:0000105</td>
<td>3</td>
<td>histidine biosynthetic process</td>
</tr>
<tr>
<td>GO:0003747</td>
<td>3</td>
<td>translation release factor activity</td>
</tr>
<tr>
<td>GO:0003774</td>
<td>3</td>
<td>motor activity</td>
</tr>
<tr>
<td>GO:0004019</td>
<td>3</td>
<td>adenylosuccinate synthase activity</td>
</tr>
<tr>
<td>GO:0004252</td>
<td>3</td>
<td>serine-type endopeptidase activity</td>
</tr>
<tr>
<td>GO:0004814</td>
<td>3</td>
<td>arginine-tRNA ligase activity</td>
</tr>
<tr>
<td>GO:0006139</td>
<td>3</td>
<td>nucleobase-containing compound metabolic process</td>
</tr>
<tr>
<td>GO:0006364</td>
<td>3</td>
<td>rRNA processing</td>
</tr>
<tr>
<td>GO:0006420</td>
<td>3</td>
<td>nucleobase-containing compound metabolic process</td>
</tr>
<tr>
<td>GO:0006526</td>
<td>3</td>
<td>arginine biosynthetic process</td>
</tr>
<tr>
<td>GO:0008685</td>
<td>3</td>
<td>2-C-methyl-D-erythritol 2,4-cyclodiphosphate synthase activity</td>
</tr>
<tr>
<td>GO:0009405</td>
<td>3</td>
<td>pathogenesis</td>
</tr>
<tr>
<td>GO:0016597</td>
<td>3</td>
<td>amino acid binding</td>
</tr>
<tr>
<td>GO:0016616</td>
<td>3</td>
<td>oxidoreductase activity, acting on the CH-OH group of donors, NAD or NADP as acceptor</td>
</tr>
<tr>
<td>GO:0016829</td>
<td>3</td>
<td>lyase activity</td>
</tr>
<tr>
<td>GO:0016836</td>
<td>3</td>
<td>ligase activity, forming aminoacyl-tRNA and related compounds</td>
</tr>
<tr>
<td>GO:0017111</td>
<td>3</td>
<td>nucleoside-triphosphatase activity</td>
</tr>
<tr>
<td>GO:0019028</td>
<td>3</td>
<td>viral capsid</td>
</tr>
<tr>
<td>GO:0020037</td>
<td>3</td>
<td>heme binding</td>
</tr>
<tr>
<td>GO:0030976</td>
<td>3</td>
<td>thiamine pyrophosphate binding</td>
</tr>
<tr>
<td>GO:0045263</td>
<td>3</td>
<td>proton-transporting ATP synthase complex, coupling factor F(o)</td>
</tr>
<tr>
<td>GO:0046080</td>
<td>3</td>
<td>dUTP metabolic process</td>
</tr>
<tr>
<td>GO:0051287</td>
<td>3</td>
<td>NAD binding</td>
</tr>
</tbody>
</table>
Table (B.2) Table of IPR-number and their Domain

<table>
<thead>
<tr>
<th>IPR-number</th>
<th>Occ.</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPR002942</td>
<td>5</td>
<td>RNA-binding S4 domain</td>
</tr>
<tr>
<td>IPR001750</td>
<td>4</td>
<td>NADH:ubiquinone/plastoquinone oxidoreductase</td>
</tr>
<tr>
<td>IPR001912</td>
<td>4</td>
<td>Ribosomal protein S4/S9, N-terminal</td>
</tr>
<tr>
<td>IPR003439</td>
<td>4</td>
<td>ABC transporter-like</td>
</tr>
<tr>
<td>IPR000352</td>
<td>3</td>
<td>Peptide chain release factor class I/class II</td>
</tr>
<tr>
<td>IPR000640</td>
<td>3</td>
<td>Translation elongation factor EFG, V domain</td>
</tr>
<tr>
<td>IPR000795</td>
<td>3</td>
<td>Elongation factor, GTP-binding domain</td>
</tr>
<tr>
<td>IPR001450</td>
<td>3</td>
<td>4Fe-4S binding domain</td>
</tr>
<tr>
<td>IPR003526</td>
<td>3</td>
<td>2-C-methyl-D-erythritol 2,4-cyclodiphosphate synthase</td>
</tr>
<tr>
<td>IPR004161</td>
<td>3</td>
<td>Translation elongation factor EFTu/EFTa, domain 2</td>
</tr>
<tr>
<td>IPR005824</td>
<td>3</td>
<td>KOW</td>
</tr>
<tr>
<td>IPR006073</td>
<td>3</td>
<td>GTP binding domain</td>
</tr>
<tr>
<td>IPR011115</td>
<td>3</td>
<td>SecA DEAD-like, N-terminal</td>
</tr>
<tr>
<td>IPR011116</td>
<td>3</td>
<td>SecA Wing/Scaffold</td>
</tr>
<tr>
<td>IPR011130</td>
<td>3</td>
<td>Preprotein, cross-linking domain</td>
</tr>
<tr>
<td>IPR012947</td>
<td>3</td>
<td>Threonyl/alanyl tRNA synthetase, SAD</td>
</tr>
<tr>
<td>IPR013842</td>
<td>3</td>
<td>GTP-binding protein LepA, C-terminal</td>
</tr>
<tr>
<td>IPR018449</td>
<td>3</td>
<td>NIL domain</td>
</tr>
<tr>
<td>IPR020069</td>
<td>3</td>
<td>Ribosomal protein L9, C-terminal</td>
</tr>
<tr>
<td>IPR020070</td>
<td>3</td>
<td>Ribosomal protein L9, N-terminal</td>
</tr>
<tr>
<td>IPR000022</td>
<td>2</td>
<td>Carboxyl transferase</td>
</tr>
<tr>
<td>IPR000073</td>
<td>2</td>
<td>Alpha/beta hydrolase fold-1</td>
</tr>
<tr>
<td>IPR000157</td>
<td>2</td>
<td>Toll/interleukin-1 receptor homology (TIR) domain</td>
</tr>
<tr>
<td>IPR000194</td>
<td>2</td>
<td>ATPase, F1/V1/A1 complex, alpha/beta subunit,nucleotide-binding domain</td>
</tr>
<tr>
<td>IPR000305</td>
<td>2</td>
<td>GIY-YIG nuclease superfamily</td>
</tr>
<tr>
<td>IPR000534</td>
<td>2</td>
<td>Semialdehyde dehydrogenase, NAD-binding</td>
</tr>
<tr>
<td>IPR000713</td>
<td>2</td>
<td>Mur ligase, N-terminal</td>
</tr>
<tr>
<td>IPR000722</td>
<td>2</td>
<td>RNA polymerase, alpha subunit</td>
</tr>
<tr>
<td>IPR000793</td>
<td>2</td>
<td>ATPase, F1/V1/A1 complex, alpha/beta subunit, C-terminal</td>
</tr>
<tr>
<td>IPR000836</td>
<td>2</td>
<td>Phosphoribosyltransferase domain</td>
</tr>
<tr>
<td>IPR01347</td>
<td>2</td>
<td>Sugar isomerase (SIS)</td>
</tr>
<tr>
<td>IPR012413</td>
<td>3</td>
<td>Phenylalanyl-tRNA synthetase</td>
</tr>
<tr>
<td>IPR003594</td>
<td>2</td>
<td>V-ATPase proteolipid subunit C-like domain</td>
</tr>
<tr>
<td>IPR003594</td>
<td>2</td>
<td>Histidine kinase-like ATPase, ATP-binding domain</td>
</tr>
<tr>
<td>IPR003661</td>
<td>2</td>
<td>Signal transduction histidine kinase</td>
</tr>
<tr>
<td>IPR004088</td>
<td>2</td>
<td>K Homology domain, type 1</td>
</tr>
<tr>
<td>IPR004100</td>
<td>2</td>
<td>ATPase, alpha/beta subunit, N-terminal</td>
</tr>
<tr>
<td>IPR004101</td>
<td>2</td>
<td>Mur ligase, C-terminal</td>
</tr>
<tr>
<td>IPR014115</td>
<td>2</td>
<td>GAD domain</td>
</tr>
<tr>
<td>IPR004364</td>
<td>2</td>
<td>Aminoacyl-tRNA synthetase, class II (D/K/N)</td>
</tr>
<tr>
<td>IPR004365</td>
<td>2</td>
<td>Nuclease acid binding, OB-fold, tRNA/helicase-type</td>
</tr>
<tr>
<td>IPR005139</td>
<td>2</td>
<td>Peptide chain release factor</td>
</tr>
<tr>
<td>IPR005798</td>
<td>2</td>
<td>Cytochrome b/b6, C-terminal</td>
</tr>
<tr>
<td>IPR005843</td>
<td>2</td>
<td>Alpha-D-phosphohexomutase, C-terminal</td>
</tr>
<tr>
<td>IPR005844</td>
<td>2</td>
<td>Alpha-D-phosphohexomutase, alpha/beta/alpha domain I</td>
</tr>
</tbody>
</table>
Table (B.3) Table of IPR-number and their Family

<table>
<thead>
<tr>
<th>IPR-number</th>
<th>Occ.</th>
<th>Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPR002132</td>
<td>4</td>
<td>Ribosomal protein L5</td>
</tr>
<tr>
<td>IPR000630</td>
<td>3</td>
<td>Ribosomal protein S8</td>
</tr>
<tr>
<td>IPR001114</td>
<td>3</td>
<td>Adenylosuccinate synthetase</td>
</tr>
<tr>
<td>IPR001705</td>
<td>3</td>
<td>Ribosomal protein L33</td>
</tr>
<tr>
<td>IPR002501</td>
<td>3</td>
<td>Pseudouridine synthase II</td>
</tr>
<tr>
<td>IPR003509</td>
<td>3</td>
<td>Uncharacterised protein family UPF0102</td>
</tr>
<tr>
<td>IPR004506</td>
<td>3</td>
<td>tRNA-specific 2-thiouridylase</td>
</tr>
<tr>
<td>IPR008180</td>
<td>3</td>
<td>DeoxyUTP pyrophosphatase</td>
</tr>
<tr>
<td>IPR000146</td>
<td>2</td>
<td>Fructose-1,6-bisphosphatase class 1/Sedoheputulose-1,7-bisphosphatase</td>
</tr>
<tr>
<td>IPR000276</td>
<td>2</td>
<td>G protein-coupled receptor, rhodopsin-like</td>
</tr>
<tr>
<td>IPR000850</td>
<td>2</td>
<td>Adenylate kinase</td>
</tr>
<tr>
<td>IPR001015</td>
<td>2</td>
<td>Ferrocnelatase</td>
</tr>
<tr>
<td>IPR001063</td>
<td>2</td>
<td>Ribosomal protein L22/L17</td>
</tr>
<tr>
<td>IPR001209</td>
<td>2</td>
<td>Ribosomal protein S14</td>
</tr>
<tr>
<td>IPR001518</td>
<td>2</td>
<td>Argininosuccinate synthase</td>
</tr>
<tr>
<td>IPR001585</td>
<td>2</td>
<td>Transaldolase</td>
</tr>
<tr>
<td>IPR001763</td>
<td>2</td>
<td>Rhodanese-like domain</td>
</tr>
<tr>
<td>-----------</td>
<td>---</td>
<td>-----------------------</td>
</tr>
<tr>
<td>IPR001872</td>
<td>2</td>
<td>Peptidase A8, signal peptidase II</td>
</tr>
<tr>
<td>IPR002146</td>
<td>2</td>
<td>ATPase, F0 complex, subunit B/B', bacterial/chloroplast</td>
</tr>
<tr>
<td>IPR002180</td>
<td>2</td>
<td>6,7-dimethyl-8-ribityllumazine synthase</td>
</tr>
<tr>
<td>IPR002220</td>
<td>2</td>
<td>Dihydropicolinate synthetase-like</td>
</tr>
<tr>
<td>IPR002305</td>
<td>2</td>
<td>Aminoacyl-tRNA synthetase, class Ic</td>
</tr>
<tr>
<td>IPR002423</td>
<td>2</td>
<td>Chaperonin Cpn60/TCP-1</td>
</tr>
<tr>
<td>IPR004214</td>
<td>2</td>
<td>Conotoxin</td>
</tr>
<tr>
<td>IPR004401</td>
<td>2</td>
<td>Nucleoid-associated protein YbaB</td>
</tr>
<tr>
<td>IPR004881</td>
<td>2</td>
<td>Ribosome biogenesis GTPase RsgA, putative</td>
</tr>
<tr>
<td>IPR006032</td>
<td>2</td>
<td>Ribosomal protein S12/S23</td>
</tr>
<tr>
<td>IPR007457</td>
<td>2</td>
<td>Fe(II) trafficking protein YggX</td>
</tr>
<tr>
<td>IPR008735</td>
<td>2</td>
<td>Beta-microseminoprotein</td>
</tr>
<tr>
<td>IPR013025</td>
<td>2</td>
<td>Ribosomal protein L25/L23</td>
</tr>
<tr>
<td>IPR018317</td>
<td>2</td>
<td>Queuosine biosynthesis protein QueC</td>
</tr>
</tbody>
</table>

### Table (B.4) Secondary structure prediction

<table>
<thead>
<tr>
<th>sequence ID</th>
<th>alpha helix</th>
<th>beta strand</th>
<th>loops</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp A0JX91 RS4 ARTS2</td>
<td>71</td>
<td>27</td>
<td>110</td>
</tr>
<tr>
<td>sp A0RRG4 END4 CAMFF</td>
<td>103</td>
<td>27</td>
<td>152</td>
</tr>
<tr>
<td>sp A1AFE8 FETP ECOK1</td>
<td>44</td>
<td>1</td>
<td>46</td>
</tr>
<tr>
<td>sp A1B9H4 RL27 PARDP</td>
<td>0</td>
<td>36</td>
<td>54</td>
</tr>
<tr>
<td>sp A1JP96 XNI YERE8</td>
<td>109</td>
<td>25</td>
<td>117</td>
</tr>
<tr>
<td>sp A1RED3 RL15 SHESW</td>
<td>21</td>
<td>27</td>
<td>96</td>
</tr>
<tr>
<td>sp A1RT13 SYA PYRIL</td>
<td>328</td>
<td>134</td>
<td>430</td>
</tr>
<tr>
<td>sp A1UZV8 COXX BURMS</td>
<td>212</td>
<td>2</td>
<td>86</td>
</tr>
<tr>
<td>sp A1VER5 MNME DESVV</td>
<td>195</td>
<td>67</td>
<td>195</td>
</tr>
<tr>
<td>sp A1VYR0 EFP CAMJJ</td>
<td>8</td>
<td>87</td>
<td>94</td>
</tr>
<tr>
<td>sp A2AKB9 DCA10 MOUSE</td>
<td>0</td>
<td>202</td>
<td>364</td>
</tr>
<tr>
<td>sp A2BUD7 NU1C PROM5</td>
<td>257</td>
<td>6</td>
<td>109</td>
</tr>
<tr>
<td>sp A3Q6V2 RL332 MYCSJ</td>
<td>0</td>
<td>24</td>
<td>30</td>
</tr>
<tr>
<td>sp A4FZ98 COFD METM5</td>
<td>96</td>
<td>56</td>
<td>157</td>
</tr>
<tr>
<td>sp A4IT5 OLM2A XENTR</td>
<td>107</td>
<td>122</td>
<td>413</td>
</tr>
<tr>
<td>sp A4QKC9 RR12 BARVE</td>
<td>7</td>
<td>36</td>
<td>80</td>
</tr>
<tr>
<td>sp A4W778 QUEA ENT38</td>
<td>65</td>
<td>78</td>
<td>213</td>
</tr>
<tr>
<td>sp A4WD10 MNTH ENT38</td>
<td>311</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>sp A4XHM5 RiSB CALS8</td>
<td>74</td>
<td>25</td>
<td>57</td>
</tr>
<tr>
<td>sp A4YXQ6 SYA BRASO</td>
<td>338</td>
<td>126</td>
<td>427</td>
</tr>
<tr>
<td>sp A5F5P1 SECA VIBC3</td>
<td>459</td>
<td>52</td>
<td>392</td>
</tr>
<tr>
<td>sp A5G6H1 LPXK GEOUR</td>
<td>108</td>
<td>61</td>
<td>191</td>
</tr>
<tr>
<td>sp A5UBA3 Y3135 HAEIE</td>
<td>304</td>
<td>61</td>
<td>186</td>
</tr>
<tr>
<td>sp A6L0A5 PNP BACV8</td>
<td>211</td>
<td>131</td>
<td>369</td>
</tr>
<tr>
<td>sp A6LPU7 RS9 CLOB8</td>
<td>45</td>
<td>20</td>
<td>65</td>
</tr>
<tr>
<td>sp A6VKC6 RL7 ACTSZ</td>
<td>59</td>
<td>11</td>
<td>52</td>
</tr>
<tr>
<td>sp A7FME9 TAL YERP3</td>
<td>165</td>
<td>22</td>
<td>130</td>
</tr>
<tr>
<td>sp A7MFI9 TIG CROS8</td>
<td>201</td>
<td>64</td>
<td>167</td>
</tr>
<tr>
<td>File Name</td>
<td>Spots</td>
<td>Job</td>
<td>Output</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td>A7VN14 MSMB2 PROFL</td>
<td>15</td>
<td></td>
<td>23 71</td>
</tr>
<tr>
<td>A7X3H9 Y1729 STAA1</td>
<td>105</td>
<td></td>
<td>64 116</td>
</tr>
<tr>
<td>A7JJ30 MOAA ECO24</td>
<td>99</td>
<td></td>
<td>36 194</td>
</tr>
<tr>
<td>A8ES24 HIS4 ARCB4</td>
<td>79</td>
<td></td>
<td>52 104</td>
</tr>
<tr>
<td>A8H1E5 HCP SHEPA</td>
<td>254</td>
<td></td>
<td>40 260</td>
</tr>
<tr>
<td>A8HS90 NDK AZOC5</td>
<td>59</td>
<td></td>
<td>18 63</td>
</tr>
<tr>
<td>A8KZ0F PDXT FRASN</td>
<td>60</td>
<td></td>
<td>49 94</td>
</tr>
<tr>
<td>A8SEC9 PSBB CERDE</td>
<td>149</td>
<td></td>
<td>88 271</td>
</tr>
<tr>
<td>A8Z4G7 MURC STAAT</td>
<td>149</td>
<td></td>
<td>94 194</td>
</tr>
<tr>
<td>A9A514 RL24 NITMS</td>
<td>33</td>
<td></td>
<td>37 98</td>
</tr>
<tr>
<td>A9LYX9 F16PA NEIM0</td>
<td>95</td>
<td></td>
<td>70 159</td>
</tr>
<tr>
<td>A9MEJI PURR SALAR</td>
<td>140</td>
<td></td>
<td>50 151</td>
</tr>
<tr>
<td>B0VBY5 MNMA ACIBY</td>
<td>80</td>
<td></td>
<td>96 201</td>
</tr>
<tr>
<td>B1HMT3 GLMM LYSSC</td>
<td>167</td>
<td></td>
<td>63 220</td>
</tr>
<tr>
<td>B1I956 MRAY STRP1</td>
<td>203</td>
<td></td>
<td>19 104</td>
</tr>
<tr>
<td>B1II34 RSGA CLOBK</td>
<td>75</td>
<td></td>
<td>69 148</td>
</tr>
<tr>
<td>B2A4C3 RL33 NATTJ</td>
<td>0</td>
<td></td>
<td>20 29</td>
</tr>
<tr>
<td>B2VJA4 PURT ERWT9</td>
<td>125</td>
<td></td>
<td>88 179</td>
</tr>
<tr>
<td>B3Q5X2 RL20 RHOP3</td>
<td>90</td>
<td></td>
<td>0 29</td>
</tr>
<tr>
<td>B3R808 Y3032 CUPTR</td>
<td>35</td>
<td></td>
<td>33 63</td>
</tr>
<tr>
<td>B4T750 NANA SALNS</td>
<td>137</td>
<td></td>
<td>35 125</td>
</tr>
<tr>
<td>B5F4C2 YJHX SALA4</td>
<td>27</td>
<td></td>
<td>17 41</td>
</tr>
<tr>
<td>B5F514 ARGO SALA4</td>
<td>191</td>
<td></td>
<td>0 20</td>
</tr>
<tr>
<td>B5JK3 RL24 SALDC</td>
<td>0</td>
<td></td>
<td>49 55</td>
</tr>
<tr>
<td>B5XAM2 ICT1 SALSA</td>
<td>70</td>
<td></td>
<td>22 99</td>
</tr>
<tr>
<td>B5XJ20 PUR9 STRPZ</td>
<td>176</td>
<td></td>
<td>75 264</td>
</tr>
<tr>
<td>B5XJP1 NANE STRPZ</td>
<td>91</td>
<td></td>
<td>37 106</td>
</tr>
<tr>
<td>B5YH59 DNK THEYD</td>
<td>213</td>
<td></td>
<td>111 308</td>
</tr>
<tr>
<td>B7GJC8 UPPP ANOFW</td>
<td>199</td>
<td></td>
<td>0 74</td>
</tr>
<tr>
<td>B7H9V0 Y577 BACC4</td>
<td>97</td>
<td></td>
<td>38 104</td>
</tr>
<tr>
<td>B7J6C1 PUR7 ACT1F2</td>
<td>65</td>
<td></td>
<td>44 131</td>
</tr>
<tr>
<td>B7L649 RISB ECO55</td>
<td>72</td>
<td></td>
<td>26 58</td>
</tr>
<tr>
<td>B7M3B3 MTF3 ECO5A</td>
<td>110</td>
<td></td>
<td>17 138</td>
</tr>
<tr>
<td>B7VH36 Y2060 VIBSL</td>
<td>12</td>
<td></td>
<td>7 40</td>
</tr>
<tr>
<td>B8GVN4 RL9 CAUCN</td>
<td>63</td>
<td></td>
<td>42 89</td>
</tr>
<tr>
<td>B8HYL7 DAPA CYAP4</td>
<td>131</td>
<td></td>
<td>34 129</td>
</tr>
<tr>
<td>B8Z5A6 RS14Z MYCLB</td>
<td>29</td>
<td></td>
<td>0 32</td>
</tr>
<tr>
<td>B9LPW4 RONP HALLT</td>
<td>34</td>
<td></td>
<td>0 30</td>
</tr>
<tr>
<td>C0QB49 PURA DESAH</td>
<td>121</td>
<td></td>
<td>74 234</td>
</tr>
<tr>
<td>C0QMN7 RS8 PERMH</td>
<td>31</td>
<td></td>
<td>35 70</td>
</tr>
<tr>
<td>C0SDJ3 COQ4 PARBP</td>
<td>156</td>
<td></td>
<td>0 129</td>
</tr>
<tr>
<td>C1A051 ASSY ROHE4</td>
<td>154</td>
<td></td>
<td>68 177</td>
</tr>
<tr>
<td>C1CPRT RHN3 STRZT</td>
<td>126</td>
<td></td>
<td>46 121</td>
</tr>
<tr>
<td>C1ETQ0 Y4566 BACC3</td>
<td>99</td>
<td></td>
<td>35 91</td>
</tr>
<tr>
<td>C4ZRB8 PYRB ECOB</td>
<td>115</td>
<td></td>
<td>40 156</td>
</tr>
<tr>
<td>C50A83 K6PF ECOB</td>
<td>136</td>
<td></td>
<td>51 133</td>
</tr>
<tr>
<td>sp</td>
<td>C5CV58 DCD VARPS</td>
<td>12</td>
<td>71</td>
</tr>
<tr>
<td>----</td>
<td>------------------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>sp</td>
<td>E1WAB4 ORGB SALTS</td>
<td>152</td>
<td>22</td>
</tr>
<tr>
<td>sp</td>
<td>F7D4X9 SIR5 MONDO</td>
<td>91</td>
<td>29</td>
</tr>
<tr>
<td>sp</td>
<td>O22431 RL10 PINTA</td>
<td>43</td>
<td>44</td>
</tr>
<tr>
<td>sp</td>
<td>O23324 APS3 ARATH</td>
<td>145</td>
<td>67</td>
</tr>
<tr>
<td>sp</td>
<td>O24385 CPI7 SOLTU</td>
<td>0</td>
<td>66</td>
</tr>
<tr>
<td>sp</td>
<td>O31609 YJBK BACSU</td>
<td>49</td>
<td>61</td>
</tr>
<tr>
<td>sp</td>
<td>O43002 SC61B SCHPO</td>
<td>38</td>
<td>7</td>
</tr>
<tr>
<td>sp</td>
<td>O61069 KAD TRYBR</td>
<td>93</td>
<td>13</td>
</tr>
<tr>
<td>sp</td>
<td>O74759 NTO1 SCHOPO</td>
<td>228</td>
<td>27</td>
</tr>
<tr>
<td>sp</td>
<td>O79213 CYB OCETR</td>
<td>228</td>
<td>0</td>
</tr>
<tr>
<td>sp</td>
<td>O94225 HOSM PENCW</td>
<td>168</td>
<td>49</td>
</tr>
<tr>
<td>sp</td>
<td>P01667 KV3AF MOUSE</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>sp</td>
<td>P03330 GAG WMSV</td>
<td>202</td>
<td>4</td>
</tr>
<tr>
<td>sp</td>
<td>P09125 MSP8 EIMAC</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>sp</td>
<td>P0A6H9 CLS ECOL6</td>
<td>229</td>
<td>80</td>
</tr>
<tr>
<td>sp</td>
<td>P0A712 RF1 ECO57</td>
<td>177</td>
<td>29</td>
</tr>
<tr>
<td>sp</td>
<td>P0AA49 YFDV ECOLI</td>
<td>258</td>
<td>0</td>
</tr>
<tr>
<td>sp</td>
<td>P0AAS2 YLAC SHIFL</td>
<td>85</td>
<td>22</td>
</tr>
<tr>
<td>sp</td>
<td>P0ABB4 ATPB ECOLI</td>
<td>119</td>
<td>96</td>
</tr>
<tr>
<td>sp</td>
<td>P0C2Z9 ATPH ORYSA</td>
<td>69</td>
<td>0</td>
</tr>
<tr>
<td>sp</td>
<td>P0C5Q6 YM94A YEAST</td>
<td>17</td>
<td>28</td>
</tr>
<tr>
<td>sp</td>
<td>P0CG46 CKBR CONSL</td>
<td>54</td>
<td>0</td>
</tr>
<tr>
<td>sp</td>
<td>P0CP10 NAR1 CRYNJ</td>
<td>188</td>
<td>71</td>
</tr>
<tr>
<td>sp</td>
<td>P12138 CPNA1 ARATH</td>
<td>267</td>
<td>66</td>
</tr>
<tr>
<td>sp</td>
<td>P22087 FBRL HUMAN</td>
<td>42</td>
<td>70</td>
</tr>
<tr>
<td>sp</td>
<td>P25508 COCA1 BOVIN</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sp</td>
<td>P25779 CYSP TRYCR</td>
<td>112</td>
<td>57</td>
</tr>
<tr>
<td>sp</td>
<td>P27581 ADH2 DROAR</td>
<td>112</td>
<td>35</td>
</tr>
<tr>
<td>sp</td>
<td>P29363 THRC PSEAE</td>
<td>214</td>
<td>59</td>
</tr>
<tr>
<td>sp</td>
<td>P31552 CAIC ECOLI</td>
<td>160</td>
<td>86</td>
</tr>
<tr>
<td>sp</td>
<td>P36512 UDB13 RABIT</td>
<td>250</td>
<td>62</td>
</tr>
<tr>
<td>sp</td>
<td>P40219 OSW5 YEAST</td>
<td>83</td>
<td>2</td>
</tr>
<tr>
<td>sp</td>
<td>P42528 ARP3 DICDI</td>
<td>113</td>
<td>59</td>
</tr>
<tr>
<td>sp</td>
<td>P46950 SNG1 YEAST</td>
<td>266</td>
<td>30</td>
</tr>
<tr>
<td>sp</td>
<td>P46953 3HAO RAT</td>
<td>27</td>
<td>99</td>
</tr>
<tr>
<td>sp</td>
<td>P48377 RXL1 MOUSE</td>
<td>285</td>
<td>10</td>
</tr>
<tr>
<td>sp</td>
<td>P53573 ETFA BRAJA</td>
<td>91</td>
<td>76</td>
</tr>
<tr>
<td>sp</td>
<td>P53948 YNF7 YEAST</td>
<td>0</td>
<td>55</td>
</tr>
<tr>
<td>sp</td>
<td>P55919 CXCR1 GORGO</td>
<td>213</td>
<td>15</td>
</tr>
<tr>
<td>sp</td>
<td>Q72J71 VATF THET2</td>
<td>45</td>
<td>18</td>
</tr>
<tr>
<td>sp</td>
<td>P57087 JAM2 HUMAN</td>
<td>7</td>
<td>144</td>
</tr>
<tr>
<td>sp</td>
<td>P57103 NAC3 HUMAN</td>
<td>279</td>
<td>201</td>
</tr>
<tr>
<td>sp</td>
<td>P58063 TRUB CAUCR</td>
<td>61</td>
<td>79</td>
</tr>
<tr>
<td>sp</td>
<td>P59511 ATS20 MOUSE</td>
<td>93</td>
<td>385</td>
</tr>
<tr>
<td>Accession</td>
<td>Description</td>
<td>Length</td>
<td>Width</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>P61806</td>
<td>DAD1 MESAU</td>
<td>81</td>
<td>2</td>
</tr>
<tr>
<td>P76241</td>
<td>YEAM ECOLI</td>
<td>102</td>
<td>45</td>
</tr>
<tr>
<td>Q566C7</td>
<td>NUDT3 RAT</td>
<td>32</td>
<td>48</td>
</tr>
<tr>
<td>B4SJB0</td>
<td>GLYA STRM5</td>
<td>177</td>
<td>30</td>
</tr>
<tr>
<td>P83038</td>
<td>HDAC4 CHICK</td>
<td>378</td>
<td>33</td>
</tr>
<tr>
<td>P84391</td>
<td>UL13 EHV1V</td>
<td>124</td>
<td>64</td>
</tr>
<tr>
<td>P93746</td>
<td>ECI ARATH</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>P96602</td>
<td>DCTR BACSU</td>
<td>100</td>
<td>37</td>
</tr>
<tr>
<td>Q00408</td>
<td>VHUD METVO</td>
<td>51</td>
<td>21</td>
</tr>
<tr>
<td>Q00758</td>
<td>SP5B BACSU</td>
<td>457</td>
<td>0</td>
</tr>
<tr>
<td>Q01JR9</td>
<td>MRS2D URYSI</td>
<td>211</td>
<td>17</td>
</tr>
<tr>
<td>Q02944</td>
<td>URED KLEPN</td>
<td>29</td>
<td>118</td>
</tr>
<tr>
<td>Q02DE9</td>
<td>ATPL PSEAB</td>
<td>73</td>
<td>0</td>
</tr>
<tr>
<td>Q02VS5</td>
<td>MUTL LACLS</td>
<td>185</td>
<td>107</td>
</tr>
<tr>
<td>Q03667</td>
<td>MIC17 YEAST</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>Q03DS8</td>
<td>GLPK PEDPA</td>
<td>172</td>
<td>54</td>
</tr>
<tr>
<td>Q03Q17</td>
<td>TRUB LACBA</td>
<td>67</td>
<td>74</td>
</tr>
<tr>
<td>Q06199</td>
<td>YL456 YEAST</td>
<td>34</td>
<td>62</td>
</tr>
<tr>
<td>Q07ZS6</td>
<td>UVRC SHEFN</td>
<td>241</td>
<td>81</td>
</tr>
<tr>
<td>Q086C3</td>
<td>NUSB SHEFN</td>
<td>90</td>
<td>0</td>
</tr>
<tr>
<td>Q0ARD7</td>
<td>PDXH MARMm</td>
<td>56</td>
<td>58</td>
</tr>
<tr>
<td>Q0C5F3</td>
<td>SUCC HYPNA</td>
<td>140</td>
<td>78</td>
</tr>
<tr>
<td>Q0G9M8</td>
<td>RPOC1 LIRTU</td>
<td>211</td>
<td>69</td>
</tr>
<tr>
<td>Q01Q59</td>
<td>BIOB CLOP1</td>
<td>133</td>
<td>23</td>
</tr>
<tr>
<td>Q0VSV6</td>
<td>HEMH ALCBS</td>
<td>148</td>
<td>42</td>
</tr>
<tr>
<td>Q13478</td>
<td>IL18R HUMAN</td>
<td>50</td>
<td>228</td>
</tr>
<tr>
<td>Q14032</td>
<td>BAAT HUMAN</td>
<td>81</td>
<td>105</td>
</tr>
<tr>
<td>Q1CFH7</td>
<td>METN2 YERPn</td>
<td>114</td>
<td>86</td>
</tr>
<tr>
<td>Q1CNS4</td>
<td>G6P1 YERPn</td>
<td>255</td>
<td>40</td>
</tr>
<tr>
<td>Q1GBJ2</td>
<td>RPOA LACDA</td>
<td>76</td>
<td>81</td>
</tr>
<tr>
<td>Q1H0V1</td>
<td>VP1 MPRVN</td>
<td>171</td>
<td>692</td>
</tr>
<tr>
<td>Q1I0V1</td>
<td>VP1 MPRVN</td>
<td>171</td>
<td>692</td>
</tr>
<tr>
<td>Q1J570</td>
<td>GATB STRPF</td>
<td>194</td>
<td>43</td>
</tr>
<tr>
<td>Q1PEW8</td>
<td>FB127 ARATH</td>
<td>106</td>
<td>0</td>
</tr>
<tr>
<td>Q21276</td>
<td>YZVL CAEEL</td>
<td>295</td>
<td>20</td>
</tr>
<tr>
<td>Q21L55</td>
<td>Y1312 SACD2</td>
<td>89</td>
<td>41</td>
</tr>
<tr>
<td>Q256C3</td>
<td>CLPX CHLFF</td>
<td>149</td>
<td>49</td>
</tr>
<tr>
<td>Q27288</td>
<td>OBP2 HELVI</td>
<td>118</td>
<td>0</td>
</tr>
<tr>
<td>Q2FIC3</td>
<td>MNHA1 STAA3</td>
<td>537</td>
<td>19</td>
</tr>
<tr>
<td>Q2JPT2</td>
<td>TRPF SYNJB</td>
<td>74</td>
<td>38</td>
</tr>
<tr>
<td>Q2K4E6</td>
<td>Y3534 RHIEC</td>
<td>149</td>
<td>0</td>
</tr>
<tr>
<td>Q31XD9</td>
<td>RL19 SHIBS</td>
<td>22</td>
<td>51</td>
</tr>
<tr>
<td>Q31ZS3</td>
<td>YCHJ SHIBS</td>
<td>45</td>
<td>22</td>
</tr>
<tr>
<td>Q39837</td>
<td>ALB1 SOYBN</td>
<td>34</td>
<td>17</td>
</tr>
<tr>
<td>Q3B8E9</td>
<td>IFT43 XENLA</td>
<td>72</td>
<td>0</td>
</tr>
<tr>
<td>Q3K0D3</td>
<td>AROD STRA1</td>
<td>89</td>
<td>41</td>
</tr>
<tr>
<td>Q3Z0K4</td>
<td>COBT SHISS</td>
<td>165</td>
<td>21</td>
</tr>
<tr>
<td>Accession</td>
<td>Description</td>
<td>Start</td>
<td>End</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>sp Q47LL2</td>
<td>RL15 THEFY</td>
<td>22</td>
<td>28</td>
</tr>
<tr>
<td>sp Q493M5</td>
<td>CSRA BLOPB</td>
<td>14</td>
<td>27</td>
</tr>
<tr>
<td>sp Q4FVP3</td>
<td>SSTT PSYA2</td>
<td>290</td>
<td>7</td>
</tr>
<tr>
<td>sp Q4KK0T</td>
<td>DNAA PSEF5</td>
<td>265</td>
<td>32</td>
</tr>
<tr>
<td>sp Q58350</td>
<td>Y940 METJA</td>
<td>248</td>
<td>0</td>
</tr>
<tr>
<td>sp Q5DU25</td>
<td>IQEC2 MOUSE</td>
<td>337</td>
<td>23</td>
</tr>
<tr>
<td>sp Q5E218</td>
<td>MURI VIBF1</td>
<td>114</td>
<td>39</td>
</tr>
<tr>
<td>sp Q5HG77</td>
<td>TKT STAAC</td>
<td>271</td>
<td>54</td>
</tr>
<tr>
<td>sp Q5KQS4</td>
<td>FETC GLOBR</td>
<td>49</td>
<td>88</td>
</tr>
<tr>
<td>sp Q5LLT4</td>
<td>TRUB RUEPO</td>
<td>64</td>
<td>75</td>
</tr>
<tr>
<td>sp Q5M5L0</td>
<td>ARGJ STRT2</td>
<td>131</td>
<td>84</td>
</tr>
<tr>
<td>sp Q5PBP7</td>
<td>CTAA ANAAM</td>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>sp Q5RA17</td>
<td>RRP7A PONAB</td>
<td>102</td>
<td>38</td>
</tr>
<tr>
<td>sp Q5VYY2</td>
<td>LIPM HUMAN</td>
<td>172</td>
<td>43</td>
</tr>
<tr>
<td>sp Q5XY7</td>
<td>RL17 FELCA</td>
<td>66</td>
<td>26</td>
</tr>
<tr>
<td>sp Q63811</td>
<td>CANB2 MOUSE</td>
<td>89</td>
<td>9</td>
</tr>
<tr>
<td>sp Q63H76</td>
<td>RS8 BACCZ</td>
<td>30</td>
<td>36</td>
</tr>
<tr>
<td>sp Q6ANN6</td>
<td>NUON DESPS</td>
<td>355</td>
<td>0</td>
</tr>
<tr>
<td>sp Q6FPK6</td>
<td>RG11 CANGA</td>
<td>31</td>
<td>35</td>
</tr>
<tr>
<td>sp Q6FSP5</td>
<td>CSM2 CANGA</td>
<td>91</td>
<td>32</td>
</tr>
<tr>
<td>sp Q6GJ03</td>
<td>SARX STAAR</td>
<td>93</td>
<td>5</td>
</tr>
<tr>
<td>sp Q6GZM8</td>
<td>097R FRG3G</td>
<td>107</td>
<td>0</td>
</tr>
<tr>
<td>sp Q6LME3</td>
<td>Y3228 PHOPR</td>
<td>36</td>
<td>34</td>
</tr>
<tr>
<td>sp Q70RT7</td>
<td>CYB PLAMN</td>
<td>226</td>
<td>1</td>
</tr>
<tr>
<td>sp Q71W03</td>
<td>Y2748 LISMF</td>
<td>71</td>
<td>29</td>
</tr>
<tr>
<td>sp Q73PM9</td>
<td>RL2 TREDE</td>
<td>3</td>
<td>75</td>
</tr>
<tr>
<td>sp Q746Q5</td>
<td>RSMG GEOSL</td>
<td>82</td>
<td>42</td>
</tr>
<tr>
<td>sp Q7N364</td>
<td>END4 PHOLL</td>
<td>107</td>
<td>29</td>
</tr>
<tr>
<td>sp Q7TX80</td>
<td>THTR2 MYCBO</td>
<td>79</td>
<td>29</td>
</tr>
<tr>
<td>sp Q7V336</td>
<td>LEUC PROMP</td>
<td>150</td>
<td>64</td>
</tr>
<tr>
<td>sp Q7VIY0</td>
<td>YIDC HELHP</td>
<td>201</td>
<td>87</td>
</tr>
<tr>
<td>sp Q7VNF7</td>
<td>RS4 HAEDU</td>
<td>75</td>
<td>28</td>
</tr>
<tr>
<td>sp Q7VZG1</td>
<td>KCY BORPE</td>
<td>102</td>
<td>24</td>
</tr>
<tr>
<td>sp Q7YQM2</td>
<td>AFF2 PANTR</td>
<td>189</td>
<td>2</td>
</tr>
<tr>
<td>sp Q7Z2W4</td>
<td>ZCCHV HUMAN</td>
<td>145</td>
<td>113</td>
</tr>
<tr>
<td>sp Q7Z408</td>
<td>CSMD2 HUMAN</td>
<td>13</td>
<td>1151</td>
</tr>
<tr>
<td>sp Q82S93</td>
<td>COAX NITEU</td>
<td>93</td>
<td>61</td>
</tr>
<tr>
<td>sp Q83CV9</td>
<td>DEF1 COXBU</td>
<td>39</td>
<td>51</td>
</tr>
<tr>
<td>sp Q87E80</td>
<td>RL23 XYLFT</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>sp Q88A15</td>
<td>MSRA PSESM</td>
<td>53</td>
<td>28</td>
</tr>
<tr>
<td>sp Q88QV5</td>
<td>PQQB PSEP3</td>
<td>53</td>
<td>68</td>
</tr>
<tr>
<td>sp Q8CTB2</td>
<td>METN1 STAES</td>
<td>116</td>
<td>90</td>
</tr>
<tr>
<td>sp Q8E5P8</td>
<td>GLMS STRA3</td>
<td>228</td>
<td>109</td>
</tr>
<tr>
<td>sp Q8FQ24</td>
<td>ATPF COREF</td>
<td>162</td>
<td>0</td>
</tr>
<tr>
<td>sp Q8JTH2</td>
<td>PHOSP ABLVH</td>
<td>105</td>
<td>13</td>
</tr>
<tr>
<td>sp Q8KE85</td>
<td>Y805 CHLITE</td>
<td>54</td>
<td>20</td>
</tr>
<tr>
<td>Species ID</td>
<td>Gene ID</td>
<td>Protein Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>sp Q8LPJ4</td>
<td>AB2E</td>
<td>ARATH</td>
<td></td>
</tr>
<tr>
<td>sp Q8NVL8</td>
<td>LUKL2</td>
<td>STAAR</td>
<td></td>
</tr>
<tr>
<td>sp Q8WJ37</td>
<td>MATK</td>
<td>ANEAN</td>
<td></td>
</tr>
<tr>
<td>sp Q8WVZ7</td>
<td>RN133</td>
<td>HUMAN</td>
<td></td>
</tr>
<tr>
<td>sp Q8WYJ6</td>
<td>SEPT1</td>
<td>HUMAN</td>
<td></td>
</tr>
<tr>
<td>sp Q8Y2E1</td>
<td>KPRS</td>
<td>RALSO</td>
<td></td>
</tr>
<tr>
<td>sp Q8Y4B6</td>
<td>ATP6</td>
<td>LISMO</td>
<td></td>
</tr>
<tr>
<td>sp Q8Y9D6</td>
<td>METX</td>
<td>LISMO</td>
<td></td>
</tr>
<tr>
<td>sp Q8YFN7</td>
<td>RL9</td>
<td>BRUME</td>
<td></td>
</tr>
<tr>
<td>sp Q8YIS0</td>
<td>Y373</td>
<td>BRUME</td>
<td></td>
</tr>
<tr>
<td>sp Q8ZCC1</td>
<td>HDA</td>
<td>YERPE</td>
<td></td>
</tr>
<tr>
<td>sp Q8ZGV8</td>
<td>BETI</td>
<td>YERPE</td>
<td></td>
</tr>
<tr>
<td>sp Q925N2</td>
<td>SFXN2</td>
<td>MOUSE</td>
<td></td>
</tr>
<tr>
<td>sp Q96RD1</td>
<td>OR6C1</td>
<td>HUMAN</td>
<td></td>
</tr>
<tr>
<td>sp Q99WT9</td>
<td>ESSC</td>
<td>STAAN</td>
<td></td>
</tr>
<tr>
<td>sp Q9BPB1</td>
<td>O226A</td>
<td>CONTE</td>
<td></td>
</tr>
<tr>
<td>sp Q9BW60</td>
<td>ELOV1</td>
<td>HUMAN</td>
<td></td>
</tr>
<tr>
<td>sp Q9DHP6</td>
<td>VHR2</td>
<td>YLDV</td>
<td></td>
</tr>
<tr>
<td>sp Q9FI78</td>
<td>HST</td>
<td>ARATH</td>
<td></td>
</tr>
<tr>
<td>sp Q9GQ38</td>
<td>MAB21</td>
<td>CAEBR</td>
<td></td>
</tr>
<tr>
<td>sp Q9HM28</td>
<td>Y042</td>
<td>THEAC</td>
<td></td>
</tr>
<tr>
<td>sp Q9KD76</td>
<td>LEPA</td>
<td>BACHD</td>
<td></td>
</tr>
<tr>
<td>sp Q9M4C0</td>
<td>RR4</td>
<td>HAPHO</td>
<td></td>
</tr>
<tr>
<td>sp Q9MUL9</td>
<td>CYST</td>
<td>MESVI</td>
<td></td>
</tr>
<tr>
<td>sp Q9N0Z0</td>
<td>CXCR6</td>
<td>CERAT</td>
<td></td>
</tr>
<tr>
<td>sp Q9N9R7</td>
<td>TLR8</td>
<td>HUMAN</td>
<td></td>
</tr>
<tr>
<td>sp Q9PA83</td>
<td>RL1</td>
<td>XYLFA</td>
<td></td>
</tr>
<tr>
<td>sp Q9PTU1</td>
<td>DBX1A</td>
<td>DANRE</td>
<td></td>
</tr>
<tr>
<td>sp Q9RQQ9</td>
<td>DIVL</td>
<td>CAUCR</td>
<td></td>
</tr>
<tr>
<td>sp Q9U0M8</td>
<td>YPF06</td>
<td>PLAF7</td>
<td></td>
</tr>
<tr>
<td>sp Q9UTI7</td>
<td>TYSY</td>
<td>SCHPO</td>
<td></td>
</tr>
<tr>
<td>sp Q9Y6L6</td>
<td>S01B1</td>
<td>HUMAN</td>
<td></td>
</tr>
<tr>
<td>sp Q9ZKW7</td>
<td>MURJ</td>
<td>HELPJ</td>
<td></td>
</tr>
<tr>
<td>sp Q9ZPY1</td>
<td>PPOX2</td>
<td>ARATH</td>
<td></td>
</tr>
</tbody>
</table>
Appendix C

Project documentation

C.0.1 Milestones

- MS0 19.02.2013 Project start, Kick-Off meeting
- MS1 13.04.2013 Status meeting with Prof. Joller and Romeo Kienzler
- MS2 07.05.2013 Setup hardware cluster
- MS3 14.05.2013 Start experiments
- MS4 28.05.2013 Hand over for abstract and A0 poster
- MS5 31.05.2013 Project end

C.0.2 Week by week breakdown

week 1 (18.02)
setting up a wiki website and redmine
getting an overview about all the required parts in the documentation

week 2 (25.02)
getting started with the subject of MapReduce
research about the classifier RaptorX
overview about Hadoop framework, HDFS and Streaming
phone-call mit Romeo und Rémy Bruggmann, introducing InterProScan analysis tool

week 3 (04.03)
research about InterProScan tool
gaining information about uniprot/swissprot database and how we can get the files
research for FASTA-file format
getting started with documentation
tool RaptorX is runnable on CLI with one sequence
setup a virtual cluster with IBM BigInsights

week 4 (11.03)
two single node clusters were installed and configured for first step and try out Hadoop Streaming
the input file in FASTA-format has two sequences,
but files in Hadoop are read block by block and are splitted line after line,
so input file has to be transformed before loading it into Hadoop

week 5 (18.03)
starting with writing bash scripts
meeting with Rémy Bruggmann about possible interpretation of InterProScan output files

**week 6 (25.03)**
bash script which writes every FASTA-sequence in one-line. But it runs extremely slow.
we write down the required steps in our bash script which will be used as mapper function on the 
*Hadoop Streaming* job

**week 7 (01.04)**
replace the bash script for one-line FASTA-record by a C++ program
run a *Streaming* job with 15 sequences and getting an output file with the filtered information of 
*RaptorX* output file
run *InterProScan* on cluster
some of the sequences are difficult to calculate for *RaptorX* and take extremely long
experiments should show overhead of using *Hadoop* vs. native runs and scale out has to be linear

**week 8 (08.04)**
setting up eclipse with *BigInsights* plugin
writing a FASTA-input file reader is not so simple, because different *Hadoop* versions supports 
different classes of lineReader

**week 9 (15.04)**
update documentation according the template we get from Prof. Joller
change properties of *InterproScan* for single threaded runs
*RaptorX* run on a cluster with more than one node

**week 10 (29.04)**
*InterProScan* runs correctly on 1 node cluster
updating documentation

**week 11 (06.05)**
setting up hardware cluster and distribution of all required files for *RaptorX* and *InterProScan* on 
all nodes
running tests with bash scripts on a cluster with more than one node

**week 12 (13.05)**
plan the measurements
review documentation and discussion on the conclusion section at the documentation
restart experiments on cluster, because of incorrect settings on *InterProScan* properties and too 
many map tasks per node.
try out *RaptorX* runs with additional parameters because of the failing tasks

**week 13 (20.05)**
analysing experiment-results
updating documentation
started with design of poster and writing the abstract

**week 14 (27.05)**
completing documentation
Appendix D
CD content

Table (D.1) CD content

<table>
<thead>
<tr>
<th>file</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>./semesterThesisBuechiMathys.pdf</td>
<td>Documentation</td>
</tr>
<tr>
<td>./sa-Poster.ppt</td>
<td>A0 Poster</td>
</tr>
<tr>
<td>./summary.doc</td>
<td>summary of our project</td>
</tr>
<tr>
<td>./Documentation/</td>
<td>LATEX-Documentation-files</td>
</tr>
<tr>
<td>./fasta-sequences/</td>
<td>Used input files for experiments</td>
</tr>
<tr>
<td>./gnuplot/</td>
<td>gnuplot source files</td>
</tr>
<tr>
<td>./scripts/</td>
<td>used bash scripts</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Protein secondary structure [3]</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Supported algorithm on InterProScan [12]</td>
<td>8</td>
</tr>
<tr>
<td>3.2</td>
<td>Workflow of Streaming job</td>
<td>10</td>
</tr>
<tr>
<td>4.1</td>
<td>Test environment: hardware cluster located at IBM Switzerland</td>
<td>12</td>
</tr>
<tr>
<td>5.1</td>
<td>Mean-overhead of a InterProScan run with 100 various sequences</td>
<td>18</td>
</tr>
<tr>
<td>5.2</td>
<td>Mean-overhead of a InterProScan run with 250 various sequences</td>
<td>18</td>
</tr>
<tr>
<td>5.3</td>
<td>Mean-overhead of a RaptorX run with 100 equal sequences</td>
<td>19</td>
</tr>
<tr>
<td>5.4</td>
<td>Mean-overhead of a RaptorX run with 250 equal sequences</td>
<td>19</td>
</tr>
<tr>
<td>5.5</td>
<td>Throughput of InterProScan with 250 various sequences</td>
<td>20</td>
</tr>
<tr>
<td>5.6</td>
<td>Throughput of RaptorX</td>
<td>20</td>
</tr>
<tr>
<td>5.7</td>
<td>Occurrence of GO-IDs</td>
<td>21</td>
</tr>
<tr>
<td>5.8</td>
<td>Occurrence of IPR-numbers</td>
<td>22</td>
</tr>
</tbody>
</table>
List of Tables

2.1 The IUB/IUPAC standard codes for amino acids ......................... 3
3.1 Three-class secondary structure prediction .............................. 8
4.1 The servers of our test environment ..................................... 13
4.2 Used parameters for Hadoop Streaming job ............................. 15
5.1 Measurements InterProScan ........................................... 17
5.2 Measurements RaptorX ............................................... 17
B.1 Table of GO-ID and their Term name .................................. 30
B.2 Table of IPR-number and their Domain ................................ 33
B.3 Table of IPR-number and their Family ................................ 34
B.4 Secondary structure prediction ........................................ 35
D.1 CD content ............................................................ 43
Glossary

**FQDN**

*FQDN* means fully qualified domain name which represents the system/node in the domain name system hierarchy.

**GO database**

*GO* database stores ontology and annotation files which are contributed by the *GO Consortium*. The following url points to the online websearch at the *GO Database*:

http://amigo.geneontology.org/cgi-bin/amigo/go.cgi

**GO-ID**

Every entry in the *GO* database has an unique identification number. It starts with the term “GO:” and is followed by a number.

**HDFS**

*HDFS* (Hadoop Distributed File System) is a part of the *Hadoop* framework. It is a distributed file system which is fault tolerant and designed to store big datasets in parts of the different nodes of a cluster.

**IPR-number**

Every entry in the *InterProScan* database has a unique identification number. It starts with the term “IPR:” and is followed by a number.

**MapReduce**

*MapReduce* is a pattern introduced by *Google* for calculating huge dataset on parallel systems.
Bibliography

Tips, tricks and pointers when setting up your first hadoop cluster to run map reduce jobs.
[Online; accessed 28-May-2013].

Hadoop performance tuning: a case study.
[Online; accessed 28-May-2013].

Click4biology - secondary structure.
[Online; accessed 30-May-2013].

[Online; accessed 30-April-2013].

[Online; accessed 28-May-2013].

Mapreduce: simplified data processing on large clusters.

Fasta-format.
[Online; accessed 28-May-2013].

Hadoop Streaming.
[Online; accessed 27-May-2013].

Hadoop user groups.
[Online; accessed 30-May-2013].

hdfs commands.
Performance tuning hadoop wiki.  
http://wiki.apache.org/hadoop/PerformanceTuning. 
[Online; accessed 30-May-2013].

Overview of the different databases used on interpro.  
http://www.ebi.ac.uk/training/online/sites/ebi.ac.uk.training.online/files/user/84/documents/figure_interpro.png. 
[Online; accessed 30-May-2013].

Apache hadoop performance-tuning methodologies and best practices.  

IUPAC/IUB Single-Letter Codes Within Nucleic Acid and Amino Acid Sequences.  

A novel structural position-specific scoring matrix for the prediction of protein secondary structures.  
Bioinformatics, 2011.

Apache hadoop: Best practices and anti-patterns.  
[Online; accessed 28-May-2013].

Introduction to hadoop, 2008.

Failure trends in a large disk drive population.  

[Online; accessed 30-April-2013].

http://www.ebi.ac.uk/uniprot/TrEMBLstats.  
[Online; accessed 30-April-2013].