Porting and running a setup for functional genomics on computational grids

Silvano Paoli (1,2)
Stefano Cozzini (2,3)
Cesare Furlanello (1)

(1) ITC-irst, Trento
(2) ICTP E-GRID Project, Trieste
(3) INFM/Democritos Trieste

BIT 2005 Tutorial, 17 March 2005
Predictive Profiling

QUESTION:

Can we predict, using a discriminating molecular signature, which patients will not respond to a therapy?
Perfect prediction (CV) may be obtained with only 8 genes (*) …

But when repeating the same experiment after a label randomization, a very similar result is reached: 14 genes are sufficient to get a zero error estimate.

Consider the public colon cancer dataset, and select genes using the data both for the feature ranking and for the model assessment (*): similar results of near perfect classification with few genes published in PNAS, Machine Learning, Genome Research, BMC Bioinformatics, etc.

The selection bias problem !!
Predictive classification and functional profiling

We develop algorithms and software systems for

1. Predictive classification, feature selection, discovery

Our BioDCV system: a set-up based on the E-RFE algorithm for Support Vector Machines (SVM)

- Control of selection bias,
  a serious experimental design issue in the use of prognostic molecular signatures

- Subtype identification
  for studies of disease evolution and response to treatment
To avoid selection bias ($p >> n$): a COMPLETE VALIDATION SCHEME*

- externally a stratified random partitioning,
- internally a model selection based on a K-fold cross-validation

$3 \times 10^5$ SVM models (+ random labels $\rightarrow 2 \times 10^6$)**

1. Extern loop:
   a. develop models on partitions of the training data.
   b. Validate on the remainder test set, disjoint from $\text{dev:VAL}$

2. Model tuning and Feature ranking: **OFS-M**

3. Optimal gene panel estimates: **ONF**

---

** Binary classification, on a 20 000 genes x 45 cDNA array, 400 loops
Examples of predictive classification

*Colon cancer*: 62 samples (40 tumoral + 22 control) described by 2000 genes (Alon et. al, 1999)

*Lymphoma*: 96 samples (74 tumoral + 24 control) described by 4096 genes (Alizadeh et. al, 2000)

*Tumor vs. Metastases*: 76 samples (64 primary tumoral + 12 metastatic) described by 16063 genes (Ramaswamy et al., 2001)

*High Sezary CTCL*: 30 samples (18 disease + 12 control) described by 6660 genes (Kari et al., 2003) – coll. Wistar Inst.

*Glioma*: 50 samples (28 glioblastoma + 22 oligodendroglioma) described by 12 627 genes (Nutt et al., 2003)

*Breast cancer*: 37 samples (18 high risk + 19 low risk) described by 12 625 genes (Huang et al., 2003)

*Mouse Model of Myocardial Infarction*: 36 samples (18 infarcted + 18 control) described by 12488 geni (Cardiogenomics PGA [http://cardiogenomics.med.harvard.edu](http://cardiogenomics.med.harvard.edu), 2003)
## With a Linux OpenMosix HPC facility

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dimension</th>
<th>Job no</th>
<th>Svm</th>
<th>Tot (h)</th>
<th>Cluster (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast Cancer</td>
<td>37 x 12625</td>
<td>200</td>
<td>158 961</td>
<td>1177</td>
<td>36</td>
</tr>
<tr>
<td>MM NI LV - LV</td>
<td>41 x 12488</td>
<td>400</td>
<td>333 163</td>
<td>2234</td>
<td>59</td>
</tr>
<tr>
<td>Glioma</td>
<td>50 x 12627</td>
<td>200</td>
<td>146 262</td>
<td>1864</td>
<td>57</td>
</tr>
<tr>
<td>High sezary Survival</td>
<td>49 x 2244</td>
<td>200</td>
<td>126 150</td>
<td>63</td>
<td>1,4</td>
</tr>
<tr>
<td>High sezary Without C0032</td>
<td>29 X 6600</td>
<td>400</td>
<td>309 681</td>
<td>655</td>
<td>23</td>
</tr>
</tbody>
</table>

Our HPC resource, **MpaCluster**, 6Xeon+ 40 Pentium CPU, OpenMosix, 3 TeraB central storage. Upgraded in 2005. → production in GRID.
Roadmap for a new grid application

Starting from a suite of C modules and Perl/shell scripts running on a local HPC resource ...

1. Optimize modules and scripts:
   - database management of data, of model structures, of system outputs, scripts for OpenMosix Linux Clusters

2. Wrap BioDCV into a grid application
   - Learn about grid computing
   - Port the serial version on a computational grid testbed
   - Analyze/verify results: identify needs/problems

3. Wrap with C MPI scripts
   - Build the MPI mechanism
   - Experiment on the testbed
   - Submit on production grid
   - Test scalability

4. Production

(*) : the BioDCV system

- Sept-Dec 04
- November-04 January 2005
- February 2005
- March 05: Up and Running!
1. Optimize modules and scripts

- Rewrite shell/Perl scripts in C language
  - control I/O costs,
  - a process granularity optimal for temporary data allocation without tmp files
  - convenient for migrations

- SQLite interface (Database engine library)
  - SQLite is small, self-contained, embeddable
  - It provides a relational access to model and data structures (inputs, outputs, diagnostics)
  - It supports transactions and multiple connections, databases up to 2 terabytes in size

---

- a local copy of setup db:
  + model definitions
  + a copy of of data
  + indexes defining the partition of the replicate sample(s)
2. Wrapping into a grid application

- Why porting into the grid?
  - Because we need “enough” computational resources...

- How to port the BioDCV in grid?
  - PRELIMINARY
    - Identify a collaborator with experience in grid computing (e.g. the Egrid Project hosted at ICTP [http://www.egrid.it](http://www.egrid.it))
    - Train human resources (SP → Trieste)
    - Join the Egrid testbed (installing a supernode in Trento)
  - HANDS-ON
    - Porting of the serial application on the testbed
    - patch code as needed: code portability is mandatory to make life easier
    - Identify requirements/problems
A few EDG definitions

**User Interface (UI):** machine to access the GRID

**Storage Element (SE):** stores the user data in the grid and makes it available for subsequent elaboration

**Computing Element (CE):** where the grid user programs are delivered for elaboration: this is usually a front-end to several elementary Worker Node machines

**Worker Node (WN):** machines where the user programs are actually executed, possibly with multiple CPUs
The ICTP Egrid project infrastructures

- **The local testbed in Trieste**
  - Small computational grid based on EDG middleware + Egrid add-ons
  - Designed for testing/training/porting of applications
  - Full compatibility with Grid.it middleware

- **The production infrastructure:**
  - A Virtual Organization within Grid.it, with its own services
  - Star topology with central node in Padova
Hands on

- Porting the serial application
  - Easy task due to portability (no actual work needed)
  - No software/library dependencies
- Testing/Evaluation
- Problems identified:
  - Job submission overhead due to EDG mechanisms
  - Managing multiple (~hundreds/thousands) jobs is difficult and cumbersome
- Answer: parallelize jobs on the GRID via MPI
  - Single submission
  - Multiple executions
3. Wrap with C MPI scripts

- **How can we use C MPI?**
  - **Prepare two wrappers, and an unifier**
    - one shell script to submit jobs (*BioDCV.sh*)
    - one C MPI program (*Mpba-mpi*)
    - one shell script to integrate results (*BioDCV-union.sh*)
  - **BioDCV.sh in action:**
    - copies file from and to Storage Element (SE) and distributes the microarray dataset to all WNs.
    - It then starts the C MPI wrapper which spawns several runs of the BioDCV program (optimize for resources)
    - When all BioDCV runs are completed, the wrapper copies all the results (SQLite files) from the WNs to the starting SE.
  - **MPBA-MPI executes the BioDCV runs in parallel**
  - **BioDCV-union.sh collates results in one SQLite file (→ R)**
Using BioDCV in Egrid

"Edg-job-submit bioDCV.jdl"

UI Egrid Live CD*

Trento
CE+SE+WN

Resource broker (PD-TN)

Padova
CE 2.8 TByte
SE 100 cpus
WNs

Palermo
CE+SE+WN

Trieste
CE+SE+WN

‘Edg-job-submit bioDCV.jdl’

a bootable Linux live-cd distribution with a complete suit of GRID tools by Egrid (ICTP Trieste)
A Job Description

```
[
    Type = "Job";
    JobType = "MPICH";
    NodeNumber = 64;
    Executable = "BioDCV.sh";
    Arguments = "Mpba-mpi 64 lfn:/utenti/spaoli/sarcoma.db 400";
    StdOutput = "test.out";
    StdError = "test.err";
    InputSandbox = {"BioDCV.sh","Mpba-mpi","run", "run.sh"};
    OutputSandbox = {"test.err","test.out","executable.out"};
    Requirements = other.GlueCEInfoLRMSType == "PBS" ||
                   other.GlueCEInfoLRMSType == "LSF";
]
```
Using BioDCV in Egrid (II)

**First step:**

BioDCV.sh runs on

Request file

BioDCV.sh copies data from SE to the WN

WN 1

sarcoma.db

SE

**Second step:**

BioDCV Sarcoma.db

WN 1

WN 2

WN 3

WN n

Mpba-mpi and Sarcoma.db are distributed to all the involved WNs
Using BioDCV in Egrid (III)

Third step:
BioDCV is executed on all involved WNs by MPI.

Fourth step:
BioDCV.sh copies all results (SQLite files) from the WNs to the starting SE.
Scalability Test 1

1. Computing time (step 3)
2. File copying (step 2)
3. Total time: steps from 1 to 4
Scalability Test 2

1. Computing time (step 3)
2. File copying (step 2)
3. Total time: steps from 1 to 4

Average on 3 experiments, CPU: Xeon 2.8 GHz
With one CPU, the full dataset is loaded by a single process
Scalability of BioDCV in Grid.it

INT Sarcoma data:

<table>
<thead>
<tr>
<th>CPUs no.</th>
<th>Computing (seconds)</th>
<th>File copying (seconds)</th>
<th>Total time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>65883</td>
<td>25</td>
<td>66018</td>
</tr>
<tr>
<td>2</td>
<td>22566</td>
<td>46</td>
<td>22749</td>
</tr>
<tr>
<td>4</td>
<td>6316</td>
<td>80</td>
<td>6545</td>
</tr>
<tr>
<td>8</td>
<td>2204</td>
<td>156</td>
<td>2533</td>
</tr>
<tr>
<td>16</td>
<td>1352</td>
<td>310</td>
<td>1762</td>
</tr>
<tr>
<td>32</td>
<td>507</td>
<td>617</td>
<td>1454</td>
</tr>
<tr>
<td>64</td>
<td>554</td>
<td>1263</td>
<td>2389</td>
</tr>
</tbody>
</table>
Semi-supervised analysis on Wistar CTCL (Kari et al, 2003): 400 runs, 300 K models

Number of features

Error rate (High Sesary)  Error rate (controls)  No-information error rate

1: C022 1 (87)  2: C021 1 (102)  3: C004 1 (103)  4: C002 2 (98)  5: C003 2 (107)  6: C010 2 (95)
7: C011 2 (95)  8: C018 6 (103)  9: C019 6 (103)  10: C020 6 (99)  11: C021 3 (117)  12: C022 3 (91)
13: S123 1 (91)  14: S127 1 (84)  15: S112 5 (97)  16: S115 1 (93)  17: S114 3 (93)  18: S107 2 (89)
Results

The pros:

- MPI execution on the GRID in a few days..
- The tests showed scalable behavior of our grid application for increasing numbers of CPUs
- Grid computing reduces significantly production times and allows to tackle larger problems (see next slide)

The cons:

- Data movements limit the scalability for a large number of CPU’s
  - Note: this is a GRID.it limitation: there is no shared Filesystem between the WNs, so each file needs to be copied everywhere!

To hide the latency (ideas):

- Smart data distribution from MWN to WN’s:
  - Reduce the amount of data to be moved
  - Proportionate BioDCV subtasks to local cache
- Data transferred via MPI communication
  - Requires MPI coding and some MPI adaptation of the code)
Challenges (AIRC-BICG)

HPC-Interaction: access through web front-ends to GRID HPC

- **BASIC CLASSIFICATION:** MODELS, lists, additional tools
- **Tools for researchers:** subtype discovery, outlier detection
- **Connection to data** (DB–MIAME)

---

**Sample Analysis**

- Save listed
- Save selected
- Select outliers
- View clustering
- Select
- Remove

**Selection list:**

- lv2_3w_574 [1] (88)
- lv2_24h_517 [1] (70)
- lv_24h_510 [1] (80)
- lv_24h_517 [1] (87)
- lv_8w_340 [1] (89)
- mlm_339 [1] (64)
- mlb_48h_351 [-1] (63)

**Outliers**

- Delete
- Bigger image
- Info
## Acknowledgments

<table>
<thead>
<tr>
<th>ITC-irst, Trento</th>
<th>IFOM-FIRC and INT, Milano</th>
</tr>
</thead>
<tbody>
<tr>
<td>Davide Albanese</td>
<td>James Reid</td>
</tr>
<tr>
<td>Giuseppe Jurman</td>
<td>Manuela Gariboldi</td>
</tr>
<tr>
<td>Stefano Merler</td>
<td>Marco A. Pierotti</td>
</tr>
<tr>
<td>Roberto Flor</td>
<td></td>
</tr>
<tr>
<td>Alessandro Soraruf</td>
<td></td>
</tr>
<tr>
<td><strong>ICTP E-GRID Project, Trieste</strong></td>
<td><strong>Grants:</strong></td>
</tr>
<tr>
<td>Angelo Leto</td>
<td>BICG (AIRC)</td>
</tr>
<tr>
<td>Cristian Zoicas</td>
<td>Democritos</td>
</tr>
<tr>
<td>Riccardo Murri</td>
<td></td>
</tr>
<tr>
<td>Ezio Corso</td>
<td>Data:</td>
</tr>
<tr>
<td>Alessio Terpin</td>
<td>ShoweLab (Wistar)</td>
</tr>
<tr>
<td></td>
<td>Cardiogenomics PGA</td>
</tr>
</tbody>
</table>

Grants: BICG (AIRC) Democritos

Data: ShoweLab (Wistar) Cardiogenomics PGA