intcomp:
Benchmarking tool for gene expression - copy number data integration models

Leo Lahti*and Martin Schäfer

September 9, 2011

1 Introduction

Several algorithms have been suggested to integrate gene expression and DNA copy number measurement to discover novel cancer-associated genes and chromosomal regions. However, quantitative comparison of these models has been missing. This R package provides a benchmarking pipeline for integrative cancer gene detection methods. Cancer gene detection performance of each algorithm is evaluated by comparing the prioritized candidate gene list from each method to a golden standard list of known cancer genes in simulated and real data sets.

The package is experimental beta-release and provided as is. The main purpose of this vignette is to provide reference to algorithmic details of the benchmarking framework used in [?] and sufficient documentation of the functionality such that similar experiments can be replicated by experienced R users.

1.1 Comparison methods

The current version implements the comparison of the following ge/cn integration algorithms: CNAmet [?, ?], DRI [?], edira [?], intCNGEan [?], Ortiz-Estevez [?], pint [?], PMA [?], PREDA/SODEGIR [?], and SIM [?]. Multiple variants are evaluated for some methods [?].

2 Examples

2.1 Installing dependencies

The benchmarking pipeline depends on various external R packages. Install the dependencies from within R using the following commands:

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite(c("biomaRt", "DNAcopy", "DRI", "edira", "hgu133ahsentrezg.db", "intCNGEan",
+     "org.Hs.eg.db", "PMA", "SIM", "PREDA", "CGHcall", "CNAmet"))
```

*leo.lahti@iki.fi

You may need to install the following packages manually: curl library[1] and the R packages XML[2], RCurl[3], edira[4], intCNGEan[5], org.Hs.eg.db[6], PREDA/SODEGIR[7], and CNAmet[8].

3 Running the comparison tests

3.1 Loading example data

The package contains the data sets from [?] and [?]. These two data sets are available from public sources and have been used to validate multiple alternative integration algorithms.

3.1.1 Pollack et al. (2002)

Loading Pollack et al. (2002) [?] data set[9].

```
> library(intcomp)
```

dmt Copyright (C) 2008-2011 Leo Lahti and Olli-Pekka Huovilainen. This program comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under FreeBSD, see the licensing terms for details.

pint Copyright (C) 2008-2011 Olli-Pekka Huovilainen and Leo Lahti.

This program comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under the FreeBSD license. See the licensing terms for details.

```
> data(pollack)
> dat <- read.pollack(chrs = 1:22, CopyNoGeneDataset4719, clone2geneid)
> ge <- dat$ge$data
> cn.raw <- dat$cn$data
> gene.info <- dat$ge$info
```

3.1.2 Hyman et al. (2002)

Loading Hyman et al. (2002) [?] data set[10].

```
> data(hyman)
> library("org.Hs.eg.db")
> dat <- read.hyman(cdna, cgh, genenames, chrs = 1:22, as.list(org.Hs.egALIAS2EG))
```

[1] http://curl.haxx.se/download.html

[2] http://cran.r-project.org/web/packages/XML/index.html

[3] http://www.omegahat.org/RCurl/

[4] http://www.statistik.tu-dortmund.de/ schaefer/

[5] http://www.few.vu.nl/ wvanwie/software/intCNGEan/intCNGEan.html

[6] http://www.bioconductor.org/packages/release/data/annotation/html/org.Hs.eg.db.html

[7] http://www.xlab.unimo.it/PREDA/PREDAinstall.R; http://www.xlab.unimo.it/PREDA/PREDAsampledata_0.1.7.tar.gz; http://www.xlab.unimo.it/PREDA/PREDA_0.2.12.tar.gz

[8] http://csbi.ltdk.helsinki.fi/CNAmet/

[9] http://www.pnas.org/content/suppl/2002/09/23/162471999.DC1/4719CopyNoGeneDatsetLegend.html accessed June 2, 2010.

[10] HymancdnaDataA.tab, HymancghDataA.tab and HymanAcc.mat obtained from http://www.ece.ucsb.edu/pubs/ieee/index.shtml accessed June 2, 2010.

```
> ge <- dat$ge
> cn.raw <- dat$cn
> gene.info <- dat$cn.raw$info
```

3.2 Preprocessing copy number data

Some methods require segmented and/or called copy number data, obtained with

```
> cgh <- process.copynumber(cn.raw)
> cn.seg <- list(data = assayDataElement(cgh, "segmented"), info = gene.info)
> cn.call <- list(data = assayDataElement(cgh, "calls"), info = gene.info)
> rownames(cn.call$data) <- rownames(cn.seg$data) <- rownames(ge$data)
> cn.call$info <- cn.seg$info <- cn.raw$info
```

3.3 Golden standard list of known cancer genes

The golden standard list of known breast cancer genes from The Breast Cancer Gene Database[11] [?] was downloaded and stored to the tgdb object. This script details how to load the list, convert to Entrez GeneIDs and select the genes that are included in the experimental data:

```
> library("org.Hs.eg.db")
> data(tgdb)
> cancerGenes <- get.brca.genes(rownames(dat$ge$data), as.list(org.Hs.egALIAS2EG),
+     tgdb)
```

3.4 Simulating data

To evaluate the methods based on simulations, data can be simulated following roughly the approach given in [?], but with greater flexibility. The quantile grid to be simulated can be defined entirely by the user, as well as the mixing weight, the number of different variances to be considered and the call probabilities (for testing *intcngean*).

```
> library(ediraAMLdata)
> data(AMLdata, package = "ediraAMLdata")
> test_schaefer <- test.simulation(GE, CN, probespanGE = 16, probespanCN = 100,
+     method = "schaefer", Inner = 3:5, Outer = c(1:2, 6:7), probs_GE = c(0.025,
+         0.075, 0.3, 0.5, 0.7, 0.925, 0.975), probs_CN = c(0.025, 0.075, 0.3,
+         0.5, 0.7, 0.925, 0.975), cancer_GE = c(1, 1, 2, 2, 6, 6, 7, 7), cancer_CN = c(1,
+         2, 1, 2, 3, 4, 3, 4), n = 100, weight = 1/10, variances = c(1/4, 1/2,
+         1, 2, 4), GE_norm = 4, CN_norm = 2, seed = 42, call_probs = c(0.001,
+         0.005, 0.01, 0.0125, 0.04, 0.925, 0.99))
```

Also, testing on the data simulated by Francesco Ferrari is provided. Here, the data are given, i.e., the user has no influence on their characteristics.

```
> test_ferrari <- test.simulation(GE, CN, probespanGE = 16, probespanCN = 100,
+     method = "ferrari")
```

[11] http://www.tumor-gene.org/cgi-bin/TGDB/tgdb_by_name.cgi accessed 5.6.2010; 'tgdb_by_name.cgi.html' and 'tgdb.txt'

4 Running the benchmarking pipeline

The example data set now contains (i) gene expression data (ge), (ii) gene copy number data (cn), (iii) optional sample class labels (tumor/normal), and (iv) golden standard list of known cancer genes. The ge and cn data sets are lists containing *data* and *info* fields. The probes in gene expression and gene copy number are paired; *data* is a data matrix with gene expression (ge$data) or gene copy number (cn$data) data; *info* field is a data frame containing additional information about genes: *loc* indicates the genomic location of the probes in base pairs (numeric); *chr* and *arm* are factors indicating the chromosome and chromosomal arm of the probe, respectively. To run the benchmarking tests for all methods, use:

```
> methods <- c("CNAmet", "edira")
> res_real <- test.geneorder.pipeline(ge = ge, cn.raw = cn.raw, cn.seg = cn.seg,
+     cn.call = cn.call, cghCall = cgh, Labels = NULL, cancerGenes = cancerGenes,
+     nperm = 20, input = "real", version = "normal", methods = methods, references = "non
> auc.ordered <- sort(unlist(res_real$auc))
```

```
> methods <- c("CNAmet", "edira")
> res_simulated <- test.geneorder.pipeline(ge = test_schaefer$ge, cn.raw = test_schaefer$c
+     cn.seg = NULL, cn.call = test_schaefer$cn.call, cghCall = test_schaefer$cn.cghCall,
+     ge.norm = test_schaefer$ge.norm, cn.norm = test_schaefer$cn.norm, Labels = test_scha
+     cancerGenes = test_schaefer$cancerGenes, nperm = 20, input = "simulations.equal.dime
+     version = "normal", methods = methods, callprobs = sim$callprobs, references = "both
> auc.ordered <- sort(unlist(res_simulated$auc))
```

If segmented data is supplied, then it will be used for analysis. If segmented data is missing, then raw data will be used if supplied. If both segmented and raw data are missing, then called data will be used if supplied.

The following *methods* are available: *OrtizEstevez*, *intcngean*, *edira*, *pint*, *SIM.window}, SIM.full}, DRI.cp, DRI.cs, PMA.raw, CNAmet, PREDA and for two-group comparisons (requiring Labels) DRI.ct, DRI.ss, DRI.srank, DRI.sraw. The cancer gene prioritization of each method is compared to the golden standard list of known cancer genes; the result contains running times of the algorithms and the AUC values from ROC analysis [?]. The AUC values provide quantitative estimates of model performance in cancer gene detection and provide the basis for comparisons.*

Acknowledgements

The package has been supported by EuGESMA COST Action BM0801: European Genetic and Epigenetic Study on AML and MDS.