Integrating literature-constrained and data-driven inference of signalling networks with *CNORfeeder*
(version 1.0.0 and above)

March 4, 2013

Contents

1 Installation

This software is written in the R language, so in order to use it you will need to have R installed on your computer. For more information and download of R, please refer to http://www.r-project.org/. For more information about how to install R packages, please refer to http://cran.r-project.org/doc/manuals/R-admin.html#Installing-packages. This package relies on several Bioconductor packages (CellNOptR, RBGL, graph, minet) and CRAN packages (igraph catnet). The following code installs the *CNORfeeder* dependencies and suggested packages that are on Bioconductor website, open an R session and type:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("RBGL")
> biocLite("graph")
> biocLite("minet")
> biocLite("CellNOptR")
```

The following code installs the *CNORfeeder* dependencies that are available on CRAN website. Open a R session and type:

```
> install.packages("igraph")
> install.packages("catnet")
```

Before starting this tutorial you also need to install the package *CNORfeeder*. You can either install *CNORfeeder* from Bioconductor by typing:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("CNORfeeder")
```
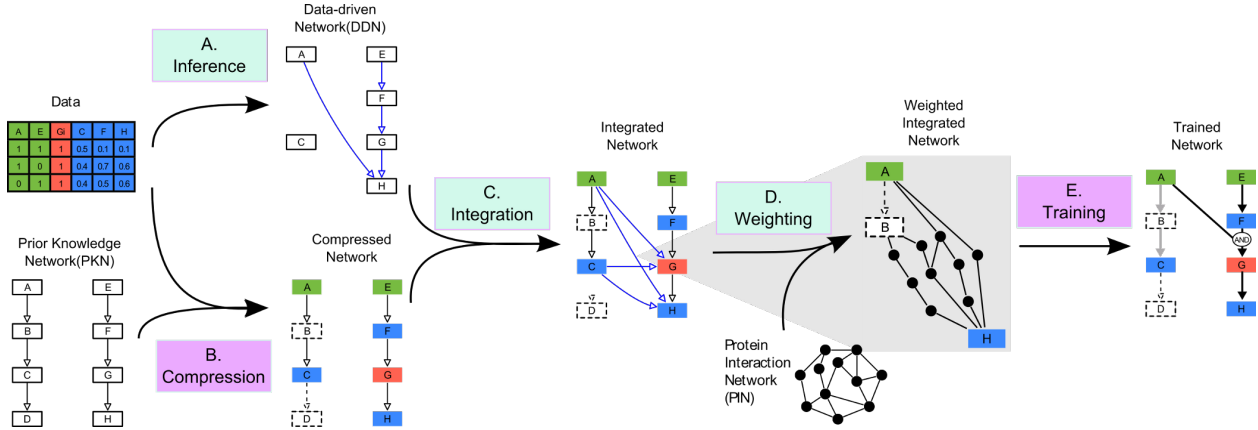
Figure 1: Integrated *CNORfeeder* - *CellNOptR* pipeline. Figure taken from [1].

or from a tar ball as follows:

```
> install.packages("path_to_CNORfeeder/CNORfeeder_1.0.0.tar.gz",
+     repos=NULL, type="source")
```

or, using the R GUI by clicking on "Packages & Data" then "Package installer", then choosing "local source" from the dropdown menu, clicking "install", choosing *CNORfeeder_1.0.0.tar.gz* and finally clicking "open".

A series of books about R can be found on the R project website (http://www.r-project.org/), and many tutorials are available on the internet. If you are a complete beginner, all you need to know is that by typing "?nameOfFunction" you get the help page about the function that you are interested in.

2 Introduction

The package *CNORfeeder* permits to extend a network derived from literature with links derived strictly from the data via various inference methods using information on physical interactions of proteins to guide and validate the integration of links, as described in [1]. The package is designed to be integrated with *CellNOptR*, a package described in [4] and based on methods described in [3] (see http://www.cellnopt.org for further details on the project). The integrated pipeline is illustrated in Figure 1 where steps performed by *CNORfeeder* are represented in cyan boxes and the ones performed by *CellNOptR* are in magenta boxes.

A Data are used to infer a strictly data-driven network (DDN) using reverse-engineering methods (as for now FEED, ARACNe, CLR and Bayesian networks are implemented);

B the prior knowledge network (PKN) is compressed according to the data (green, red and blue nodes are respectively stimulated, inhibited and measured), removing non-identifiable nodes (dashed ones);

C the compressed network is integrated with the DDN (blue links are obtained from the DDN and black ones from the PKN);

D information derived from protein-protein interaction network (PIN) is used to support and prioritize integrated links;

E the integrated network is used as input for the training: in trained model, thick black lines denote interactions (and gates) in final model, and light-grey links present in integrated network but not in trained model.

3 Example

We will illustrate the use of package *CNORfeeder* showing, step by step, the analysis described in [1]. It is performed on a real data set, which is a part of the network analysed in [3] and comprises 40 species and 58 interactions in the PKN. This network was also used for the signaling challenge in DREAM 4 (see http://www.the-dream-project.org/). The associated data was collected in hepatocellular carcinoma cell line HepG2 (see [2]).

First of all we load the library

```
> library(CNORfeeder)
```

and the data

```
> # load the data already formatted as CNOlist
> data(CNOlistDREAM,package="CellNOptR")
> # load the model (PKN) already in the CNO format
> data(DreamModel,package="CellNOptR")
```

For additional information about how to read the data see *CellNOptR* documentation.

3.1 A. Inference - CNORfeeder

Different methods can be used to reverse-engineer a network purely from data:

1. *makeBTables* can be used to infer a cause-effect network interpreted as Boolean tables using the method FEED [5];

2. *MIinference* allows to use the R package *minet* ([6]) to infer networks based on mutual information, i.e. ARACNE and CLR methods;

3. *Binference* allows to use the R package *catnet* (http://cran.r-project.org/web/packages/catnet/index.html) to derive Bayesian networks;

In this example, the FEED method (descirbed in [5]) is applied. The inferred DDN is encoded in Boolean tables. For each protein, a Boolean table is inferred having one columns for each stimulus and one row for each inhibitor. If a stimulus produces a significant effect on the activity level of the protein this is codified with a 1 in the corresponding column, if also the inhibitor affects the protein there is a 2 in the corresponding cell. The sign of the regulation is coded in separate tables. Significance is assessed taking into account the experimental error. Parameter k is the parameter which determines the threshold of significancy of the effect of stimuli and inhibitors, being multiplied by the measurement error in order to assess the relevance of a link. Tuning parameter k allows to adjust the sparsity of the network.

```
> BTable <- makeBTables(CNOlist=CNOlistDREAM, k=2, measErr=c(0.1, 0))

[1] "p38 : the condition with stimulus il1a and inhibitor p38  is missing or NA"
[1] "p38 : the condition with stimulus tgfa and inhibitor p38  is missing or NA"
[1] "mek12 : the condition with stimulus il1a and inhibitor mek12  is missing or NA"
[1] "mek12 : the condition with stimulus tgfa and inhibitor mek12  is missing or NA"
```

The resulting network is shown in Figure 2, upper panel. FEED method also allows to rank links according to the upper limit value of parameter k allowing the presence of the link.

```
> Lrank <- linksRanking(CNOlist=CNOlistDREAM, measErr=c(0.1, 0), savefile=FALSE)
```

3.2 B. Compression - CellNOptR

The preprocessing step includes the compression of the network to exclude non-identifiable nodes; the full details of preprocessing the model can be found in the *CellNOptR* package (see vignette for comprehensive explanation).

```
> model<-preprocessing(data=CNOlistDREAM, model=DreamModel)

[1] "The following species are measured: akt, erk12, ikb, jnk12, p38, hsp27, mek12"
[1] "The following species are stimulated: igf1, il1a, tgfa, tnfa"
[1] "The following species are inhibited: ikk, mek12, pi3k, p38"
[1] "The following species are not observable and/or not controllable: "
```

3.3 C. Integration - CNORfeeder

The inferred DDN is integrated with the prior knowledge, taking into account that networks inferred directly from data (DDNs) are limited only to measured, inhibited and stimulated nodes, thus a link in the DDN can correspond to multiple links in th PKN.

```
> modelIntegr <- mapBTables2model(BTable=BTable,model=model,allInter=TRUE)

> modelIntegr$reacID[modelIntegr$indexIntegr]

 [1] "igf1=akt"          "tgfa=akt"          "tgfa=mek12"
 [4] "!il1a+mek12=erk12" "il1a=ikk"          "tnfa=ikk"
 [7] "jnk12=ikk"         "il1a=jnk12"        "il1a=p38"
[10] "ikk=p38"           "ikb=p38"           "mek12=p38"
[13] "erk12=p38"         "hsp27=p38"         "pi3k=p38"
[16] "akt=p38"           "!tgfa+p38=hsp27"   "il1a=mek12"
```

The resulting network model can be plotted as follows (Figure 2, lower panel), using the *indexIntegr* argument to highlight in purple integrated links:

```
> plotModel(model=modelIntegr, CNOlist=CNOlistDREAM, indexIntegr=modelIntegr$indexIntegr)
```

3.4 D. Weighting - CNORfeeder

Links inferred based only on data are, in principle, less reliable than those derived from prior knowledge, thus we allow to weight added links differently. The attribute *integrFac* is the multiplicative factor to differently prioritize integrated links during the optimization (next step). A higher value of *integrFac* forces the genetic algorothm used for optimization, to strongly prioritize links from the PKN and to limit the choice of integrated links, even if their presence in the network model improves the fit of the model to the data. If *integrFac* is set equal to 1, all links have the same weight and the optimization algorithm will select those links which minimize the fit, disregarding if they are derived from data or from prior knowledge.

```
> modelIntegrWeight <- weighting(modelIntegr=modelIntegr, PKNmodel=DreamModel,
+                                CNOlist=CNOlistDREAM, integrFac=10)
```

This function return the same model with an additional field that is the weight assigned to each link, thus preparing it for the optimization (see next section).

Information derived from protein-protein networks (PIN) can also be used to differently prioritize integrated links, with the basic idea that a shorter path in the PIN corresponds to a more reliable link in the PKN. If *PPI* attribute is set to TRUE, the function *weighting* uses a PIN provided with the package to weight links (note that Uniprot IDs must be provided, see *UniprotIDdream* as an example of the list structure to be used).
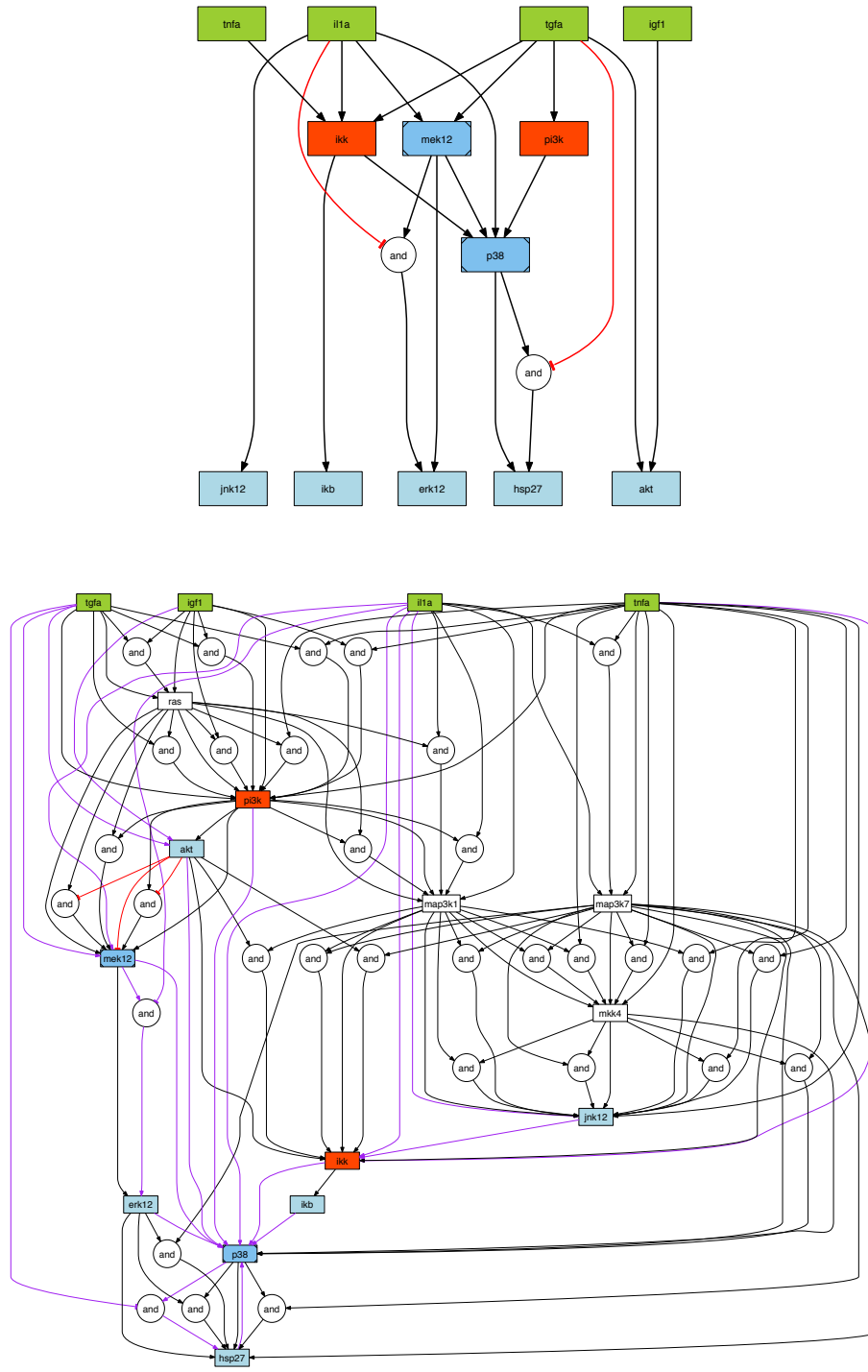
Figure 2: *Upper panel*: data-drive network (DDN) inferred using FEED method. *Lower panel*: integrated network, the prior knowledge network is intepreted as a logic model with all possible AND and OR gates, links added based on data-driven reverse-engineering methods are highlighted in purple.

```
> data(PPINigraph,package="CNORfeeder")
> data(UniprotIDdream,package="CNORfeeder")
> modelIntegrWeight <- weighting(modelIntegr=modelIntegr, PKNmodel=DreamModel,
+                                CNOlist=CNOlistDREAM, integrFac=10,
+                                UniprotID=UniprotIDdream, PPI=PPINigraph)
```

3.5   E. Training - CellNOptR

The integrated network can be optimized using *CellNOptR*; the function *gaBinaryT1W* of *CNORfeeder* package is a slightly modified version of *gaBinaryT1* function of package *CellNOptR* in order to take into account the different weigths that can be assigned to each link. A bipartite objective function is used to balance fit and size, that is to find the model which best describe data with the minumum number of links (see [3] and the vignette of *CellNOptR* for a comprehensive description).

```
> initBstring<-rep(1,length(modelIntegrWeight$reacID))
> # training to data using genetic algorithm (run longer to obtain better results)
> DreamT1opt<-gaBinaryT1W(CNOlist=CNOlistDREAM, model=modelIntegrWeight,
+                         initBstring=initBstring, maxGens=2, popSize=5, verbose=FALSE)
```

The optimal model and the simulated and real data (shown in Figure 3) can be visualized using the following commands of *CellNOptR*:

```
> # model
> plotModel(model=modelIntegrWeight, CNOlist=CNOlistDREAM, bString=DreamT1opt$bString)
> # data
> cutAndPlotResultsT1(model=modelIntegrWeight, CNOlist=CNOlistDREAM,
+                     bString=DreamT1opt$bString)
```
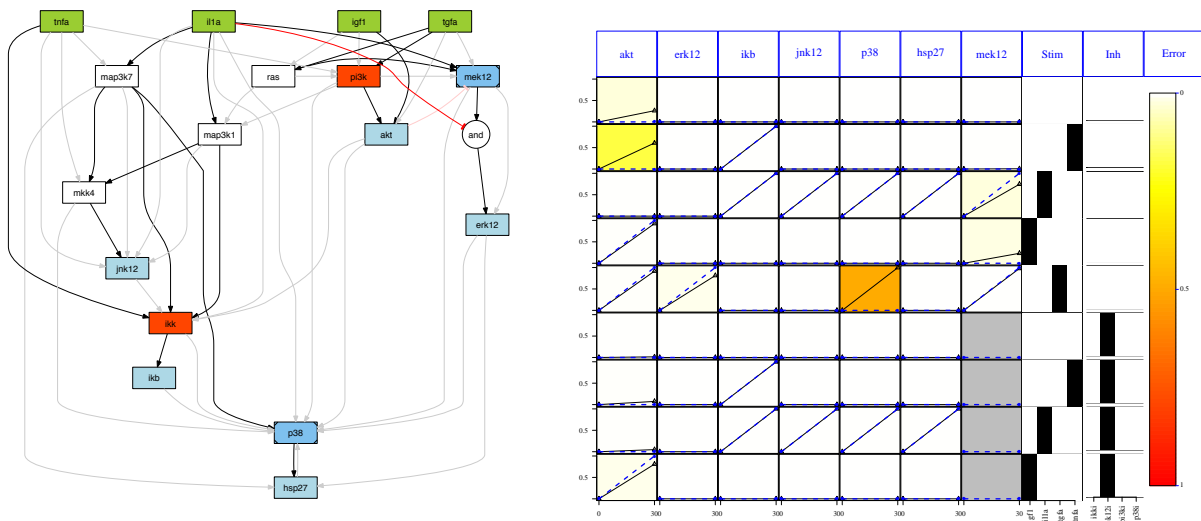

Figure 3: *Left panel*: optimal model. *Right panel*: fit to experimental data.

References

[1] F. Eduati, J. De Las Rivas, B. Di Camillo, G. Toffolo, J. Saez-Rodriguez. Integrating literature-constrained and data-driven inference of signalling networks. *Bioinformatics*, 28:18, 2311–2317, 2012.

[2] L.G. Alexopoulos, J. Saez-Rodriguez, B.D. Cosgrove, D.A. Lauffenburger, P. K. Sorger. Networks inferred from biochemical data reveal profound differences in toll-like receptor and inflammatory signaling between normal and transformed hepatocytes. *Molecular & Cellular Proteomics*, 9:9, 1849–1865, 2010.

[3] J. Saez-Rodriguez, L. Alexopoulos, J. Epperlein, R. Samaga, D. Lauffenburger, S. Klamt, P.K. Sorger. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology*, 5:331, 2009.

[4] C. Terfve, T. Cokelaer, D. Henriques, A. MacNamara, E. Goncalves, M. K. Morris, M. van Iersel, D. A. Lauffenburger, J. Saez-Rodrigues. CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst Biol*, 6:133, 2012.

[5] F. Eduati, A. Corradin, B. Di Camillo, G. Toffolo. A Boolean approach to linear prediction for signaling network modeling. *PLoS ONE*, 5(9): e12789, 2010.

[6] P. E. Meyer, F. Lafitte, G. Bontempi MINET: An open source R/Bioconductor package for mutual information based network inference. *BMC bioinformatics*, 9(1), 2008.