**André Catarino Guerra**

Bachelor in Cellular and Molecular Biology

# Genome-scale Metabolic Network Reconstruction of *Polaromonas sp.* strain JS666: Analysis of cDCE Degradation Rates and Design of Experiments for Bioremediation Improvement

A thesis submitted in conformity with the requirements for the degree

Master of Science in Biotechnology

| | |
|---|---|
| Supervisor: | Dr. Moritz von Stosch, Ph.D. Researcher, FCT-UNL |
| Co-supervisors: | Prof. Dr. Rui Oliveira, Associated Professor, FCT-UNL |
| | Dr. Anthony Danko, Ph.D. Researcher, FE-UP |

Jury:

| | |
|---|---|
| President: | Prof. Dr. Pedro Simões |
| Examiner: | Prof. Dr. Ana Teixeira |

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

**September, 2015**

**Genome-scale Metabolic Network Reconstruction of Polaromonas sp. strain JS666: Analysis of cDCE Degradation Rates and Design of Experiments for Bioremediation Improvement**

‘Su corazón fue un campo
Ebrio de hierbabuena
Joven de heridas
Joven de sueños
Retozaba esperanza
Y sabía reír
Embriagado de amor
en la madrugada de luz
Dijo adiós’
Olivia Guerra

# Acknowledgments

First of all, I would like to offer my gratitude towards my supervisor Dr. Moritz von Stosch for all the given help and guidance and towards Dr. Rui Oliveira for having me in the SBE group.

I also would like to thank everyone in the Systems Biology and Engineering group that helped me in this project. Especially, Rui Portela, Mauro Luis, Rodolfo Marques and Cristiana Azevedo. I'm grateful for all the given knowledge and free humor on the Office 334.

Also, I would like to demonstrate my gratitude to Dr. Anthony Danko for his interest on the continuity of my project.

In addition, I would like to thank my parents, friends and family, some of which reside no longer between us; rest in peace, Olivia.

# Abstract

Release of chloroethene compounds into the environment often results in groundwater contamination, which puts people at risk of exposure by drinking contaminated water. cDCE (cis-1,2-dichloroethene) accumulation on subsurface environments is a common environmental problem due to stagnation and partial degradation of other precursor chloroethene species. *Polaromonas* sp. strain JS666 apparently requires no exotic growth factors to be used as a bioaugmentation agent for aerobic cDCE degradation. Although being the only suitable microorganism found capable of such, further studies are needed for improving the intrinsic bioremediation rates and fully comprehend the metabolic processes involved. In order to do so, a metabolic model, iJS666, was reconstructed from genome annotation and available bibliographic data. FVA (Flux Variability Analysis) and FBA (Flux Balance Analysis) techniques were used to satisfactory validate the predictive capabilities of the iJS666 model. The iJS666 model was able to predict biomass growth for different previously tested conditions, allowed to design key experiments which should be done for further model improvement and, also, produced viable predictions for the use of biostimulant metabolites in the cDCE biodegradation.

**Keywords:** *Polaromonas* JS666 cDCE Biostimulation GEM FBA

# Resumo

A libertação no ambiente de compostos clorados derivados do etileno muitas vezes resulta em contaminação de águas subterrâneas, o que coloca as pessoas em risco de exposição por beber água contaminada. A acumulação de cDCE (cis-1,2-dicloroeteno) nas águas subsuperficiais é um problema ambiental recorrente, devido à estagnação e degradação parcial de outras espécies precursoras deste cloroeteno. A estirpe *Polaromonas sp*. strain JS666 aparentemente não requer factores de crescimento adicionais para ser utilizada na biodegradação aeróbia do cDCE. Apesar de ser o único microrganismo encontrado capaz de tal degradação, são necessários mais estudos para a melhoria das taxas de biorremediação intrínsicas e para compreender plenamente os processos metabólicos envolvidos. Com esse intuito, o modelo metabólico, iJS666, foi reconstruído a partir da prévia anotação do genoma e dados bibliográficos disponíveis. As técnicas de FVA (Análise à Variabilidade dos Fluxos) e FBA (Análise ao Balanço dos Fluxos) foram utilizadas para validar satisfatoriamente as capacidades preditivas do modelo iJS666. O modelo iJS666 foi capaz de prever o crescimento da biomassa em diferentes condições previamente testadas, permitiu projectar experimentos-chave que devem ser realizados para melhorar as capacidades predictivas do modelo e, também, preveu o uso de compostos viáveis para servirem de bioestimulantes na biodegradação cDCE.

**Palavras-chave:** *Polaromonas* JS666 cDCE Bioestimulação GEM FBA

# Table of Contents

# List of Figures

# List of Tables

# Abbreviations

TCE – trichloroethene

DCE – 1,1-dichlrorethene

cDCE – cis-1,2-dichloroethene

tDCE – trans-1,2-dichloroethene

VC – vinyl chloride

PVC – polyvinylchloride

USEPA – United States Environment Protection Agency

ETH – ethene

DNAPL – Dense Non-Aqueous Phase Liquid

MW – Molecular Weight

$K_h$ – Henry's constant

AOTC – Air Odor Threshold Concentration

MCL – Maximum Concentration Level

P&T – Pump and Treatment

rRNA – ribosomal ribonucleic acid

*k* – Specific substrate utilization rate

$K_s$ – Half-velocity transformation constant

kbp - $10^3$ base pairs

AkMO – alkene monooxygenase

CSIA – Coumpound-Specific Isotope Analysis

DCA – 1,2-dichloroethane

SJCA – St. Julien's Creek Annex

GENRES – Genome-scale Metabolic Network Reconstructions

BLAST – Basic Local Alignment Search Tool

PRIAM – Enzyme Specific Profiles for Genome Annotation

KEGG – Kyoto Encyclopedia of Genes and Genomes

BRENDA – BRaunschweig ENzyme DAtabase

GPR – Gene-Protein-Reaction Association

GEM – Genome-scale Model

ODE – Ordinary Differential Equations

FBA – Flux Balance Analysis

FVA – Flux Variability Analysis

COBRA – COnstraints Based Reconstruction and Analysis

NCBI – National Center for Biotechnology Information

NAD – Nicotinamide adenine dinucleotide

ATP – Adenosine triphosphate

GAM – Growth-Associated Maintenance

NGAM – Non-Growth-Associated Maintenace

Rs – Specific Substrate Consumption Rate

$RO_2$ – Specific Oxygen Consumption Rate

TCA – TriCarboxylic Acid cycle

Rb – Specific Biostimulant Consumption Rate

# 1. Introduction

## 1.1. Chlorinated Ethenes: A Worldwide Pollution Problem

The class of compounds known as chlorinated ethenes (chloroethenes or chloroethylenes) include all the molecules with an ethene backbone and one or more chlorines replacing the hydrogen positions. These include perchloroethene (PCE), trichloroethene (TCE), three forms of dichloroethene: 1,1-dichloroethene (DCE), cis-1,2-dichloroethene (cDCE), and trans-1,2-dichloroethene (tDCE), and vinyl chloride (VC). PCE, TCE and cDCE are worldwide for use as industrial solvents and degreasers and are particularly commonly used in dry cleaning and metal fabrication (USEPA, 2000). Likewise, in plastic industry, VC is produced for polymerization into the plastic polyvinylchloride (PVC) (Kielhorn *et al.,* 2000). Therefore, it is not surprising to find them among the "Top 25 Most Frequently Detected Priority Pollutants at Hazardous Waste Sites in North America and in Europe" (Alvarez & Illman, 2006) (See Table 1.1). The cleanup of these and other volatile organic compounds is estimated to cost "more than $45 billion dollars (1996 dollars) over the next several decades" (USEPA, 2000).

**Table 1.1.** The 25 Most Detected Pollutants at Waste Sites in North America and Europe. Adapted from Alvarez & Illman, 2006.

1. Trichloroethene (TCE)
2. Lead (Pb)
3. Perchloroethene (PCE)
4. Benzene
5. Toluene
6. Chromium
7. Dichloromethane (DCM)
8. Zinc
9. 1,1,1-Trichloroethane (TCE)
10. Arsenic
11. Chloroform
12. 1,1-Dichloroethene (DCE)
13. 1,2-Dichloroethene (cDCE)
14. Cadmium
15. Magnesium
16. Copper
17. 1,1-Dichloroethane (DCA)
18. Vinyl Chloride (VC)
19. Barium
20. 1,2-Dichloroethane (1,2-DCE)
21. Ethylbenzene
22. Nickel
23. Di(ethylhexyl)phthalate
24. Xylenes
25. Phenol

Frequently the chloroethenes are leaked into the ecosystem as PCE or TCE. On the anaerobic zones where they are introduced, different microorganisms from *Dehalobacter*, *Dehalospirillum*, and *Dehalococcoides* genera reduce those compounds into dehalogenated daughter products usually by using hydrogen, acetate or lactate as electron donors (Christ *et al.,* 2005). Although communities of these organisms have been shown to completely dehalogenate PCE and TCE to ethene (ETH), a nontoxic compound degradable by many other bacteria) this process is often stalled at the daughter products cDCE or VC usually due to insufficient supply of electron donor and/or inadequate microbial-community composition (Löffter *et al.,* 2013).

Chloroethenes have a higher density than water and therefore have the tendency to accumulate below the aquifer forming a Dense Non-Aqueous Phase Liquid (DNAPL). Since, they are also hydrophobic compounds usually they are adsorbed into soil particles and therefore have an increased recalcitrance. VC only occurs in the dissolved and gaseous phases is the only known the exception. This phenomenon is one of the main causes for the long term chloroethenes pollution in aquifers (Christ *et al.,* 2005). Release of these chloroethene compounds to the environment from leakage or improper disposal often results in groundwater contamination, which puts people at risk of exposure through their drinking water. Chlorinated ethenes have been shown to have serious liver and central nervous system effects, and several are proven carcinogens (USEPA, 2000). Some of the daughter products produced in the partial anaerobic dehalogenation (especially VC) are more toxic and carcinogenic than their precursors (Kielhorn *et al.,* 2000). For this reason, the USEPA has set very low limits for chlorinated ethene pollution in groundwater (USEPA, 2000), as reported in Table 1.2.

**Table 1.2.** Properties of Chlorinated Ethenes. Adapted from Löffter *et al.,* 2013; $K_h$ – Henry's Law Constant; AOTC – Air Odor Threshold Concentration; MCL – Maximum Concentration Level.

| Compound | MW [g/mol] | Liquid Density (20°C) [g/cm$^3$] | Vapour Pressure (20°C) [kPa] | Solubility (H20-20°C) [mM] | $K_h$ | AOTC [ppm, v/v] | MCL [mg/L] |
|---|---|---|---|---|---|---|---|
| PCE | 165.83 | 1.62 | 1.90 | 1.20 | 0.72 | 27 | 0.005 |
| TCE | 131.39 | 1.46 | 5.78 | 8.40 | 0.39 | 28 | 0.005 |
| cDCE | 96.94 | 1.25 | 24.00 | 36.10 | 0.17 | - | 0.07 |
| tDCE | 96.94 | 1.26 | 35.30 | 65.00 | 0.38 | 17 | 0.10 |

## 1.2. (Bio)remediation of Chlorinated Ethenes

In order to remediate the contaminated groundwater of the affected sites several different treatment technologies were tested over the years. Pump and Treatment (P&T) was initially used along with the reduction of the chloroethenes. This process involves pumping large volumes of groundwater out of the sediment, followed by on-site treatment with chemical oxidation or adsorption onto activated carbon, then return of the treated water to the ground. Although this technology was able to decontaminate dissolved chloroethenes, long operational periods were often needed in order to allow dissolution of this recalcitrant compounds at great monetary expense (De Wildeman & Verstraete, 2003) and still residual contamination would be present in the ecosystem (Christ *et al.,* 2005).

Permeable reactive barriers have also been used to treat chlorinated ethene pollution by reductive dechlorination to ethene, a more benign compound that occurs naturally in the environment. These barriers consist of trenches filled with an electron donor matrix (such as zero-valent iron or zinc) that will reduce the contaminants as the groundwater plume flows through. Although, the degradation efficiency of this using chemical donors could exceed 85% for cDCE (Mahendra *et al.,* 2007) and had the advantage of lower operational costs than pump-and-treat remediation, some problems arise when geological site singularities make the construction of a barrier difficult or when long-term efficacy of this strategy with the less oxidized of the chlorinated ethenes (cDCE and VC) decreases (De Wildeman & Verstraete, 2003).

Of the technologies used to remediate contaminated sites, in situ bioremediation is recognized as being the most promising and cost-effective solution. Bioremediation, the removal or detoxification of xenobiotic compounds by living organisms, is currently a popular and proven strategy for sites contaminated with chlorinated ethenes in a variety of ways (Löffter *et al.,* 2013).The most common bioremediation strategy takes advantage of the ability of some microbes to reductively dechlorinate these compounds sequentially in a similar way to the permeable reactive barriers action. As referred before, microbes capable of dehalogenate PCE and TCE are fairly common, but only members of the genus *Dehalococcoides* have been shown to achieve complete reductive dechlorination to ethene and chloride ion (Dworkin *et al*., 2001). Subsequently after the identification of this microorganisms and their potential role in biodegradation, a huge number of bioaugmentation applications were executed worldwide (See Figure 1.1).

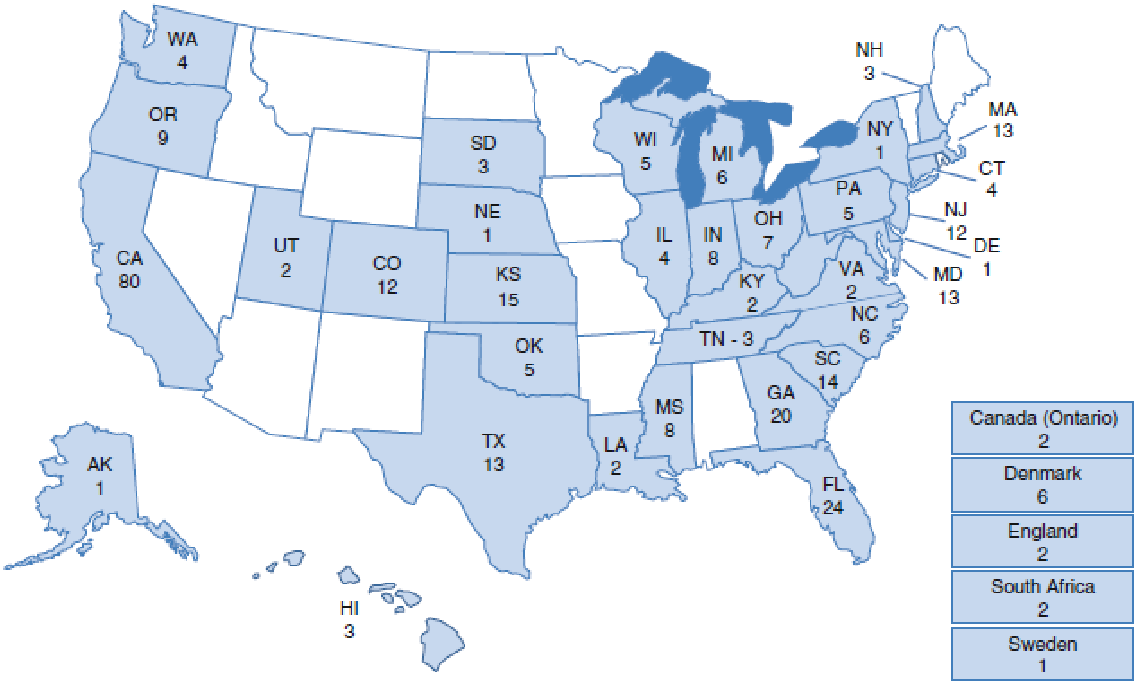


**Figure 1.1. Bioaugmentation using Dehalococcoides spp. for site cleanup. Adapted from Lyon & Vogel, 2013.**

Complete biological reductive dechlorination (dehalorespiration) therefore fundamentally requires the presence of Dehalococcoides (increasing cell numbers in soil by bioaugmentation), a sufficient amount of electron donors and an anaerobic environment. The absence of any one of these prerequisites will result in partial reductive dechlorination, as denoted before. In practice, this reductive bioremediation of chlorinated ethene plumes frequently stalls, resulting in accumulation of cDCE (the most common form of DCE) and/or VC (DiStefano *et al.,* 1991), see Figure 1.2.

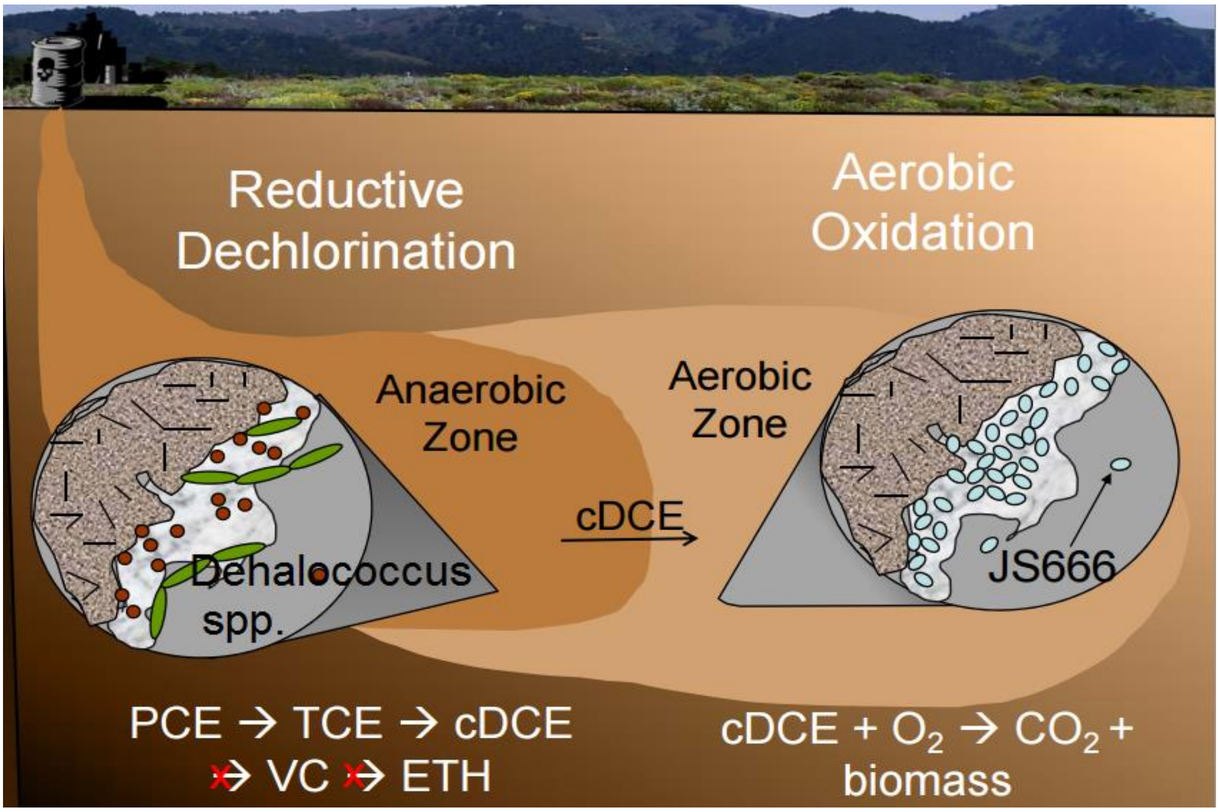


**Figure 1.2. Representation of chloroethene groundwater contamination. Adapted from Jennings, 2008.**

Although the by-products from of the initial dehalogenation (cDCE and VC) can migrate into aerobic plume zones and be totally mineralized by many aerobic co-metabolizing bacteria holding nonspecific monooxygenases, these processes are generally co-metabolic, since those microorganisms are unable to convert the product of this oxidation into a useful compound for the cell and therefore require presence of co-substrates (i.e. methane or toluene as carbon sources) to survive (McCarty *et al*., 1998; Verce *et al*., 2000; Bradley, 2003; Sun *et al*., 2010). In many cases where aerobic co-metabolism has been observed in the field, it was generally an unexpected occurrence at the edges of plumes where the systems had become aerobic, and there was still need for the presence of other contaminants/carbon sources in order to maintain the main metabolism (Bradley, 2003). Usually, the bioremediation of groundwater with those endogenous microorganisms was achieved supplementing additional oxygen since the degradation of the co-substrates could otherwise cause the site to become anaerobic (Hopkins & McCarty, 1995). Co-metabolic VC-oxidizing and cDCE-oxidizing bacteria have been used successfully for bioremediation, but have several limitations (Van Hylckama Vlieg *et al*., 1996; McCarty *et al*., 1998; Van Hylckama Vlieg *et al*., 1998). For instance, co-metabolic degraders can encounter toxicity problems due to accumulation of the mutagenic chlorinated epoxide by-products (Alvarez-Cohen & Speitel, 2001; Bradley, 2003) and they exhibit slower degradation rates due to the competition between chlorinated ethene and carbon-source substrate for the monooxygenase active site (Van Hylckama Vlieg *et al*., 1996).

An alternative to co-metabolic oxidation of chlorinated ethenes is assimilative oxidation, where chlorinated ethenes are used simultaneously as carbon and energy source. Several microbial species were isolated from soil that are capable of VC-assimilation (Hartmans *et al.,*. 1985; Verce *et al*., 2000; Verce *et al.*, 2001), but early work prospecting contaminated aerobic plumes showed that indigenous microorganisms in black-water stream sediments were also capable of aerobic assimilative oxidation of cDCE without any additional co-substrates, yet no causative organism was initially isolated (Bradley & Chapelle, 2000). More recently, Coleman *et al*. (2002a and 2002b) sought after aerobic bacteria that use VC and cDCE as sole carbon and energy sources. In that work 12 isolates were produced that achieved autonomous growth in a medium with VC as the only carbon and energy source, corroborating that such microbes are commonly found in the aerobic zones of VC-contaminated plumes. Also as a result of that work, only one isolate microorganism, *Polaromonas* sp. strain JS666, was able to aerobically oxidize cDCE as sole carbon and energy source (Coleman *et al.*, 2002b).

## 1.3. *Polaromonas* sp. strain JS666

*Polaromonas* sp*.* strain JS666 apparently requires no exotic growth factors, it is considered a promising bioaugmentation agent for aerobic sites where cDCE has accumulated since this accumulation in aerobic subsurface environments is a common problem in the remediation of contaminated sites where other chloroethenes were previously partially degraded (Bradley, 2003). Also, aerobic remediation might be preferred over anaerobic reductive dechlorination in situations where the cDCE concentration is low (but still above maximum concentration limit) due to the co-metabolic competition previously referred; where the aquifer is partial or fully aerobic since anaerobic microorganisms could not endure this conditions; and/or where the byproducts of anaerobic biological activity (methane, sulfides, reduced iron, etc.) are adverse to *Polaromonas* sp. strain JS666 growth (Giddings *et al*., 2010a).

*Polaromonas* sp. strain JS666 is a member of the family *Comamonadaceae* in the *β*-proteobacteria class. This Gram-negative bacteria is a yellow-colored, devoid of vacuoles, non-motile, psychrotrophic with an optimal growth temperature around 20°C (Coleman *et al*., 2002a). It was first discovered in activated-carbon of a P&T plant being used to degrade PCE, TCE and cDCE in Dortmound, Germany (Coleman *et al*., 2002a).  Based on analysis of 16S rRNA sequences, the *Polaromonas* sp. strain JS666 most closely relates with the psicotrophic arsenite oxidizing isolate *Polaromonas* sp. strain GM1, with a 98% sequence identity (Osborne *et al*., 2010), 97,9% sequence identity with *Polaromonas* vacuolata (Coleman *et al.*, 2002b) and having a 97% sequence identity with *Polaromonas* naphthalenivorans strain CJ2, (Jeon, 2006), see Figure 1.3.

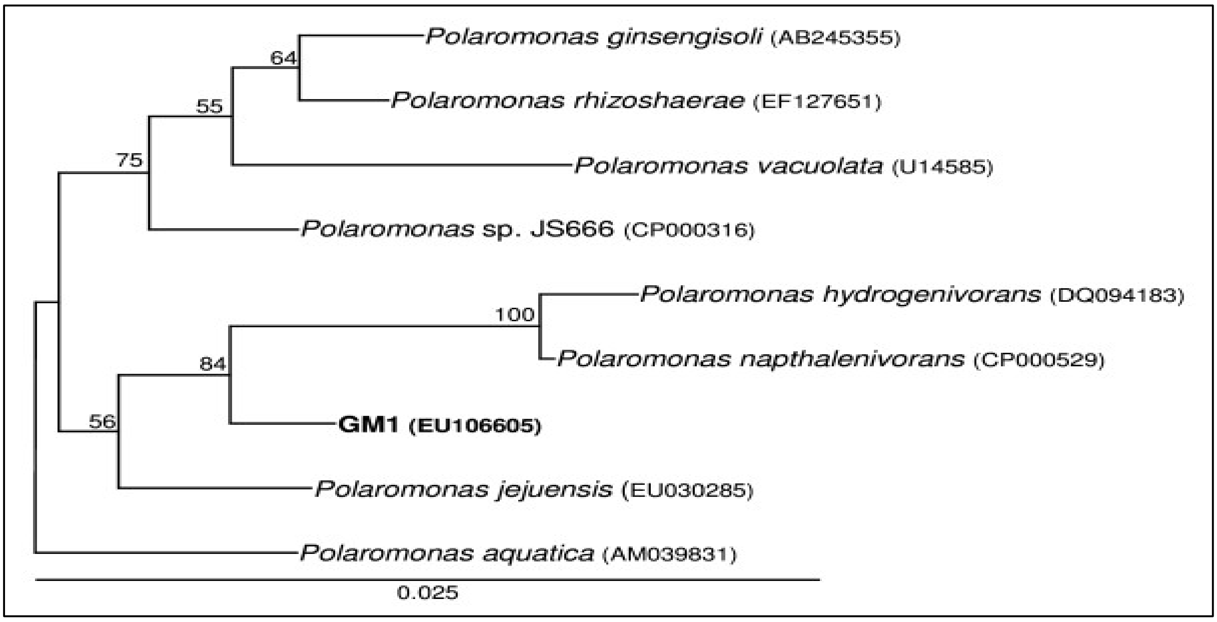

Polaromonas ginsengisoli (AB245355)
Polaromonas rhizoshaerae (EF127651)
Polaromonas vacuolata (U14585)
Polaromonas sp. JS666 (CP000316)
Polaromonas hydrogenivorans (DQ094183)
Polaromonas napthalenivorans (CP000529)
GM1 (EU106605)
Polaromonas jejuensis (EU030285)
Polaromonas aquatica (AM039831)
64
55
75
100
84
56
0.025

**Figure 1.3.** 16S rRNA phylogenetic tree of Polaromonas spp. Bootstrap values for 100 trials and the percentage of variation in sequence identity are both shown in the figure. Adapted from Osborne et al., 2010.

Recently it was found that *Polaromonas* is one of the most commonly occurring genus in granular activated carbon filters used to treat surface and ground water (Magic-Knezev *et al.*, 2009). This is consistent with the common observation of *Polaromonas* species in extremely oligotrophic environments (Page *et al*., 2004; Loy *et al*., 2005; Kämpfer *et al*., 2006). In addition to that, *Polaromonas* species tend to be slow-growing and psychrotolerant, which hamper their isolation from environmental samples by ordinary methods (Irgens *et al*., 1996; Darcy *et al*., 2011; Margesin *et al*., 2012). The strain JS666 is no exception, having long latency times and a doubling time of 74 ± 8 hours being very difficult to cultivate *in vitro*. The strain number "666" praises those characteristics. (Coleman *et al.*, 2002b).

The growth yield of *Polaromonas* sp. strain JS666 on cDCE is 6.1 ± 0.4 g protein/mole cDCE, which is comparable to VC-assimilating bacteria, despite the lower energy content of cDCE (Coleman *et al*., 2002b). A maximum specific substrate utilization rate (*k*) of 12.6 ± 0.3 nmol/min/mg protein and a half-velocity constant for cDCE transformation (Ks) of 1.6 ± 0.2 μM were determined (Coleman *et al*., 2002b). Moreover, cDCE was degraded routinely to below 0.03 μg/L by *Polaromonas* sp. strain JS666 (Coleman *et al*., 2002b). So, in the context of bioaugmentation or natural attenuation potential, the *Polaromonas* sp. strain JS666 have the capability to extract enough energy from cDCE for reasonable growth and simultaneously degrade cDCE to a concentration below the MCL (see Table 1.2.) without significant effects on substrate utilization rate (Coleman *et al*., 2002a; Verce *et al*., 2001).

Elucidation of the metabolic pathway responsible for cDCE assimilation in *Polaromonas* sp. strain JS666 would provide important insights into the use of this bacteria as a bioaugmentation agent. Due to the importance and uniqueness of the *Polaromonas* sp. strain JS666 in biodegradative processes, an effort to completely sequence the genome of this *Polaromonas* specie was necessary (Mattes *et al*., 2008). The genome of *Polaromonas* sp. strain JS666 contains many mobile genetic elements and evidences of putative horizontal gene transfer, including two plasmids (pPOL338 and pPOL360 with 338 and 360 kbp, respectively) (Mattes *et al*., 2008). A putative haloalkane biodegradation gene cluster is present in nearly identical copies on the chromosome and the 360 kbp plasmid within a 9.9 kbp duplicated region. This duplication may have been mediated by two nearby transposases, and the increase in gene dosage could have played a role in the adaptation of *Polaromonas* sp. strain JS666 to growth on chlorinated alkenes or aromatics (Mattes *et al*., 2008). Many other catabolic genes are found closely associated with transposable elements throughout the genome, indicating recent acquisition and/or rearrangement of genes necessary for the degradation of various xenobiotic compounds, including alkanes, cycloalkanes, and cyclic alcohols (Mattes *et al*., 2008).

While the specifics remain to be determined, it appears likely that the acquisition of the two plasmids by *Polaromonas sp.* strain JS666 was a major step in the evolution of its cDCE assimilating capability. This hypothesis is supported by the observation that these two plasmids have been maintained by the strain throughout enrichment, isolation, and many generations of growth in the laboratory, suggesting that they are necessary for growth on cDCE (Coleman *et al.*, 2002b; Mattes *et al.*, 2008). The cDCE degrading phenotype has also been found to be unstable, which would be consistent with plasmid or transposon-carried genes, but may also be due to imperfect regulatory control of the newly constructed pathway (Alexander, 2010).

Concerning the cDCE degradative pathway(s), due to the structural similarities between cDCE, VC, and ethene, *Polaromonas* sp. strain JS666 was expected to grow on both VC and ethene. VC-assimilating microbes oxidize the VC molecule by an alkene monooxygenase (*AkMO*), then this chlorinated epoxide metabolite is conjugated with coenzyme M and incorporated into the metabolism of the cell (Mattes *et al.*, 2010). Therefore, epoxidation of cDCE by a monooxygenase was a reasonable hypothesis supported by the observation that this microorganism also produces epoxyethane from ethene at an increased rate after growth on cDCE (Coleman *et al.*, 2002a). However, no homologue of the typical downstream epoxyalkane coenzyme M transferase was identified in the *Polaromonas* sp. strain JS666 genome.

An integrated '*omics'* study was conducted using proteomic mass spectrometry, microarray techniques, CSIA (Compound-Specific Isotope Analysis) and enzyme assays in order to establish the cDCE degradative pathway (Jennings *et al*., 2009). This approach revealed upregulated genes of *Polaromonas* sp. strain JS666 bacteria grown in cDCE. They included genes for cyclohexanone monooxygenase, glutathione-*S*-transferase, cytochrome P450 and genes for (di)chloroacetaldehyde, (di)chloroacetate, and (chloro)glycolate transformation were also upregulated in a pattern expected for growth on *c*DCE or 1,2-dichloroethane (DCA). The results of both molecular techniques and CSIA suggested that cDCE degradation via monooxygenase catalyzed epoxidation (teorically achieved by cyclohexanone monoxygenase, *Bpro_5565*) may be only a minor pathway and that the initial step in the major cDCE degradation pathway involves carbon-chloride bond cleavage due to the isotope fractionation pattern obtained in the CSIA, most likely to be a glutathione-S-transferase catalyzed initial dehalogenation reaction. Further research was needed to identify the functional activity of upregulated enzymes and to identify their roles in the cDCE degradation pathways of this unique *Polaromonas* specie (Jennings *et al*., 2009).

In the research made by Nishino *et al*., in 2013, several upregulated enzymes of *Polaromonas* sp. strain JS666 were tested in order to clarify their roles. Several lines of evidence indicate that cytochrome P450 monooxygenase (*Bpro_5301*) is responsible for the initial steps in *c*DCE biodegradation because *c*DCE was degraded only in the presence of oxygen, degradation was inhibited by cytochrome P450-specific inhibitors, heterologously expressed cytochrome P450 monooxygenase catalyzes the transformation of *c*DCE to dichloroacetaldehyde (Alexander, 2010), and *Bpro_5301* gene was upregulated 3.5-fold by cDCE accordingly to the findings of Jennings *et al*., 2009. Probably, the glutathione-S-transferase, upregulated 99.8-fold on cDCE medium, participates in the detoxification of the minor cDCE-epoxide compound produced by cytochrome P450 or in the detoxification of the (di)chloroacetaldehyde produced in the main degradation pathway (Cox, 2012; Nishino *et al*., 2013), as displayed in Figure 1.4.

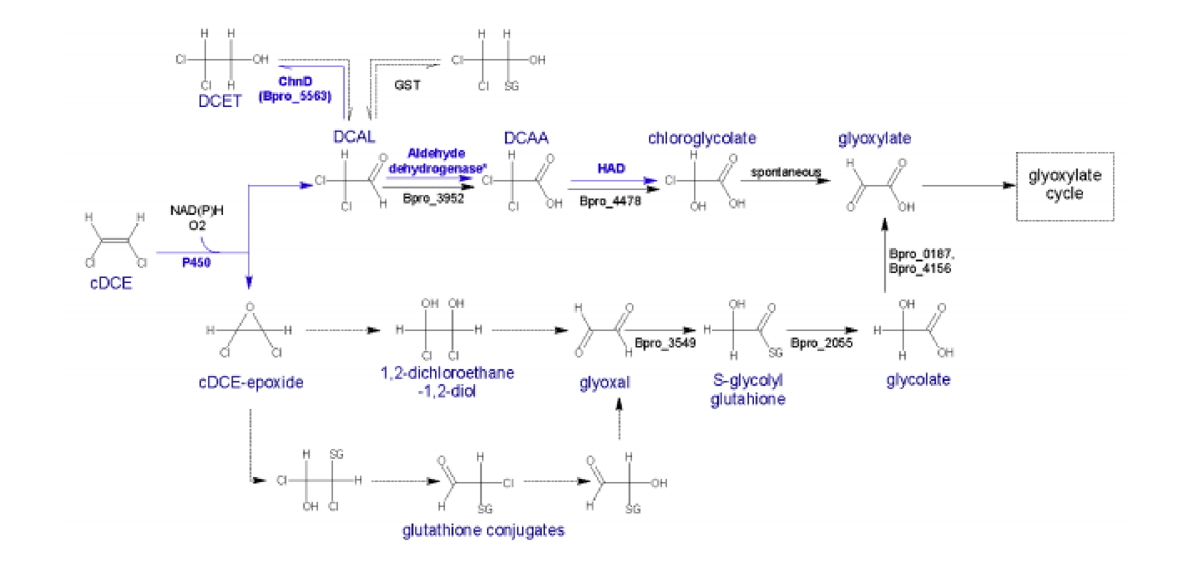


**Figure 1.4.** Simplified pathway(s) of cDCE degradation in Polaromonas sp. strain JS666. Blue arrows represent pathways that were confirmed with biochemical evidence. Black arrows indicate pathways supported by iTRAQ study. Dotted arrows indicated proposed and not yet confirmed reactions. Genes involved in the reactions are represented bellow the arrows and the generic terms on top. From Cox, 2012.

A recent study found that *Polaromonas* sp. strain JS666 degraded cDCE in a microcosms constructed with contaminated sediment and groundwater, even when presented with alternative carbon sources or competitive/predatory microbes (Giddings *et al*., 2010a; Kurt *et al*., 2014). This activity was also reliably correlated with the abundance of a developed DNA probe for the JS666 chromosomal gene encoding isocitrate lyase, an enzyme apparently not directly associated to cDCE degradation (Giddings *et al*., 2010b). Additionally, a pilot study using *Polaromonas* sp. strain JS666 as a bioaugmentation agent began in October, 2008, at St. Julien's Creek Annex (SJCA), Chesapeake, Virginia, USA, predicted that 100 L of inoculum culture ($OD_{600}$=1.0) should be able to treat a 10x30×80 meters plot (24,000 square meters) within 2 months after inoculation, a clear demonstration of *Polaromonas* sp. strain JS666 biodegradative potentiality (Giddings *et al.*, 2010a).

## 1.4. Genome-scale Metabolic Modeling

### 1.4.1. GENRES: Genome-scale Metabolic Network Reconstructions

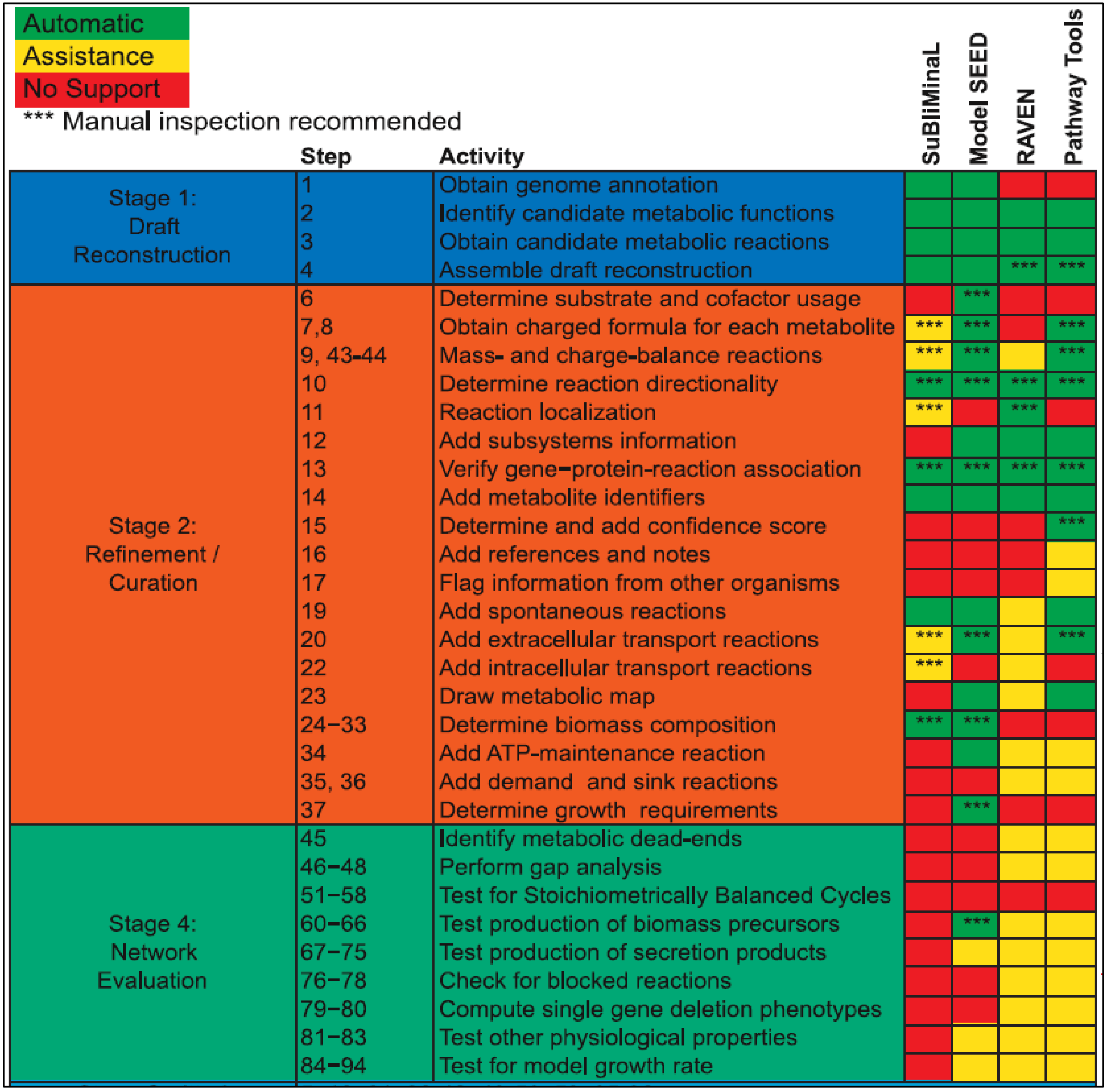
With the recent growth in genomics research, sustained by the decreasing DNA sequencing prices, complete genomic sequences of a multitude of species are assembled at an unprecedented rate (Wetterstrand, 2015) Therefore, it's evident that full comprehension of encoded functionality is displaced from that increased knowledge rate. A perfect example of this discrepancy is evident for *Polaromonas sp*. strain JS666 whose genome is completely sequenced although few biochemical evidences were further obtained (Mattes *et al*., 2008; Jennings *et al*., 2009).

Genome-scale metabolic network reconstructions (GENRES) try to achieve a complete understanding of the metabolic features of an organism by assembly a network of metabolic reactions catalyzed by enzymes and transporters found on the annotations made for the genome sequence (Palsson, 2009). Such gene annotations are often generated by applying prior knowledge to the genomic sequence using automated algorithms, like for example BLAST, which compares the sequence similarity to previously known enzyme sequences and identifies best matching homologs or PRIAM, which attempts to fit a novel sequence into position-specific enzyme profiles based on the discovered domains.

There are several public domain database's that are capable of incorporating these organism-specific gene annotations and the biochemical functionality of the encoded enzymes, in a manual or automatic way. The ultimate objective of different databases, like KEGG, BRENDA and MetaCyc, is the association between different *omics* information. Since the main objective is the reconstruction of metabolic networks usually the data is classified and stored in hierarchical GPR (Gene-Protein-Reaction) associations although the storage of transcriptional and regulatory associations/networks in many cases are also included. Although all those databases could be used in the reconstruction of metabolic networks they have very different proprieties and incorporated information: KEGG does not provide reliable gene annotations for metabolic network reconstruction purposes (Green & Karp, 2006) but have some of the most complete pathway schemes and can still serve as a reference framework for network modeling and network gap filling; BRENDA (BRaunschweig ENzyme DAtabase) is an enzyme database which contains manually curated data from organism-specific enzyme assays or protein structure studies (Scheer *et al*., 2011) and therefore is suitable for high confidence reconstructions although the overall association coverage is very low for a single specific organism; MetaCyc is a metabolic network database with more than 2260 pathways from 2600 different organisms having a description of every reaction, metabolite, gene association on each pathway and also is complemented with literature citations. The pathways are hierarchized accordingly to phyla in superpathways that decomposed in smaller organism specific pathways. In the study made by Wittig & Beuckelaer, 2001, a complete review and comparison between the advantages and potentialities of different databases in the reconstruction of metabolic networks were assessed.

With the increasing number of whole sequenced genomes, there was also an increasing number of newly annotated genes and pathways discovered. The number of GENRES is therefore also increasing although at much slower rate (Palsson, 2009). In order to help researcher in the infamous task of building a genome-scale metabolic networks, automated reconstruction software packages were developed. These bioinformatics tools retrieve the stored information in the previously described databases in order to assemble automatically a draft reconstruction of the metabolism of a specific organism. Some software like Model SEED and PathwayTools have the ability of automatically predict the GPRs based solo in the annotation (Karp *et al*., 2011), filling up network gaps (Karp *et al*., 2009) and even converting the network to functional models (Karp *et al*., 2009; Overbeek *et al*., 2014). Hamilton & Reed, in 2014, made a review that reported the principal functionalities of the most commonly used prediction software, as shown in Table 1.3.

**Table 1.3.** Visual comparison between some of the most used GENRES predictive software packages. The displayed stages and steps arise from Thiele & Palsson, 2010. From Hamilton & Reed, 2014.



🟩 Automatic
🟨 Assistance
🟥 No Support
*** Manual inspection recommended

| | Step | Activity | SuBliMinaL | Model SEED | RAVEN | Pathway Tools |
|---|---|---|---|---|---|---|
| Stage 1: Draft Reconstruction | 1 | Obtain genome annotation | 🟩 | 🟩 | 🟥 | 🟥 |
| | 2 | Identify candidate metabolic functions | 🟩 | 🟩 | 🟩 | 🟩 |
| | 3 | Obtain candidate metabolic reactions | 🟩 | 🟩 | 🟩 | 🟩 |
| | 4 | Assemble draft reconstruction | 🟩 | 🟩 | 🟩 *** | 🟩 *** |
| Stage 2: Refinement / Curation | 6 | Determine substrate and cofactor usage | 🟥 | 🟩 *** | 🟥 | 🟥 |
| | 7,8 | Obtain charged formula for each metabolite | 🟨 *** | 🟩 *** | 🟥 | 🟩 *** |
| | 9, 43-44 | Mass- and charge-balance reactions | 🟨 *** | 🟩 *** | 🟨 | 🟩 *** |
| | 10 | Determine reaction directionality | 🟩 *** | 🟩 *** | 🟩 *** | 🟩 *** |
| | 11 | Reaction localization | 🟨 *** | 🟥 | 🟩 *** | 🟥 |
| | 12 | Add subsystems information | 🟥 | 🟩 | 🟩 | 🟩 |
| | 13 | Verify gene-protein-reaction association | 🟩 *** | 🟩 *** | 🟩 *** | 🟩 *** |
| | 14 | Add metabolite identifiers | 🟩 | 🟩 | 🟩 | 🟩 |
| | 15 | Determine and add confidence score | 🟥 | 🟥 | 🟥 | 🟩 *** |
| | 16 | Add references and notes | 🟥 | 🟥 | 🟥 | 🟨 |
| | 17 | Flag information from other organisms | 🟥 | 🟥 | 🟥 | 🟨 |
| | 19 | Add spontaneous reactions | 🟩 | 🟩 | 🟨 | 🟩 |
| | 20 | Add extracellular transport reactions | 🟨 *** | 🟩 *** | 🟨 | 🟩 *** |
| | 22 | Add intracellular transport reactions | 🟨 *** | 🟥 | 🟨 | 🟥 |
| | 23 | Draw metabolic map | 🟥 | 🟩 | 🟨 | 🟩 |
| | 24-33 | Determine biomass composition | 🟩 *** | 🟩 *** | 🟥 | 🟥 |
| | 34 | Add ATP-maintenance reaction | 🟥 | 🟩 | 🟨 | 🟨 |
| | 35, 36 | Add demand and sink reactions | 🟥 | 🟥 | 🟨 | 🟨 |
| | 37 | Determine growth requirements | 🟥 | 🟩 *** | 🟥 | 🟥 |
| Stage 4: Network Evaluation | 45 | Identify metabolic dead-ends | 🟥 | 🟥 | 🟨 | 🟨 |
| | 46-48 | Perform gap analysis | 🟥 | 🟥 | 🟨 | 🟨 |
| | 51-58 | Test for Stoichiometrically Balanced Cycles | 🟥 | 🟥 | 🟥 | 🟥 |
| | 60-66 | Test production of biomass precursors | 🟥 | 🟩 *** | 🟨 | 🟨 |
| | 67-75 | Test production of secretion products | 🟥 | 🟨 | 🟨 | 🟨 |
| | 76-78 | Check for blocked reactions | 🟥 | 🟥 | 🟨 | 🟨 |
| | 79-80 | Compute single gene deletion phenotypes | 🟥 | 🟥 | 🟨 | 🟨 |
| | 81-83 | Test other physiological properties | 🟥 | 🟨 | 🟨 | 🟨 |
| | 84-94 | Test for model growth rate | 🟥 | 🟨 | 🟨 | 🟨 |

However, since many of the functions of gene products are predicted from prior knowledge derived from orthologous genes, some predictions result in unannotated or even missannotated enzymes (Schnoes *et al*. 2009). Even if the enzyme annotation is correct, some of those enzymes are able to use different substrates simultaneously, challenging the identification of their function inside the cell. Also, with the increasing number of annotated genes on those databases the predictions made by software that use those same databases have lesser confidence about specific functionality. For example, PathwayTools (more specifically PathoLogic) have included predictive algorithms based on the phylogeny of the organisms to sort out false positives reactions from the metabolic reconstruction (Karp *et al*., 2009). Simultaneously, with the increased interest in systems biology and especially in the genome-scale metabolic modelling, illustrative high quality protocols were developed in order to help researchers in the manual curation stage necessary to debug the obtained draft network (Thiele & Palsson, 2010).

Currently produced genome-scale reconstructions usually have a genome coverage around 20%, due to the large percentage (30-40%) of hypothetical proteins unannotated in the genome (Wittig & De Beucklelaer, 2001) and due to the fact that many of the predicted coding sequences belong to non-physiologically relevant proteins. That means that it's necessary to complement the initial list of enzymes with other sources of biochemical knowledge, in order to fill gaps, add new pathways that confer specific attributes or to have a globally higher confidence level in the reconstructed metabolic network. In instances where the metabolic network model is missing one or more reactions to complete a metabolic pathway, the researcher has to decide whether to include a biochemical reaction lacking any source of evidence.

### 1.4.2. GEMS: a Constraint-based Approach

A genome-scale metabolic network can be adapted into a mathematical model, GEM (GEnome-scale Model), in order to simulate biological behavior (Palsson *et al*., 2006). One approach to metabolic modeling is the use of ordinary differential equations (ODEs) to evaluate the thermodynamic equilibrium between the metabolites of each reaction in the model. However, such approach is almost unfeasible for modeling a large complex system with little *a priori* knowledge, since it requires a large amount of pre-determined kinetic parameters and/or intracellular concentration measures. Some improvements on predicting thermodynamic parameters of those reactions were achieved using algorithms that incorporate group-contribution theory to calculate the reactional Gibbs free energy (Feist *et al*., 2007).

Another approach, more suitable for genome-scale metabolic modeling, is constraint-based modeling, which imposes zero-order kinetic constraints (mass balanced reactions) to limit the possible behaviors of a reactional event and simultaneously optimizes for the maximization of a flux rate or a metabolite production, usually biomass for growth (Thiele & Palsson, 2010). Although all the work developed in this study is based on the genome-scale metabolic modeling, due to the vast amount of bibliographic information already published related to the constrain-based methodologies applied for analyzing those models (Palsson, 2006), the decision of not include a complete explanatory review over those techniques in this dissertation was made since they are not the main focus of this work. Instead, a brief review on the used methodologies - Flux Balance Analysis (FBA) and Flux Variability Analysis (FVA) - is hereby given;

- FBA is a widely used technique for constraint-based modeling of metabolic networks. It defines each enzymatic and transport reaction in the metabolic network as a flux, and computes a pseudo steady-state distribution of the flow of metabolites within constrained flux bounds set for the model, thereby allowing the researcher to examine the interdependency of various metabolic pathways from a systems perspective (Palsson *et al*., 2006; Orth *et al*., 2010).

The first step in a FBA is to mathematically represent metabolic reactions. The core feature of this representation is in the form of a stoichiometric matrix (**S**) of size **m** × **n**. Every row of this matrix represents one unique compound in each compartment (for a system with **m** compounds) and every column represents one reaction (for a system with **n** reactions). The entries in each column are the stoichiometric coefficients of the metabolites participating in a reaction being a negative coefficient for every metabolite consumed and a positive coefficient for every metabolite that is produced, assuming that substrates are in the left side and products on the right side of the balanced equation (Orth *et al*., 2010).

The matrix of stoichiometries imposes mass-balanced fluxes in the system, ensuring that the total amount of any compound being produced must be equal to the total amount being consumed at the pseudo steady-state. The other constraint of this model defines the space of allowable flux distributions of a system—that is, the rates at which every metabolite is consumed or produced by each reaction.

Therefore, to every reaction will be given upper bound (**ub**) and lower bound (**lb**), which define the maximum and minimum allowable fluxes of the reactions, respectively. In irreversible reactions, the minimum flux is always set to zero. The flux through all of the reactions in a network is represented by the vector **v**, which has a length of ***n***. The system of mass balance equations at a pseudo steady-state will assume that **x**, the vector of metabolite concentrations, will not have any change over time since the production and consumption of all metabolites are balanced (Orth *et al*., 2010).

$$\frac{d\mathbf{x}}{dt} = \mathbf{0}$$

In the biological system's level, that means;

$$\mathbf{S} \,.\, \mathbf{v} = \mathbf{0}$$

In any realistic large-scale metabolic model, there are more reactions than metabolites (***n*** > ***m***) since different reactions can use the same metabolites. So, like any other linear system, where there is present a higher number of variables than equations, there is no unique solution to this defined system. Even though the previously described constraints define a range of different possible solutions, it is still viable to identify and analyze single points within the solution space. As the basis of the FBA method, the identification of such interest point within a constrained space is achievable by the maximization or minimization of a specific objective function. The objective function ***Z***, which can be any linear combination of fluxes, where **c** is a vector of weights indicating how much each reaction (i.e. biomass reaction when simulating maximum growth) contributes to the objective function. If the reaction to be maximized already include all the metabolites that should be maximized then **c** will be a vector of zeros with a positive entry equal to 1 in the column of the reaction to be maximized. This is the case when the biomass components are displayed in the same equation – the biomass equation.

$$\mathbf{Z} = \mathbf{c}^{\boldsymbol{T}}.\mathbf{v}$$

In this sense, the output of FBA is a particular flux distribution, **v**, which maximizes or minimizes the objective function, can be displayed as linear problem;

$$\mathbf{maximize}\ (\mathbf{Z})$$

$$\mathit{Subject\ to}:$$

$$\mathbf{S}\,.\,\mathbf{v} = \mathbf{0}$$

$$\boldsymbol{lb_i} \leq \mathbf{v}_i \leq \boldsymbol{ub_i}\ \ ,\ \ \mathbf{1} \leq \boldsymbol{i} \leq \boldsymbol{n}$$

In order to solve this equation system with many variables, the use of dedicated computational linear solvers is needed. Several system biology software that included dedicated solvers are available to the public, but the COBRA (COnstraint-Based Reconstruction and Analysis) toolbox is one of the most popular and includes methods to simulate, analyze and predict a variety of metabolic phenotypes, network gap filling, $^{13}$C analysis, metabolic engineering, *omics*-guided analysis and network visualization (Schellenberger *et al*., 2011). Over the years, new methodologies for analyzing genome-scale metabolic networks were developed based on FBA. Some of those are gene deletion studies (OptKnock), minimization of metabolic adjustment (MOMA), dynamical FBA (dFBA), parsimonious FBA (pFBA), robustness analysis and FVA, being the last one also used in this dissertation and described below (Palsson, 2006; Schellenberger *et al*., 2011);

- FVA is used to find the minimum and maximum flux for reactions in the network while maintaining some pre-established state of the network. Applications of FVA for molecular systems biology include the exploration of alternative optimal solutions, studying flux distributions under suboptimal growth, investigating network flexibility and network redundancy for example. FVA starts as a regular FBA, maximizing or minimizing a particular objective function, but then uses the achieved objective flux(es) value(s) as a fixed (optimal (***γ***=1***)*** or suboptimal (***γ*** <1)) constraint and executes simultaneously a minimization and maximization of each remaining individual fluxes in order to predict their variability in the system, as described in Gudmundsson *et al*., 2010;

$$\mathbf{Maximize}\ (\mathbf{Z})$$

$$\mathit{Subject\ to}:$$

$$\mathbf{S\,.\,v = 0}$$
$$\boldsymbol{lb_i} \le \mathbf{v}_i \le \boldsymbol{ub_i}\ ,\ \mathbf{1} \le \boldsymbol{i} \le \boldsymbol{n}$$

*Then;*

$$\boldsymbol{Maximize/Minimize}\ (\mathbf{v}_i)$$

*Subject to*:

$$\mathbf{S\,.\,v = 0}$$
$$\boldsymbol{w^T}\,.\,\mathbf{v} \ge \boldsymbol{\gamma.Z}\ ,\ \mathbf{0} \le \boldsymbol{\gamma} \le \mathbf{1}$$
$$\boldsymbol{lb_i} \le \mathbf{v}_i \le \boldsymbol{ub_i}\ ,\ \mathbf{1} \le \boldsymbol{i} \le \boldsymbol{n}$$

### 1.4.3. Genome-scale Metabolic Networks in Bioremediation

Biotechnology industries have beneficiated significantly from the development of metabolic networks and respective modeling (Saha *et al*., 2014). However, concerning the field of bioremediation, the development of genome-scale metabolic networks has only recently emerged (Oberhardt *et al*., 2009). As stated by Mahadevan *et al*., 2011, there exists a wide diversity of unexplored metabolic reactions encoded in the genomes of microorganisms that have an important environmental role. Similar approaches that have been used in the field of biotechnology could accelerate the elucidation of the physiology and ecology of these microorganisms and could guide optimization of the practical applications in the field of bioremediation.

Bioremediation takes advantage of a microbe's ability to reduce and potentially eliminate toxic effects of environmental pollutants. Additionally, microbes capable of degrading harmful waste produce useful chemicals as byproducts, and hence are intriguing production organisms as well (DESe Lorenzo, 2008). An updated list of the genome-scale metabolic networks that were used in order to correlate and predict bioremediation events are listed below;

- *Acinetobacter baylyi* is an innocuous soil bacterium that degrades pollutants (e.g. biphenyl, phenol, benzoate, crude oil, nitriles) and produces lipases, proteases, bioemulsifiers, cyanophycine, and biopolymers. *Acinetobacter* baylyi is easily transformed and manipulated by homology-directed recombination, enabling straightforward metabolic engineering. Therefore, the genome-based model is accompanied by an extensive library of mutants, and was validated against wild type growth phenotypes in 190 environments and gene essentiality data for nine environments (Durot *et al*., 2008).
- *Pseudomonas putida* KT2440 metabolic network that captures the important biotechnological capabilities, such as biodegradation of aromatic compounds (i.e. toluene, xylene), was constructed for this paradigmatic bacterium. Also, this study evaluated the metabolic network content and showed some examples of how *P. putida* could be used for biotechnological purposes (i.e. production of polyhydroxyalkanoates) (Nogales *et al*., 2008). More recently, an additional reconstruction of the previously described model was achieved in order to comprehend the degradation of polychlorinated bisphenols by *Pseudomonas putida* KT2440 in marine ecosystems (Taffi *et al*., 2014).

- *Geobacter metallireducens* reduces $Fe^{3+}$ and is used in bioremediation of uranium, plutonium, technetium, and vadium. Its ability to produce electrically conductive *pili* makes it useful for harvesting electricity from waste organic matter and as a biocatalyst in microbial fuel cell applications. Using *G. metallireducens* GEM, growth on different electron donors and electron acceptors was investigated. Model analysis revealed energy inefficient reactions in central metabolism, and experimental data suggested that the inefficient reactions were inactive during biomass optimization on acetate, but up-regulated when grown with complex electron donors. Additionally, the model was tested for flux predictions by comparison with $^{13}C$ labeling flux analysis. Simulations suggested that the tricarboxylic acid cycle was used to oxidize 91.6% of acetate, in agreement with 90.5% in $^{13}C$ labeling experiments (Sun *et al*., 2009).
- *Geobacter sulfurreducens* has similar industrial applications to *G. metallireducens* and is also able to reduce $Fe^{3+}$ (Mahadevan *et al*., 2006). OptKnock was applied to the *G. sulfurreducens* GEM to improve extracellular electron transport (Izallalen *et al*., 2008). Gene deletions in the fatty and amino acid pathways and in central metabolism were predicted to increase respiration and cellular ATP demand. To study the ATP demand increase, an ATP drain was added to the GEM. The model showed the rise in ATP usage correlated to decreased biomass flux and increased respiration rate. Experimental results confirmed that an ATP drain demonstrates the predicted results. Increasing electron transfer in *G. sulfurreducens* has advantages in both bioremediation and microbial fuel cell development, though increased fuel cell current was not found with this mutant strain (Mahadevan *et al*., 2006; Izallalen *et al*., 2008).
- *Rhodococcus erythropolis* is a remarkable bacteria used for bioremediation and fuel desulfurization. On a study developed by Aggarwal *et al*., 2011, it was reported the reconstruction of the first genome-scale metabolic model for *R. erythropolis* that could successfully predict cell growth results and explaining several experimental observations in the literature on biodesulfurization using dibenzothiophene. The *in silico* experiments and flux balance analyses allowed to propose minimal media, determine gene and reaction essentiality, and compare effectiveness of carbon, nitrogen, and sulfur sources (Aggarwal *et al*., 2011).
- *Rhodoferax ferrireducens* strain DMS 15236 is one of the few known facultative microorganisms that can grow anaerobically by oxidizing organic compounds to carbon dioxide with $Fe^{3+}$ serving as the electron acceptor. This attribute, as well as its ability to grow at the low temperatures found in many subsurface environments, suggests that it could contribute to the oxidation of organic matter coupled to the reduction of $Fe^{3+}$ in many soils sediments. The genome of *R. ferrireducens* harbors genes for benzoate degradation that are likely to be active under both aerobic and anaerobic conditions. The *R. ferrireducens* model contains a pathway for benzoate degradation, and predicts the growth of *R. ferrireducens* on benzoate with $Fe^{3+}$ as an electron acceptor (Risso *et al*., 2009).

A common feature to all those metabolic reconstructions is the comprehension of the metabolic processes that are need to be present in one microorganism in order to predict their bioremediation potential for the degradation of environmental pollutants. More recently, some studies also include the relationship between some of biodegradative microorganisms and other present in the soil environment (Zhuang *et al*., 2011).

## 1.5. Dissertation Overview

In order to comprehend the influence of different extracellular compounds on the *Polaromonas sp*. strain JS666 metabolism a representative *in silico* genome-scale metabolic model was reconstructed for this microorganism. This knowledge platform was built by integrating information provided from genome annotation and biochemical data from past studies. The produced model was further compared against physiological and phenotypic data using FBA and FVA with the purpose of validating the final model, hereby designated iJS666.

The genome-scale iJS666 model was also used with the combination of FBA with the final aim of predicting some compounds that may be added in cDCE contaminated soils towards an *in situ* biostimulation of *Polaromonas sp.* strain JS666 growth. In fact, both VC, TCE and cDCE are present down gradient from the Estarreja Chemical Complex site, an industrial complex located in North-West Portugal near Ria de Aveiro and classified as a priority remediation area under the Environmental Liabilities Recover Program (Branco, 2007). As such, ways in which to increase the metabolic rates and/or efficiency of these microorganisms is of national interest.

# 2. Methods

## 2.1. Hardware and Software Environment

The model reconstruction and the *in silico* analysis performed in this study were done on a Windows 7 Professional, 64 bit operating system with an AMD FX-4100 QuadCore Processor (3.60 GHz), 4GB of RAM.

All the initial raw data used on the draft reconstruction was extracted from PathwayTools software (V18.0 Tier 1) that simultaneously predicted the initial enzymatic reactions and GPRs.

MATLAB® (v8.0.0.783, R2012b, The MathWorks, Inc) was simultaneously used to convert the Microsoft Excel previously obtained file into a COBRA model and to execute all necessary scripts present in the COBRA Toolbox (v2.0.6), being *glpkmex* the linear solver used on the constraint-based algorithms. Since many other scripts and functions were developed during this dissertation, the complete MATLAB programming code is displayed in Supplementary Data 1 and sorted by script/function sequencial usage throughout this dissertation.

The iJS666 metabolic network visualization was achieved by converting the COBRA model into a *.sbml* file that was imported by CySBML(v1.30) into Cytoscape (v2.8.2). The CyFluxViz (v0.94) application was used for the visual representation of the reaction fluxes, simultaneously.

## 2.2. Genome-scale Metabolic Network Reconstruction

### 2.2.1. Automated Draft Reconstruction

Due to the scarce amount of biochemical evidence for the *Polaromonas* sp. strain JS666 enzymatic activity, the PathoLogic application available on the PathwayTools software was used initially to retrieved the genome annotation information from National Center for Biotechnology Information (NCBI) (NC_007948.1, NC_007950.1 and NC_007949.1) Genebank (*.gb*) and Fasta files (*.fa*) for the chromosome, plasmid pPOL338 and pPOL360, respectively and to generate the initial draft reconstruction as exemplified in Karp *et al*., 2009, and explained in Karp *et al*., 2011. The reactions predicted by PathwayTool were previously stored in MetaCyc database and were, *a priori,* mass and charge balanced to a physiological pH of 7.3 (Caspi *et al*., 2006). The predicted information for the reactions, metabolites and gene-protein-reactions were then download and stored in Microsoft Excel file which allowed faster information processing and was suitable to make the needed modifications on the manual refinement stage. Hence, the initial metabolic network was further improved by following the protocol for generating high quality genome-scale metabolic networks step-by-step (Thiele & Palsson, 2010).

### 2.2.2. Manual Refinement

The initial automated reconstruction file yielded a representation of an metabolic network that functioned as a starting point to the final reconstruction although extensive manual curation was need in order to obtain a functional model. The initial reactions list included many non-physiological reactions that should not be included in the final model since those reactions are not mass balanced or have any relevant information to the metabolic process. Some of the reactions excluded in this process are involved in post-translational modifications, non-metabolic protein phosphorylation activity, DNA/RNA modification and degradation, etc…, as exampled below (Thiele & Palsson, 2010);

**Example 1:** Non-physiological reaction excluded from model.

Protein-L-histidine + ATP → Protein-N-phospho-L-histidine + ADP + $H^+$

Some enzymatic reactions displayed in the initial draft were presented in a generic form, in which one or more of the used metabolites were chemically unspecific. Most of these reactions were immediately excluded since it was impossible to identify the specific metabolites used by them (those reactions were easily identified by searching for metabolites with no molecular mass associated or with generic chemical terms). Some generic reactions that could be easily associated (same substrates, products and gene associations) with well-defined reactions already included in the model, were also excluded in order to avoid repetitions (Thiele & Palsson, 2010);

**Example 2:** Reaction excluded due to unspecific metabolites.

Carboxylic ester + $H_2O$ → Alcohol + Carboxylate + $H^+$

Some of those enzymatic reactions present in the initial draft had unspecific cofactors. When the data revealed the generic group of the cofactor, the reaction unspecific cofactor was substituted by the most common representative cofactor for that respective group. When no information was given for the used cofactor then the reaction was modified to include NAD+/NADP+ (depending if they were present on a catabolic or anabolic pathway, respectively). This assumption lead to an insignificant flux variation since the used cofactors had interconvertible redox reactions included in the model;

**Example 3:** Reaction with unspecific cofactor.

Octanol + Oxidized electron acceptor → Octanal + Reduced electron acceptor

Many similar enzymatic reactions, present in the initial draft, had in their stoichiometric constitution isomeric convertible metabolites. In order to simplify the iJS666 model, only one isomer was selected for those reactions. This simplification was pertinent due to the huge amount of metabolites present with an isomeric form that lead to metabolic network dead-ends. This simplification did not affect the reaction fluxes, although in the real cellular metabolism if more than one isomer is produced on the biological system, those compounds could have different downstream pathways or one of those metabolites could accumulate in the cellular pool. The only way to amend this problem in a constraint-based methodology would be to add demand reactions to simulate the consumption of such metabolites although in many cases there is no information about the inclusion rate of those metabolites in the cellular pool. Also, since the metabolic network model had the purpose to be analyzed by constraint-based methods, some polymer associated reactions, in which polymer product had a stoichiometry algebraically representation, could not be inputted with the initial format cause they had to be mass balanced due to the pseudo-stationary constraint. In order to solve this problem, those reactions were simplified to the production/consumption of their respective monomers and the polymer produced was set to a sink/depletion reaction.

**Example 4:** Polymeric reactions with algebraically stoichiometry.

10-formyl-THP-[**γ**-Glu]$^{n}$ + L-glutamate + ATP → 10-formyl-THP-[**γ**-Glu]$^{n+1}$ + ADP + phosphate

In the manual curation stage, linking macromolecular reactions to the predicted metabolic reactions should be taken in consideration by the researcher in order to achieve a functional model that have the expected physiognomic characteristics, as explained in the next sections.

### 2.2.2.1. DNA, RNA and Protein Coefficients

DNA, RNA and protein are some of the most fundamental biomass constituents present in a cell. Therefore, the relative contribution to the biomass composition and monomer composition for each of those biopolymers are crucial for high-quality metabolic modelling (Thiele & Palsson, 2010).

Unfortunately, it was impossible, within the framework of this dissertation, to quantify experimentally the relative fractional composition values for all of those molecules in the *Polaromonas* sp. strain JS666 biomass and some assumptions were made based on previously published literature. More specifically, the biomass relative composition regarding those biopolymers was assessed base on the biomass composition of *R. ferrireducens*, the closest phylogenetically related microorganism with a correlated genome-scale metabolic model (Risso *et al*., 2009).

Nevertheless, the monomer composition of those polymers was estimated from genome data. The DNA composition was determined by the stoichiometric quantification of each nucleotide in all genetic elements, assuming the double strand configuration of the molecule, as described in Thiele & Palsson, 2010. Henceforward, the nucleotide relative composition of the RNA was recovered by the quantification of the different monomers present on all the coding sequences (CDS) assuming a single strand configuration for this molecule (Thiele & Palsson, 2010). Using the codon usage information displayed in Kazusa database (Nakamura, 2007) for the *Polaromonas* sp. strain JS666, the amino-acid composition of the proteins was estimated based on the coding sequence, as reported in Table 2.1.

**Table 2.1** Major macromolecules of biomass composition in iJS666 model. DNA, RNA and protein coefficients (left) are defined by the amount of constituent monomers. For example, 18.763 mmol gDW$^{-1}$ h$^{-1}$ of dATP are required to produce 1 mmol gDW$^{-1}$ h$^{-1}$ of a DNA molecule with 100 total deoxyribonucleotides. In the case of phospholipids and acyl-ACP all values are in mmol gDW$^{-1}$ h$^{-1}$ and the components of biomass are expressed in mmol gDW$^{-1}$.

| dNTPs | DNAJS666 | Phospholipids Heads | PhospholipidJS666 |
|---|---|---|---|
| dATP | 18.763 | cardiolipin | 0.057 |
| dCTP | 31.236 | phosphatidyl-ethanolamine | 0.688 |
| dGTP | 31.236 | phosphatidyl-glycerol | 0.193 |
| dTTP | 18.763 | phosphatidyl-serine | 0.062 |
| TOTAL | 100 | TOTAL | 1 |
| **NTPs** | **RNAJS666** | **Acyl-[ACP] Pool Component** | **Acyl-[ACP]** |
| ATP | 18.904 | palmitoleoyl-[acp] | 1.34 |
| CTP | 31.238 | cis-vaccenoyl-[acp] | 0.158 |
| GTP | 31.446 | palmitoyl-[acp] | 0.392 |
| UTP | 18.412 | 3-oxo-decanoyl-[acp] | 0.05 |
| TOTAL | 100 | TOTAL | 1.94 |
| **AA** | **ProteinJS666** | **Biomass Component** | **Coefficient** |
| GLY | 25.85 | DNAJS666 | 0.00104 |
| GLU | 16.56 | RNAJS666 | 0.006566 |
| ASP | 15.82 | ProteinJS666 | 0.001610 |
| VAL | 24.22 | PhospholipidJS666 | 0.00256 |
| ALA | 38.93 | 4-methylphenol | 0.00034 |
| ARG | 20.56 | coenzyme A | 0.000345 |
| SER | 18.03 | glycogen | 0.1598 |
| LYS | 11.99 | heme-O | 0.000034 |
| ASN | 8.67 | (2r,4s)-2-methyl-2,3,3,4-tetrahydroxytetrahydrofuran | 0.00102 |
| MET | 8.05 | lipid A-core | 0.00908 |
| ILE | 14.46 | peptidoglycan | 0.02614 |
| THR | 16.65 | putrescine | 0.03527 |
| TRP | 4.5 | pyridoxal 5’-phosphate | 0.000034 |
| CYS | 3.08 | spermidine | 0.00713 |
| TYR | 7.18 | tetrahydrofolate | 0.05 |
| LEU | 33.81 | thiamine diphosphate | 0.00034 |
| PHE | 11.38 | UDP-α-glucose | 0.003 |
| GLN | 12.66 | FAD | 0.00069 |
| HIS | 7.13 | NADH | 0.0022 |
| PRO | 16.47 | NADPH | 0.0017 |
| SEL | 1 | ubiquinol-8 | 0.00034 |
| TOTAL | 317 | GAM | 46.7 |

### 2.2.2.2. Membrane Phospholipid Composition

On previously reported genome-scale metabolic networks, the membrane phospholipid fraction could vary from 9.1 % (w/w) (Thiele & Palsson, 2010) to 15 % (w/w) (Mahadevan *et al*., 2006) of the biomass content and have an enormous relevance on the predictive capabilities of the metabolic model due to the variable composition, high molecular weight and energetic cost for its biosynthesis (Thiele & Palsson, 2010).

Similar to the DNA, RNA and protein relative composition, the membrane phospholipid relative composition was set to be the same as in *Geobacter sulfurreducens* and *Rhodoferax ferrireducens* (Mahadevan *et al*., 2008; Risso *et al*., 2009). Nevertheless, the phospholipid composition was determined by experimental data retrieved from other *Polaromonas* species (Margesin *et al*., 2012). Some of the discovered phospholipids acyl-groups present in the *Polaromonas* species were not included in the previously described microorganism's models. Therefore, the relative acyl phospholipid composition was set differently from the *R. ferrireducens* and *G. sulfurreducens*. In order to culminate this problem, the relative phospholipid composition of the *Polaromonas* sp. strain JS666 was set as described in the Table 2.1.

### 2.2.2.3. GAM and NGAM Balance

In biological systems, ATP hydrolysis is required in order to growth and to maintain some of the biological processes active. GAM (growth associated maintenance) is the consumption rate of ATP hydrolysis necessary to the growth of the microorganism. This rate represent mainly an output of energy cost necessary to DNA, RNA and protein polymerization reactions to occur. NGAM (non-growth associated maintenance) is the rate of ATP hydrolysis related to the maintenance of the cell when there is no growth occurring. For example, some reactions that contribute to the value of this rate are related with cell osmoregulation, repair mechanisms and cascade signaling. Both GAM and NGAM have a huge influence on the model predictions, especially on the predicted growth rate (Thiele & Palsson, 2010).

Usually GAM values used in genome-scale models are very similar even between different organisms although the same is not true to NGAM values. For example, *Escherichia Coli* model iAF1260 have a NGAM value of 8.31 mmol gDW$^{-1}$ h$^{-1}$ (Feist *et al*., 2007) while the slow growing bacteria *R. ferrireducens* and *G. sulfurreducens* models have a NGAM value of only 0.45 mmol gDW$^{-1}$ h$^{-1}$ attributed. When not mention otherwise, the NGAM was set to an invariable flux value of 0.45 mmol gDW$^{-1}$ h$^{-1}$ as reported previously by Mahadevan *et al*., 2006, and Risso *et al*., 2009. Later on in this dissertation, the influence of this flux value in the predicted growth rate was accessed in Chapter 3.2.2. The GAM rate represents an energy cost that is proportional to the growth rate and, therefore, was included directly in the biomass equation. A flux value of 46.7 mmol gDW$^{-1}$ was established for iJS666 model based on the previously reported value for the *G. sulfurreducens* (Esteve-Núñez *et al*., 2005; Mahadevan *et al*., 2006) and in *G. metallireducens* (Jun *et al*., 2009).

### 2.2.2.4. Final Biomass Equation

The biomass reaction accounts for all known biomass constituents, their fractional contributions to the overall cellular biomass (Thiele & Palsson, 2010). The metabolites in the biomass reaction may affect the *in silico* essentiality of reactions and their associated genes and when the model tries to predict the optimal growth rate accurately the fractional distribution of each compound plays an important role (Thiele & Palsson, 2010). Since the fractional quantities of the biomass precursors were not experimentally measured for *Polaromonas* sp. strain JS666, the remaining metabolites present on the biomass equation of the iJS666 model were set as reported in *R. ferrireducens* biomass equation (Rossi *et al*., 2006). The unit of the biomass reaction is $h^{-1}$ since all biomass precursor fractions are defined in mmol $gDW^{-1}$ and uptake/internal fluxes were introduced in mmol $gDW^{-1}$ $h^{-1}$. Therefore, the biomass reaction sums the mole fraction of each precursor necessary to produce 1 g dry weight of cells, as reported in Table 2.1.

### 2.2.2.5. Model Compartmentalization and Transport Reactions

*Polaromonas* are bacteria from the family of *Comamonadaceae* which belongs to the class of Betaproteobacteria and therefore are Gram-negative (Osborne *et al*., 2010). Hence, the iJS666 model was initially compartmentalized in cytoplasm, periplasm and extracellular spaces.

The PathwayTools software was able to predict the existence of some inner and outer membrane transport systems operating on *Polaromonas sp.* strain JS666 from the genome annotation. Some of those predicted reactions were in the generic form and consequently they had to be simplified or excluded in cases when the information provided was insufficient to identify the transported metabolites. There was a huge lack of information regarding the cellular localization of those enzymes and many of the periplasmic inner transport reactions would form gaps since there was no corresponding uptake from the extracellular medium.

In the early stages of the model reconstruction, an attempt was made in order to identify and debug those transport systems. PSORT 3.0 was used for the identification of membrane proteins and their intracellular localization. Subsequently, a BLASTp was performed using the CDS of those membrane proteins (data not showed) and the majority of the found results were unclear since, once again, the reactions associated to those homologous enzymes were unspecific. Also, PSORT was unable to predict the membrane localization (inner or outer) of many transmembrane proteins and consequently a simplification of the model was became absolutely necessary. All predicted transport reactions were modified, limiting the intracellular localization of a metabolite to the cytoplasm - [c] - or to the extracellular medium - [e]. The directionality of transport reactions (reversible/irreversible) was kept along with the transport mechanism (symport or antiport) as exemplified below;

**Example 5:** Draft transport reaction and the respective final transport reaction.

$$\text{ammonium[periplasmic]} + H^{+}\text{[periplasmic]} \rightarrow \text{ammonium[cytosol]} + H^{+}\text{[cytosol]}$$

$$\text{ammonium[extracellular]} + H^{+}\text{[extracellular]} \rightarrow \text{ammonium[cytosol]} + H^{+}\text{[cytosol]}$$

Although this simplification could have a negative influence on the predictive potential, since metabolites could accumulate in different compartments within the microorganism, but it simultaneously reduced the possibility of Type III Pathway formation (Thiele & Palsson, 2010). A similar simplification was also previously reported on the *R. ferrireducens* model without compromising the model predictive capabilities even that those relied mainly on the transmembrane systems (Risso *et al*., 2009).

Finally, there was experimental evidence that some compounds could be degraded and used as carbon-sources for biomass growth (Alexander, 2010) as reported in Table 3.2. An extracellular active (with ATP hydrolysis) transport reaction was added for charged metabolites as suggested in Thiele & Palsson, 2010. In order to facilitate the mass balance of extracellular compounds and easier medium control, the uptake from extracellular medium was defined by exchange reactions in the format exampled below for all the extracellular metabolites;

**Example 6:** Exchange reaction between extracellular medium and the closed system.

$$\text{ammonium[extracellular]} \leftrightarrow \text{( Closed System)}$$

## 2.2.3. Metabolic network debugging

### 2.2.3.1. Confidence score

The lack of biochemical evidence for the enzymatic reactions present in the obtained *Polaromonas sp.* strain JS666 initial draft resulted in insufficient information needed to proceed with the network debugging described in the previously referred protocol (Thiele & Palsson, 2010). In order to overcome the lack of a confidence test for the iJS666 model reactions, an automatic confidence scoring system was developed assuming that the reactions present simultaneously in iJS666 draft model and in the genome-scale metabolic network of the phylogenetically related *R. ferrireducens* microorganism had a higher probability of occurring. Also, the existence of a predicted GPR (gene-protein-reaction association) or biochemical evidence was positively accounted by the scoring system (Thiele & Palsson, 2010) as described in Table 2.2.

**Table 2.2.** Automatic confidence scoring system used in iJS666 model.

| Evidence type | Confidence score | Examples |
|---|---|---|
| Biochemical Data | 4 | Direct evidence for gene product function and biochemical reaction. |
| Strong Annotation Evidence | 3 | Gene predicted by PathwayTools and reaction present on *R. ferrireducens* |
| Weak Annotation Evidence | 2 | Gene/Reaction only predicted in PathwayTools |
| Weak Evidence | 2 | No evidence is available but reaction is required for modeling and similar reaction is present at *R. ferrireducens* |
| Modelling Data | 1 | No evidence is available but reaction is required for modeling. |
| Not evaluated | 0 | Not applied to this model since all reactions were automatically scored. Serves as a scoring error identifier. |

### 2.2.3.2. Pathway Gap Filling

In every genome-scale metabolic network reconstruction the first step on the debugging mode is the identification of the metabolic gaps present in the draft network. A metabolic/pathway gap can be seen as a hole in the network that don't allow the *in silico* model achieve a feasible result on the objective function/equation (e.g.: feasible growth under some specific growth condition previously determined given a set of nutrients and secretion products), making therefore the biological system unfeasible. If one had a complete knowledge about the functionality of the genome, no gap-filling would be needed. However, as reported before, a complete genome annotation is near impossible to obtain. With the increased rate of genome-scale metabolic networks, new parsimony-based algorithms to automatically fill the gaps of the network were developed although these software's only search for possible reactions to be added to network under the available databases and it is the curator's ultimate responsibility to decide which criteria to apply to close the metabolic gap (Latendresse, 2014).

In order to find the metabolic gaps in the iJS666 draft reconstruction, the COBRA toolbox script *GapFind.m* was used, as described in Thiele & Palsson, 2010. This script allowed to identify topological/root gaps and non-topological although a minor modification was made in order to also identify metabolites exclusively inserted in one reversible reaction since they would also form dead-ends. Depending on the type of formed gap and on the unbalanced metabolite, different procedures to correct those gaps were taken into consideration, as described in Thiele & Palsson, 2010. Since many reactions had to be manually introduced to the metabolic network, the identifier ('NEWRXN') was used in the Excel file in order to distinguish between reactions predicted by the automatic draft reconstruction software, PathwayTools, and those manually introduced. The confidence scoring system, previously described, was also applied to this reactions.

### 2.2.3.4 Stoichiometric inconsistencies and cyclic infeasibilities

Assuming a set of metabolic reactions correctly mass-balanced, the metabolic reactions with the same metabolites would always have the same (or proportional) amount of stoichiometric coefficients for each metabolite, independently of the reaction directionality. Since initially the reactions were admitted in the metabolic network as they were uploaded from the database MetaCyc, a method analogous to the one presented by Gevorgyan *et al*., 2008, was used in order to identify possible inconsistencies in the stoichiometry of similar reactions. All the reactions which had identical metabolite constitution or similar, diverging at most by one metabolite, were found by the script *similarity.m* including, but not limited to, the presence of different metabolite stoichiometry, the variation in one proton molecule or the use of different cofactor (correct as described in the Manual Refinement section of Thiele & Palsson, 2010, protocol), then the comparison of those reactions and curation of eventual stoichiometric inconsistencies was done manually.

A common problem when construction metabolic network models is the formation of undesired cycles/loops within the system. Accordingly to Schilling *et al*., 2000, metabolic pathways could be categorized based on mathematical principles such as linear algebra and convex analysis into three different types of extreme pathways; Type I pathways, requiring the use at least of one of the primary exchange fluxes to be active, Type II pathways (or futile pathways), that only need the currency exchange fluxes (energy and reductive power) to be active and finally Type III pathways (internal loops) that do not require any of the exchange fluxes to be active (Schilling *et al*., 2000). Although the latter pathways do not influence the final flux rate of the objective function, it can mislead the understanding of the network and respective dependencies by predicting existent intracellular fluxes that are unrealistic accordingly to the real thermodynamic constraints.

In order to avoid those intracellular loops from the model while using COBRA toolbox, an alternatively FBA methodology was tried: Loop Less FBA (LL-FBA) as described in Schellenberger *et al*., 2011. This method turns any linear programming problem into a modified mixed-integer problem solving the initial problem with an additional constraint: exclusion of network fluxes that contain loops by removing solutions which the sum of the intracellular Gibbs energy is null (Schellenberg *et al*., 2011). Although, solving the system with this methodology was possible (same objective function flux rates but with different and lesser intracellular fluxes), it has long-processing time especially in a large genome metabolic network. As alternative the *objectiveCbModel.m* script presented from the COBRA toolbox was used to solve all linear problems, from this point on, using norm one minimization of internal fluxes. These different approaches had very identical results in eliminating intracellular loops (data not shown) however the second methodology had faster processing time.

### 2.2.3.5. Initial Growth Testing

A viable carbon and energy source is necessary for any organism in order to accumulate biomass and reproduce. Therefore, the absence of a suitable carbon source should result in a zero flux value on the biomass reaction/equation since biomass precursors are organic molecules constituted by carbon atoms. Before any FBA on the reconstructed model, one should test for grow without carbon source (Thiele & Palsson, 2010). In order to achieve that, a script called *blank.m*, representing a medium without a carbon source, was used to verify the previously described condition. Initially the reconstructed model was able to grow (have positive flux value to biomass equation) without any carbon source because there were intracellular reactions that allowed the acquisition of carbon from $CO_2$ and, simultaneously, there were reactions present that had incorrect directionality and allowed the transport of protons to extracellular medium, allowing costless ATP formation. Also, exchange reactions for $H_2O$ and $H^+$ were set to zero, meaning that there was no addition/removal of water molecules from the medium and that pH of the cell/extracellular medium remained constant. Also, the exchange reactions for $CO_2$, $H_2$ were set in order to only release those compounds to the medium, never incorporating them in the defined system. After all those manual curations, the iJS666 model was not able to growth in the medium without carbon source as supposed.

As described in Thiele & Palsson, 2010, in the Network Evaluation section, preliminary tests were done to access if the model was able to predict growth under a known viable medium composition. In order to achieve that, an initial test condition was defined based in Minimal Salt Medium (MSM) mentioned in the bibliography (Hartmans *et al.*, 1985) and using the script *glucose.m*, which was similar to *blank.m* but included glucose as carbon-source, as shown in the Table 2.3. Also, every intracellular reaction bound was set to a value of -1000 and 0 mmol gDW$^{-1}$ h$^{-1}$ to the lower bound for reversible reactions and irreversible reactions, respectively, and a value of 1000 mmol gDW$^{-1}$ h$^{-1}$ to the upper bounds for all those reactiosn.

**Table 2.3.** MSM proposed by Hartmans *et al.*, 1985 (left). Exchanged metabolites and exchange bounds (right).

| MSM [weight/L] | Exchanged Metabolite | Lower Bound [mmol gDW$^{-1}$ h$^{-1}$] | Upper Bound [mmol gDW$^{-1}$ h$^{-1}$] |
|---|---|---|---|
| 3.88 g $K_2HPO_4$ | $Co^{2+}$ | -100 | 0 |
| 2.13 g $NaH_2PO_4$▪$2H_2O$ | $Fe^{3+}$ | -100 | 0 |
| 2.0 g $(NH_4)2SO_4$ | $Mg^{2+}$ | -100 | 0 |
| 0.1 g $MoCl_2$▪$6H_20$ | $Mo^{2+}$ | -100 | 0 |
| 10 mg EDTA | $Ni^{2+}$ | -100 | 0 |
| 2 mg $ZnSO_4$▪$7H_2O$ | $Zn^{2+}$ | -100 | 0 |
| 1 mg $CaCl_2$▪$2H_2O$ | $Na^+$ | -100 | 0 |
| 5 mg $FeSO_4$▪$7H_2O$ | $K^+$ | -100 | 0 |
| 0.2 mg $Na_2MoO_4$▪$2H_2O$ | phosphate | -100 | 0 |
| 0.2 mg $CuSO_4$▪$5H_2O$ | ammonium | -100 | 0 |
| 0.4 mg $CoCl_2$▪$6H_2O$ | chloride | -100 | 0 |
| 1 mg $MnCl_2$▪$2H_2O$ | selenite | -100 | 0 |
|  | sulfate | -100 | 0 |
|  | oxygen | -100 | 0 |
|  | D-glucose | -1 | 0 |
|  | Secreted Metabolites | 0 | 100 |

In the initial growth test, the model predicted an excessive specific growth rate indicating that some reactions within the model were allowing a costless incorporation of carbon in the metabolism. With the intention of detecting those prone-to-error reactions the incorporation of the carbon-source was set to very low value (0.001 mmol gDW$^{-1}$ h$^{-1}$), the bounds for all intracellular reactions were set to an inflated value (10000 mmol gDW$^{-1}$ h$^{-1}$) and a FVA was conducted in order to identify the reactions that were necessary to maintain that excessive growth rate (very high flux rate with low variation). Those reactions allowed free incorporation of carbon due to previously undetected errors on equation mass-balance or in reaction directionality. After this final curation, the model was able to demonstrate a reasonable growth rate for the tested carbon-source and simultaneously demonstrated that without any carbon-source added to the medium, the growth rate value was, as expected, zero.

## 2.3. Model Validation

Model validation is a necessary step on the reconstruction of a genome-scale metabolic network (Thiele & Palsson, 2010) since the predictive capabilities of the iJS666 model should reproduce flux values similar to the observable production and consumption rates in *Polaromonas* sp. strain JS666 and also the model should be able to reveal which GPRs contribute to those fluxes. For that reason the model validation was accessed by the correlation the real degradative capabilities of the microorganism in comparison to those predicted by the iJS666 model.

As stated before, *Polaromonas* sp. strain JS666 is a poorly studied microorganism and therefore few bibliographic data is available from which experimental chemostat data could be retrieved. In 2010, Alexander did an extensive work analyzing the viable mediums/carbon sources that support *Polaromonas* sp.JS666 growth. Some growth mediums only allowed *Polaromonas* sp. strain JS66 to grow in Petri plates, at very low concentrations, while others indorsed relatively faster growth in liquid cultures and therefore cellular concentration was measured alongside.

The iJS666 model was tested for all compounds referenced accordingly the following criteria; the model should not predict grow for carbon-sources that revealed negative auxanography plates; the model should predict a positive biomass flux value for compounds that allowed *Polaromonas* sp. strain JS666 to grow in auxanography plates; and for those compound which allowed growth in liquid cultures the growth rates were directly compared with the experimental extrapolated values. That was achieved by the scripts *validationnegative.m*, *validationpositive.m*, respectively. In the liquid growth cultures, the extrapolated specific growth rate were obtain using the following formulation;

$$ODfinal = ODinitial\,.\,2^{MD}$$

$$\mu\ (h^{-1}) = \frac{\ln(ODfinal) - \ln(ODinitial)}{t_{MD}}$$

Assuming that all substrate present on the medium was consumed during the growth ($S$ = 0), $X$ defining the biomass concentration, the specific substrate consumption rate (Rs) was acquired by the following expression;

$$-\frac{dS}{dt} = R_s \,.\, X \quad , \quad \frac{dX}{dt} = \ \mu \,.\, X \quad and \quad X\,(gDW\ L^{-1}) = X_0\,.\,e^{\mu.t_{MD}}$$

$$R_s(mmol\ gDW^{-1}h^{-1}) = \frac{-S_o\,.\,\mu}{X_0 - \ X_0\,.\,e^{\mu.t_{MD}}}$$

The specific growth rate specific substrate consumption rates were directly introduced in the model iJ666 as lower bounds for the respective substrate exchange reactions. Each metabolite used as substrate had a different script that defined the extracellular medium composition as exampled by the *cDCE.m* script that defines the system boundaries for the iJ666 model using cDCE. In all the predictions made using iJS666 the objective function was defined as the maximization of the growth and therefore the predicted values of the specific growth rate were directly compared to the value extrapolated from the collected data. Since the majority of the predictions obtained using the iJS666 model for the liquid cultures were not initially correlated to experimental data, further assessments were made in other to identify the influence of the specific substrate consumption rate using the script *validationvariation.m*.

Also, since some oxygen uptake rates of resting *Polaromonas* JS666 cells were acquired from Nishino *et al*., 2013, the exchange oxygen rates from the model iJS666 under different substrates and substrate consumption rates were also compared for validation using the script *validationox.m.*

After those initial validation tests, further predictions were made using the established iJS666 model in order to access the influence of model parameters, to test the effect environmental conditions on the biomass growth and to predict viable compounds to be used as biostimulants.

## 2.5. Analysis of cDCE degradation in iJS666 model

In order to analyze the intracellular fluxes of *Polaromonas* sp. strain JS666 the internal fluxes of the reactions downstream to the cDCE degradation were predicted by maximization of the biomass production using a specific cDCE consumption rate of 0.561 mmol gDW$^{-1}$ h$^{-1}$ in a FBA. Also, further analysis to the bottleneck reactions of the iJS666 model, using the previously described parameters, were assessed using the script *FVAcDCE.m*, as reported in Supplentary Data 2.

Similarly, the influence of the intracellular glutathione was evaluated by performing an FVA with different objective functions (see Supplementary Data 3) using the model iJS666 with and without the presence of the glutathione molecule in the biomass equation by the script *gluta.m* and *FVAcDCE.m*

Furthermore, with the intend of analyzing the influence of the parameters NGAM and GAM in the specific growth rate as well as to comprehend the influence of different transport systems in the iJS666 model, a FBA was done by varying those parameters using the script *TNGAM.m.*

Finally, the influence of nitrogen and sulfur sources were also taking in consideration in the predictive biodegradation capabilities of the iJS666, using the script *nitrogen.m* and *sulfur.m,* respectively, since those nutrients may have a significant role in the growth of *Polaromonas* sp. strain JS666 as reported in Mattes *et al*., 2008.

## 2.6. Biostimulants prediction using iJS666 model

The final objective of this dissertation was to predict some compounds to be used as biostimulants in the cDCE degradation by *Polaromonas* sp. strain JS666. In order to do so, all metabolites defined in this reconstruction were tested in the model iJS666 in which cDCE was the main carbon source by the script *testsub.m*. A threshold of 30% for bioaugmention, measure by the increase of the specific growth rate, was set to avoid the false discovery of potential biostimulants since many intracellular metabolites were manually defined. All the tested metabolites were introduced in the cytosol by setting a putative active transport system in order to underestimate the bioaugmentation capabilities. Also, a parallel FBA was done in order to verify if the added compound would serve as solo carbon source and allow the growth of *Polaromonas* sp. strain JS666 without cDCE degradation. The objective of this restraining condition was to avoid a possible catabolic repression that was double checked by the verification whether the cDCE consumption rate did not decrease comparing to the medium without the added compound. After that a manual selection of the positively identified putative biostimulants was done in order to exclude the presence of artefacts or complex metabolites that could not be used in field as biostimulants due to practical reasons.

# 3. Results and Discussion

## 3.1. The iJS666 Model

### 3.1.1. Main Characteristics of iJS666 GENRE

The systematic reconstruction of the metabolic capabilities of *Polaromonas sp. strain JS666*, which illustrates the extent of our current metabolic knowledge of this microorganism, was included in a spreadsheet representation of the model. The displayed reactions were manually categorized in different basic pathways for easier user comprehension/localization. The final iJS666 model was successfully implemented in the COBRA toolbox for Matlab software, yielding 1395 reactions and 1068 metabolites (6054 single interactions), as exampled by the stoichiometric matrix in the Figure 3.1.

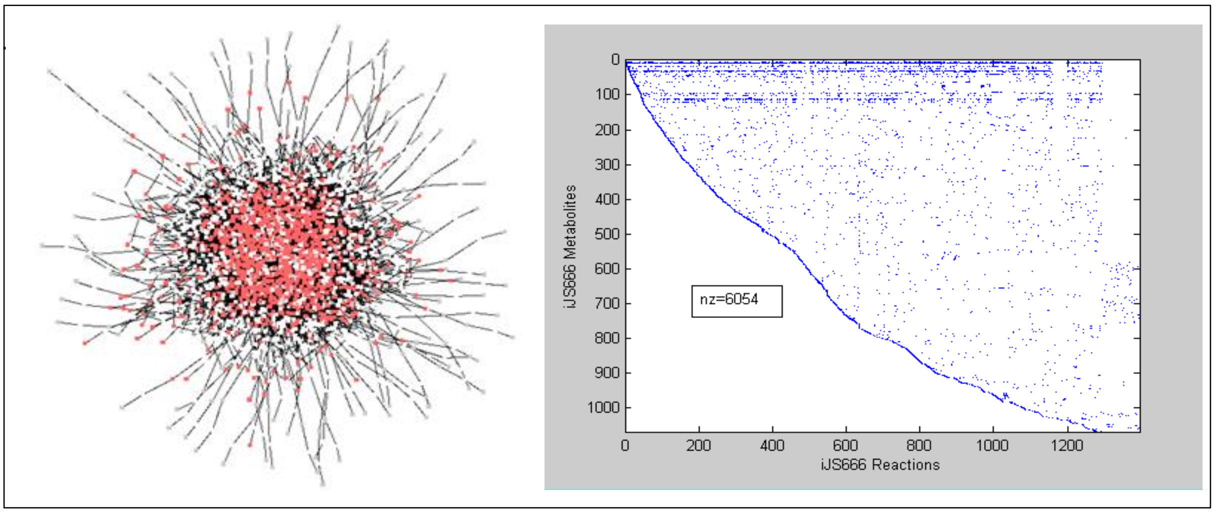


**Figure 3.1.** Representation of iJS666 network in Cytoscape (left) and the respective stoichiometric S matrix (right).

In comparison to most other genome-scale metabolic networks, the iJS666 model has a relative high number of associated reactions due to the fact that major biosynthetic pathways with linear dependent reactions were not simplified to shorten the number of reactions present in the system. That decision relied on the fact that GPR associations could be misleading and function dependency would not be totally understood in posterior analysis. Also, the total number of reactions include a great number of sink and transport reactions that were added in order to remove dead-ends or to uptake metabolites, see Table 3.1. As can be seen in the complete reactions breakdown in Table 3.1, the total number of iJS666 model's reactions also account for the automatically added exchange reactions for metabolites present in the extracellular medium. Therefore, the number of intracellular and transport reactions is similar to the number presented in different metabolic models of close related microorganisms (Risso *et al*., 2009; Mahadevan *et al*., 2006).

**Table 3.1.** Metabolic network proprieties of iJS666.

| | |
|---|---|
| Genes | 607 |
| Intracellular Reactions | |
| ∴ Draft | 1084 |
| ∴ Gap Filling | 174 |
| Metabolites | |
| ∴ Cytosol | 969 |
| ∴ Extracellular | 99 |
| Exchange Reactions | 99 |
| Transport Reactions | 118 |
| Sink Reactions | 39 |

As for the type of the reactions included in the iJS666 model, approximately 81.5% were irreversible reactions and 18.5% reversible reactions, respectively. These values are also in conformity with the previously referred metabolic reconstructions. The distribution of intracellular reactions among the different defined pathways is represented in the Figure 3.2. The significant amount of reactions present in the amino acid, cofactor, carbohydrates, lipid and nucleotide metabolism is due to the general need for those reactions in the synthesis of biomass components of the cell and therefore those reactions are more prevalent in every metabolic reconstruction.

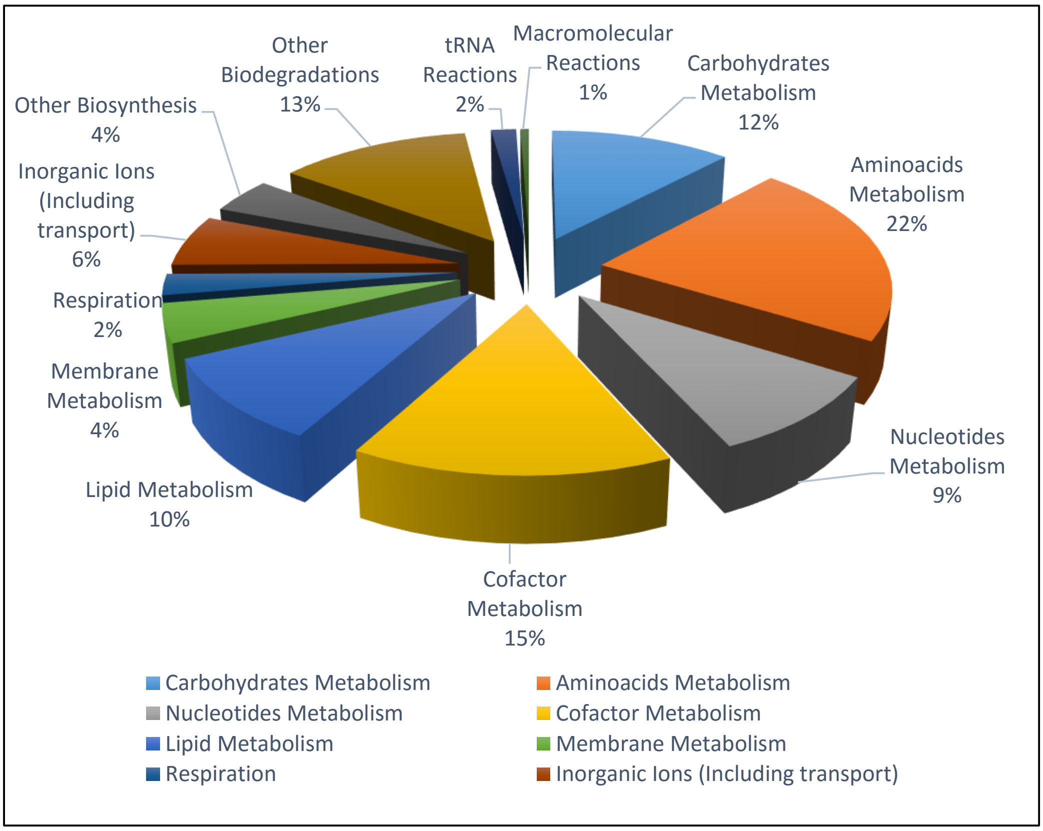


**Figure 3.2.** Affiliation of intracellular reactions to the different major metabolic pathways.

Due to insufficient knowledge of the full metabolic pathways that reside in *Polaromonas* sp. strain JS666, some metabolic root dead-ends were corrected by adding sink reactions for their depletion as referred before. Since many of those compounds are essential factors in cell development and growth, further studies should be done in order to access their relative proportion in the biomass composition. The disclosure of the complete list of those root dead-end metabolites is given in Table 3.2.

**Table 3.2.** Root dead-end metabolites added as sink reactions.

| **Root Dead-end Metabolite** | |
|---|---|
| 5'-deoxyadenosine | L-ascorbate |
| glycolaldehyde | guanylyl molybdenum cofactor |
| cadaverine | siroheme |
| benzene | myo-inositol |
| γ-linolenate | α,α-trehalose |
| 5-hydroxy-7a-methyl-1-oxooctahydro-1-h-indene-4-carboxylate | β-D-mannose-6-phosphate |
| dialurate | dTDP-β-l-rhamnose |
| protoanemonin | GDP-4-dehydro-α-D-rhamnose |
| urate | GDP-mannuronate |
| $Mg^{2+}$ | adenosine-3',5'-bisphosphate |
| $Ni^{2+}$ | cis-vaccenate |
| $Zn^{2+}$ | laurate |
| $K^{+}$ | palmitate |
| $Hg^{0}$ | palmitoleate |
| chromate | stearate |
| S-adenosyl-4-methyl,thio-2-oxobutanoate | cyanophycin |
| biotin(II) | polyphosphate |
| coenzyme B12 | polysulfide |
| cGMP | tetrahydrofolate glutamate |

### 3.1.2. Confidence Score and Robustness Analysis

Many of the reactions added to the reconstruction had essential enzymes for a specific functional pathway but no attributed gene association. In order to catalogue those reactions for further improvement and for establishing some degree of confidence about the existence of the pathways used by the iJS666 model, a relative confidence score algorithm was establish as previously described in Table 2.2. An overview of the iJS666 reaction's confidence score is displayed in the Figure 3.3.

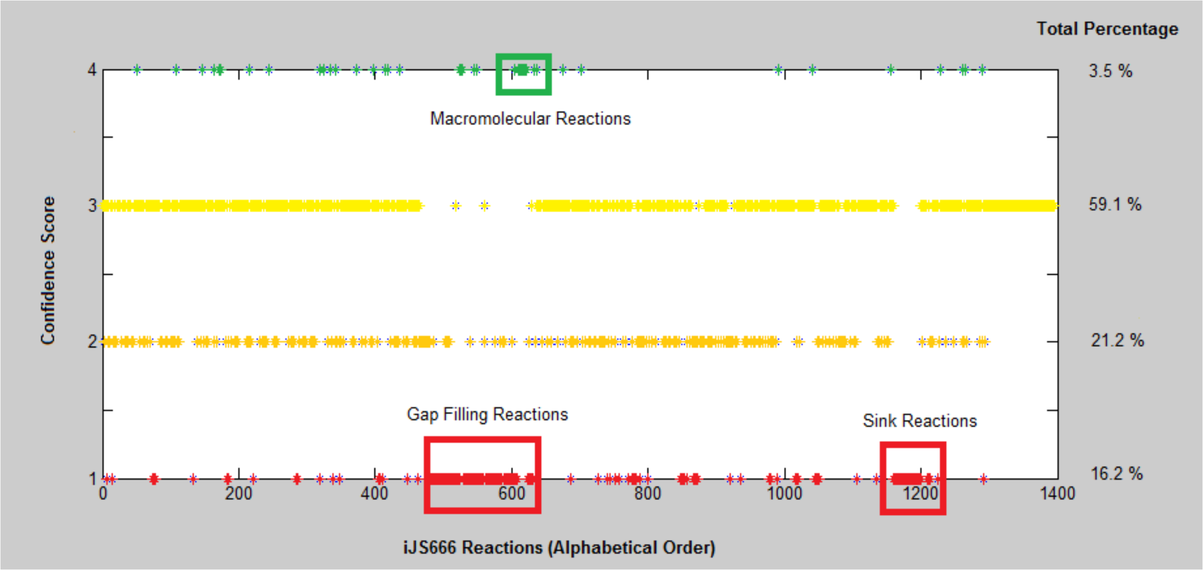


**Figure 3.3.** Confidence score attributed to the different reactions in iJS666. The total percentage of reactions with a given confidence score is displayed in right side of the figure. The reactions identified with corresponding colour rectangle correspond to artefacts identified in the algorithm.

As represented in the Figure 3.3, the majority of the reaction used in the metabolic model of iJS666 have a high confidence score (62.5% having a confidence score superior to 3) meaning that the great majority of those reactions are well defined GPR associations or could be also found in the phylogenetic related organism *R. ferrireducens*. The major contributions to the lowest confidence scores arise from reactions that were manually introduced (gap-filling reactions and sink reactions) since it was impossible to identify the corresponding gene that codes for the respective enzymes.

The initial tests carried by the model iJS666 were done using D-glucose as carbon source, as referred in Chapter 2.2.3.5., and simultaneously the system was tested under those initial constraints with reaction-by-reaction deletions in order to test the inherent robustness of the model. The results are presented on the Figure 3.4 and there was no significant variance in the distribution of those values using different carbon sources (data not displayed). As displayed by the Figure 3.4, approximately only 22% of the total reactions were essential reactions, causing a lethal effect when deleted, meaning that in the remaining cases the system have a great number of alternative pathways to reach the same objective, which is in general agreement with the properties of metabolic networks.

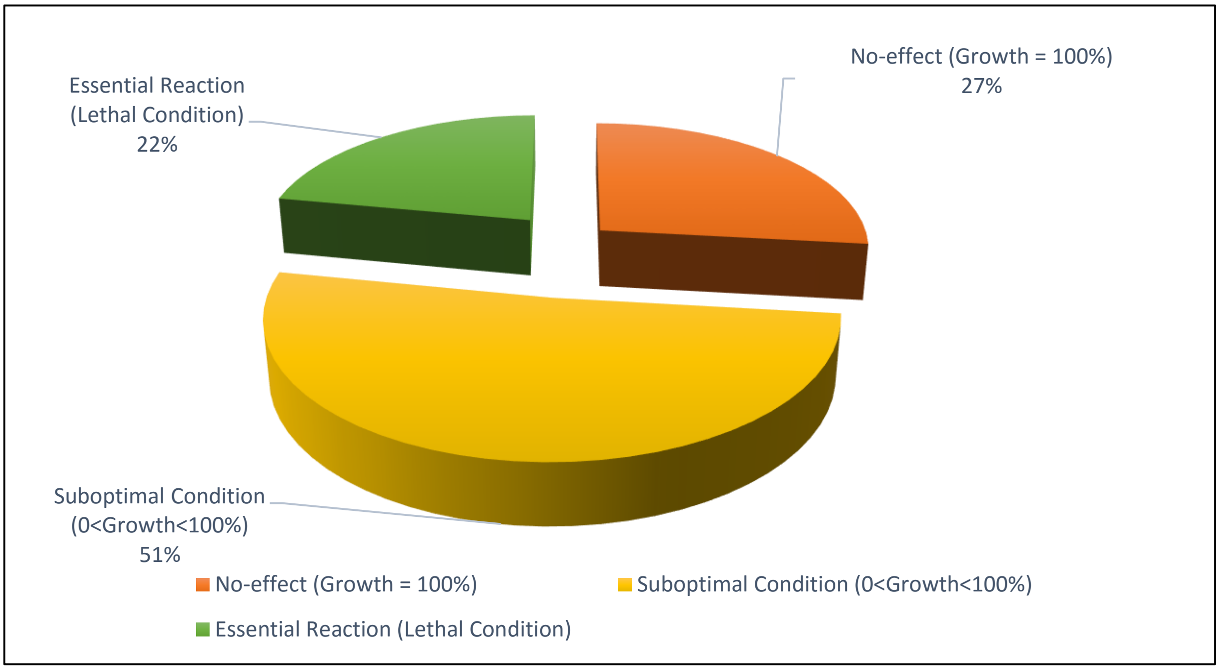


**Figure 3.4.** Robustness analysis of the iJS666 model. Substrate was set to be D-glucose and biomass equation was set as objective function.

## 3.2. iJS666 model correlation to *Polaromonas* sp. strain JS666

### 3.2.1. Culture Growth vs Model Growth

The validation of the iJS666 model was achieved by comparing model predictions with bibliographic data. The results from the initial validation, present in the Table 3.3, were generated using the scripts referred in Chapter 2.3. Initially, a flux value of 0.561 mmol gDW$^{-1}$ h$^{-1}$ (equal to the measured rate for cDCE degradation) was assumed for the specific substrate consumption rate when it was impossible to estimate the value using the expression also described in the Chapter 2.3. From 36 different mediums tested in the iJS666 model for biomass growth only one medium (phthalate medium) had a different outcome from the expected observable phenotype. Some possible explanations for this phenomena may be that the software PathwayTools could have made a false positive identification of the complete degradation pathway for phthalate or maybe the phthalate degradation really occurs in this microorganism although the experimental tested concentrations were excessive too allow growth. For example, Alexander, 2010, reported absence of *Polaromonas* sp. strain JS666 biomass growth for DCA although it was reported that *Polaromonas* sp. strain JS666 have the ability to degrade DCA and grow at very low concentrations of that carbon source (Nishino *et al*., 2013). Other possibility is that the predicted enzymes for phthalate degradation in Polaromonas sp. strain JS666 could be involved in utilization of (Meta)-phthalate or (Para)-phthalate isomers, rather than the tested (Ortho)-phthalate or, alternatively, they may encode phthalate ester degradation, which requires the ester form for transport or induction, as referred by Mattes et al., 2008.

In the last liquid culture mediums reported in the Table 3.3, the substrate consumption rate was insufficient to allow biomass growth. The initial used substrate consumption rates derived from the expression described in Chapter 2.3 and were under estimated, meaning that the real flux values were superior to the flux values used in the optimization. It is possible that the estimated values were miscalculated due to the fact that the integration values used were not taken solely from the exponential phase, where the maximum growth rate was obtained. For example the estimated/extrapolated substrate consumption rate for cDCE was 0.01028 mmol gDW$^{-1}$ h$^{-1}$ when the real observable value is 0.561 mmol gDW$^{-1}$ h$^{-1}$.

In order to further comprehend the specific substrate consumption rates influence on the biomass growth rate in the iJS666 model, specific growth rates using different Rs values were predicted for each substrate tested in liquid culture mediums. The results originated by the script *validationvariation.m* are reported in the Figure 3.5. Also, the oxygen consumption rates ($RO_2$) were assessed for the different Rs values, using the script *validationox.m*, as displayed in the Figure 3.6. As reported in the Figure 3.5, the model iJS666 was able to predict a specific growth rate similar to the observable values when higher substrate uptake rates were considered. It is perceptible that a minimum substrate uptake is necessary to predict a positive specific growth rate. This phenomena is associated with the fact that the system needs to produce enough energy for maintenance (NGAM) before any biomass growth. Due to the lack of chemostat data it was impossible to fully compare the model predictions to all experimental substrate consumption rates, but in the case of cDCE as carbon source the model predicted a specific growth rate of 0.0046 $h^{-1}$ while the extrapolated and experimental values for the specific growth rate were 0.0089 $h^{-1}$ and 0.0088 $h^{-1}$. Also, the predicted oxygen consumption rate (for cDCE degradation at a rate of 0.561 mmol $gDW^{-1}$ $h^{-1}$) was 1.05 mmol $gDW^{-1}$ $h^{-1}$. This implies that the model was predicting a suboptimal value for the biomass growth rate and an excessive oxygen consumption rate when compared to the experimental value of 0.43 mmol $gDW^{-1}$ $h^{-1}$ for $RO_2$ obtained in resting cells growing in cDCE. In order to understand the reason for this singularity in the model's predictive capabilities further assessments were done in the Chapter 3.2.2. Although, when the oxygen consumption rates predicted by the model iJS666 grown in other carbon sources were compared against their respective experimental values, a better agreement seems evident. Particularly, the predicted values of $RO_2$ for succinate and chloroacetate (1.59 mmol $gDW^{-1}$ $h^{-1}$ and 0.85 mmol $gDW^{-1}$ $h^{-1}$, respectively) were very similar to the experimental values obtained, represented in the Table 3.4. The minor differences may be due to the uncertainty of the relative protein biomass composition and average protein molecular weight, which were assumed to be equal when the model iJS666 was tested for different carbon source mediums.

**Table 3.3.** Real specific growth rate of *Polaromonas* sp. strain JS666 and predicted specific growth rate from iJS666 model using different mediums. Green and red rows represent carbon sources with positive and negative growth associated, respectively. Blue rows refer to liquid cultures further assessed in Figure 3.5 and extrapolated specific growth rate values determined as described in Chapter 2.3.*Experimental data from Alexander, 2010; $^{+}$Experimental data from Nishino *et al*., 2013.

| Tested Metabolite | Culture Type | Initial Concentrations [mM] | Growth or µ [h$^{-1}$] | Rs [mmol gDW$^{-1}$ h$^{-1}$] | Predicted µ [h$^{-1}$] | Causes |
|---|---|---|---|---|---|---|
| *Benzoate | Auxanograph | - | Positive | 0.561 | 0.0391 | - |
| *Catechol | Auxanograph | - | Positive | 0.561 | 0.0351 | - |
| *Hydroxyquinol | Auxanograph | - | ? | 0.561 | 0.0032 | - |
| Glucose | Auxanograph | - | Positive | 0.561 | 0.0417 | - |
| $^{+}$Glycolate | Auxanograph | - | Positive | 0.561 | Positive | Insufficient Rs rate |
| $^{+}$DCA | Auxanograph | - | Positive | 0.561 | 0.0305 | - |
| *Chloroacetaldehyde | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter associated |
| *Cyclohexane | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter & degradative pathway |
| *Cyclohexaneacetate | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter & degradative pathway |
| *DCP (dalapon) | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter & degradative pathway |
| *Ethane | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter associated |
| *Ethanol + Nitrate (Anoxic) | Auxanograph | 5-30 (Nitrate) | Negative | 0.561 | 0 | Not energetically viable |
| *Ethylcyclohexane | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter & degradative pathway |
| *Hexane | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter & degradative pathway |
| *4-hydroxybenzoate | Auxanograph | 2 | Negative | 0.561 | 0 | No transporter associated |
| *Naphthalene | Auxanograph | 2 | Negative | 0.561 | 0 | No transporter associated |
| *2-Nitrobenzoate | Auxanograph | 2 | Negative | 0.561 | 0 | No transporter associated |
| *4-Nitrobenzoate | Auxanograph | 2 | Negative | 0.561 | 0 | No transporter associated |
| *Phthalate | Auxanograph | 2 | Negative | 0.561 | 0.0376 | - |
| *Propane | Auxanograph | 0.2-10 | Negative | 0.561 | 0 | No transporter associated |
| *Thiosulfate | Auxanograph | 10-40 | Negative | 0.561 | 0 | No transporter associated |
| *Thiosulfate + Ethanol | Auxanograph | 10-40 | Negative | 0.561 | 0 | No transporter associated |

**To be continued…**

**Continuation of Table 3.3…**

| | | | | | | |
|---|---|---|---|---|---|---|
| *Acetate | Liquid Culture | 10.0 | 0.0375 | 0.08704 | Positive | Insufficient Rs rate |
| *cDCE | Liquid Culture | 1.3 | 0.0089 | 0.01028 | Positive | Insufficient Rs rate |
| *Chloroacetate | Liquid Culture | 6.0 | 0.0018 | 0.06287 | Positive | Insufficient Rs rate |
| *Cyclohexanecarboxylate | Liquid Culture | 6.0 | 0.0190 | 0.10162 | Positive | Insufficient Rs rate |
| *Cyclohexanol | Liquid Culture | 1.2 | 0.0097 | 0.04414 | Positive | Insufficient Rs rate |
| *Ethanol | Liquid Culture | 10.0 | 0.0090 | 0.04866 | Positive | Insufficient Rs rate |
| *Ferulate | Liquid Culture | 2.0 | 0.0035 | 0.05794 | Positive | Insufficient Rs rate |
| *Gentisate | Liquid Culture | 2.0 | 0.0079 | 0.02029 | Positive | Insufficient Rs rate |
| *Heptane | Liquid Culture | 0.9 | 0.0047 | 0.02651 | Positive | Insufficient Rs rate |
| *3-hydroxybenzoate | Liquid Culture | 2.0 | 0.0080 | 0.01326 | Positive | Insufficient Rs rate |
| *Octane | Liquid Culture | 0.2 | 0.0025 | 0.01992 | Positive | Insufficient Rs rate |
| *Protocatechuate | Liquid Culture | 2.0 | 0.0094 | 0.01357 | Positive | Insufficient Rs rate |
| *Salicylate | Liquid Culture | 0.6 | 0.0121 | 0.04594 | Positive | Insufficient Rs rate |
| *Succinate | Liquid Culture | 10.0 | 0.0231 | 0.17851 | Positive | Insufficient Rs rate |

**Table 3.4.** Converted specific oxygen consumption rates (RO$_2$) to iJS666 model. Initial data from Nishino et al., 2013.

| Tested Metabolite | RO$_2$ [nmol min$^{-1}$ mg protein$^{-1}$ ] | RO$_2$ [mmol gDW$^{-1}$ h$^{-1}$ ] |
|---|---|---|
| Glycolate | 95.00 | 3.15 |
| DCA | 23.60 | 0.78 |
| cDCE | 13.00 | 0.43 |
| Chloroacetate | 26.00 | 0.86 |
| Cyclohexanone | 276.00 | 9.15 |
| Succinate | 41.00 | 1.36 |
| MW protein [mg mmol$^{-1}$] | Relative Composition of Biomass [mmol protein gDW$^{-1}$] | |
| 34335.1 | 0.0161 | |

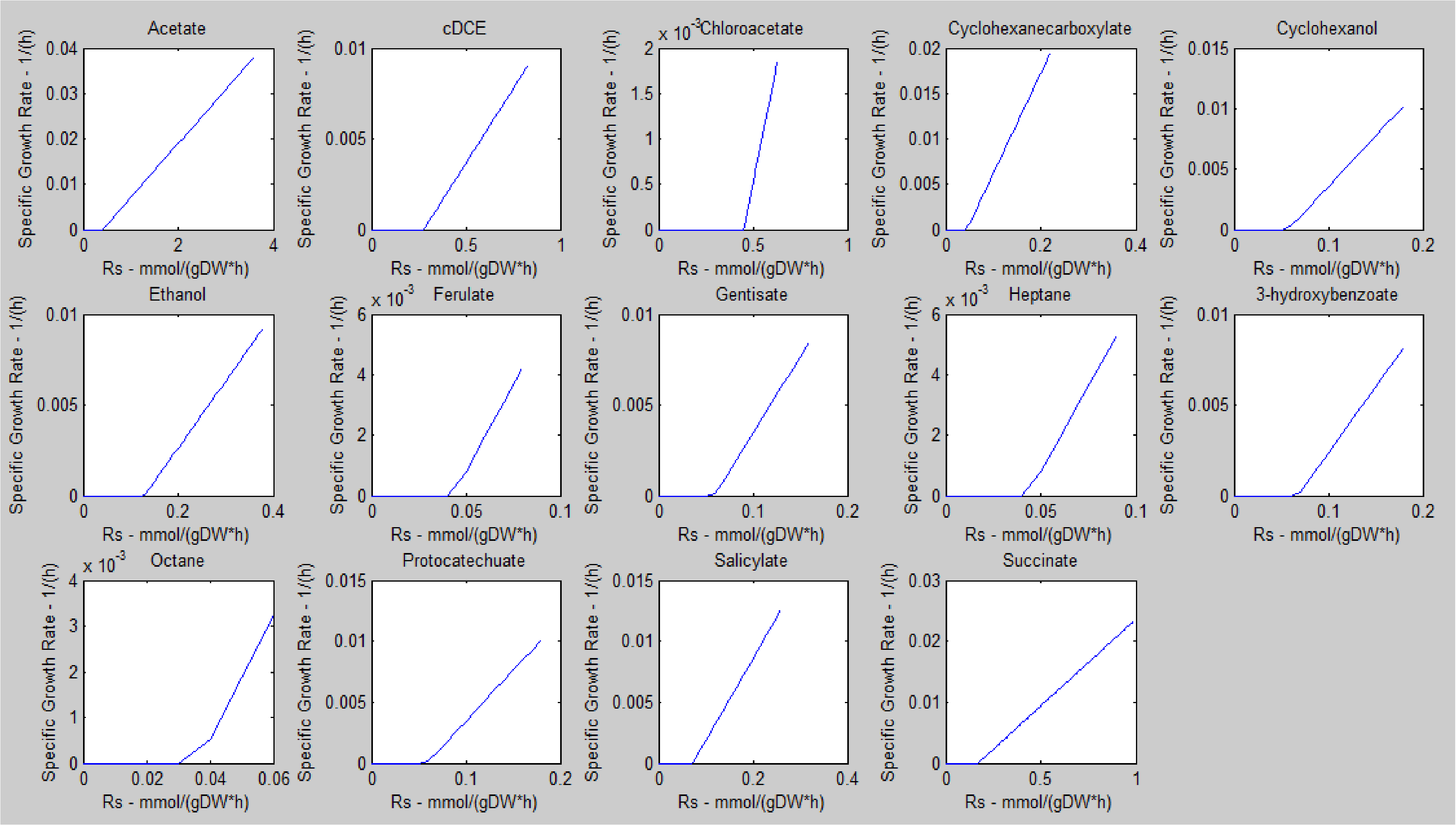


**Figure 3.5.** Influence of the specific substrate consumption rate on the specific growth rate in the iJ666 model predicted by FBA with biomass maximization as objective function. The last value represented in each individual plot corresponds to the extrapolated specific growth rate.

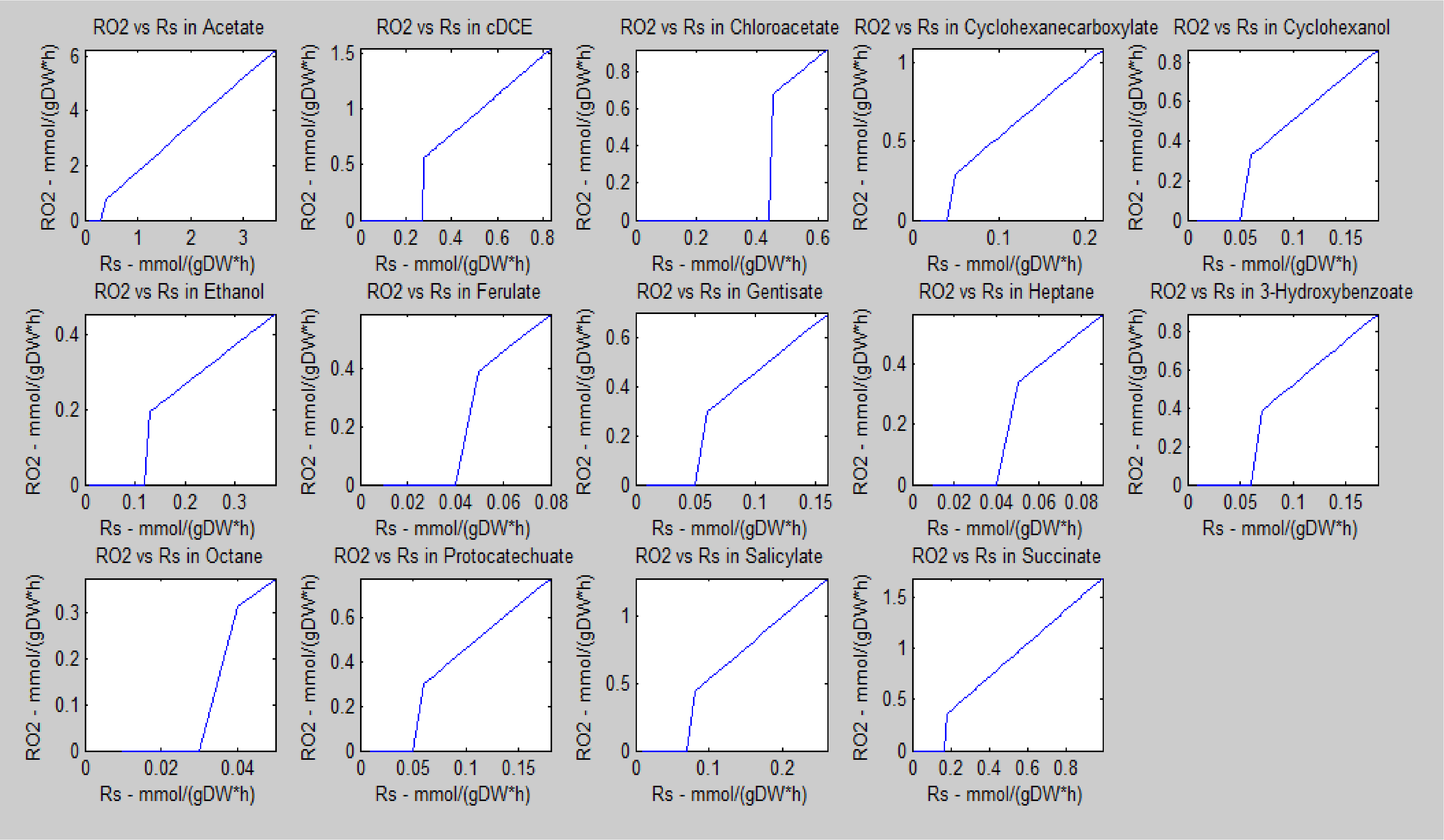


**Figure 3.6.** Influence of the specific substrate consumption rate on the specific oxygen consumption rate in the iJS666 model predicted by FBA with biomass maximization as objective function. The last value represented in each individual plot corresponds to the extrapolated specific growth rate.

## 3.2.2. cDCE degradation on iJS666 model

The cDCE degradation pathways inserted in the iJS666 model reconstruction were obtained from Cox, 2012, and Nishino *et al.*, 2013, and were introduced into the iJS666 model as displayed in the Table 3.5. Since those reactions were not present in the MetaCyc database they were manually mass-balanced. As reported by the same authors, the most likely cofactor used in the initial cDCE monooxygenase (Bpro_5301) oxidation is a ferredoxin molecule. In the iJS666 model the reduced ferredoxin molecule was replaced by the preceding regenerating cofactor, NADH. This modification had no influence in the observable fluxes, as referred in Chapter 2.2.2, since every cofactor had interchangeable redox reactions.

As displayed in Table 3.5, the two primary cDCE degradation pathways and the glycolate degradation pathway merge into the production of the glyoxylate metabolite downstream. In the Table 3.5, the intracellular fluxes for those reactions are also displayed when the system was solved for the maximization of the biomass rate (μ=0.0046 $h^{-1}$) with a specific substrate (cDCE) consumption rate of 0.561 mmol $gDW^{-1}$ $h^{-1}$.

**Table 3.5.** cDCE degradation pathways in iJS666 and respective fluxes. (μ=0.0043 $h^{-1}$, Rs=0.561 mmol $gDW^{-1}$ $h^{-1}$)

| ID | Name of Reaction | Reaction | Gene | Fluxes [mmol $gDW^{-1}$ $h^{-1}$] |
|---|---|---|---|---|
| **Glycolate Initial Degradation Pathways** | | | | |
| Glycolate Oxidation | | | | |
| RXN-969 | glycolate oxidase | glycolate + oxygen → glyoxylate + hydrogen peroxide | glcF, glcE | 0 |
| **cDCE Initial Degradation Pathways** | | | | |
| Monooxygenase Main Pathway | | | | |
| NEWRXN 6661b | cytochrome P450 | cDCE + oxygen + NADH + $H^+$ → 2,2-dichloroacetaldehyde + $NAD^+$ + $H_2O$ | Bpro_5301 | 0.561 |
| NEWRXN 6662 | dichloroacetaldehyde dehydrogenase | 2,2-dichloroacetaldehyde + NAD+ + H2O → 2,2-dichloroacetate + NADH + $H^+$ | ? | 0.561 |
| NEWRXN 6663 | dichloroacetic acid dehalogenase | 2,2-dichloroacetate + $H_2O$ → chloroglycolate + chloride + $H^+$ | Bpro_5186<br>Bpro_0530 | 0.561 |
| NEWRXN 6664 | chloroglycolate spontaneous halogenase | chloroglycolate → chloride + glyoxylate + $H^+$ | - | 0.561 |
| Monooxygenase Epoxide Pathway | | | | |
| · Spontaneous Dechlorination | | | | |
| NEWRXN 6665b | cytochrome P450 | cDCE + oxygen + NADH + H+ → cDCE-epoxide + NAD+ + $H_2O$ | Bpro_5301 | 0 |
| NEWRXN 6666 | cytochrome P450 | cDCE-epoxide + $H_2O$ → 1,2-dichloroethane-1,2-diol | Bpro_5301 | 0 |
| NEWRXN 6667 | 1,2-dichloroethane-1,2-diol spontaneous dehalogenase | 1,2-dichloroethane-1,2-diol → glyoxal + 2 chloride | - | 0 |

**To be continued…**

**Continuation of Table 3.5…**

| | | | | |
|---|---|---|---|---|
| · Glutathione Catalysed Dechlorination | | | | |
| NEWRXN 6668 | cDCE-glutathione synthase | cDCE-epoxide + glutathione → GS-conjugateI + chloride | Bpro_0645 | 0 |
| NEWRXN 6669 | Gsconjugate dehalogenase I | GS-conjugateI + H2O → GS-conjugateII + chloride | ? | 0 |
| NEWRXN 66610 | Gsconjugate dehalogenase II | GS-conjugateII → glutathione + glyoxal | ? | 0 |
| NEWRXN 66613 | glyoxylate synthase | glyoxal + $H_2O$ → glyoxylate + 3 $H^+$ | Bpro_0577 | 0 |
| Shared Degradation Pathway | | | | |
| GLYOCARBOLIG-RXN | glyoxylate carboligase | 2 glyoxylate + $H^+$ → $CO_2$ + tartronate semialdehyde | Bpro_4561 | 0.2798 |
| SERINE-GLYOXYLATE-AMINOTRANSFERASE-RXN | serine-glyoxylate transaminase | glyoxylate + L-serine ↔ hydroxypyruvate + glycine | Bpro_0548 | -0.0022 |
| MALSYN-RXN | malate synthase | acetyl-CoA + glyoxylate + $H_2O$ → (S)-malate + coenzyme A + $H^+$ | Bpro_4517 | 0.0035 |

As described in Table 3.5, the iJS666 model predicted an intracellular flux value of 0.561 mmol gDW$^{-1}$ h$^{-1}$ for the reactions of the main monooxygenase pathway, using the cytochrome P450 enzyme that is coded by the Bpro_5301 gene. The predicted intracellular flux for that reaction was equal to the absolute value of the substrate consumption rate meaning that all degraded cDCE is oxidized by this reaction and no side epoxide metabolite is produced. The iJS666 model was, therefore, unable to predict the cDCE-epoxide that is a minor side product, as reported in Nishino *et al*., 2013.

Also, the iJS666 model predicted flux values of 0.2798, -0.0022 and 0.0035 mmol gDW$^{-1}$ h$^{-1}$ for the glyoxylate carboligase (Bpro_4561), serine-glyoxylate transaminase (Bpro_0548) and malate synthase (Bpro_4517), respectively. In the reaction catalysed by the serine-glyoxylate transaminase, the negative flux value means that glyoxylate and L-serine are produced from hydroxypyruvate and glycine. The overall sum of the previously reported fluxes, taking in consideration their respective stoichiometry, adds up to the absolute value of the substrate consumption rate. The fact that glyoxylate follows the anabolic tartronate semialdehyde biosynthesis pathway over 400 % times more than the (S)-malate/TCA pathways could be explained, at first glance, by some unbalanced proton in either of the different pathway reactions, since the malate synthase would produce (S)-malate and would be further incorporated into TCA cycle, a more direct route to the central metabolism. However, in an unlimited energy system (obtained by allowing costless proton exchange) only a small increment in the absolute fluxes values was observed for the reactions catalysed by the serine-glyoxylate transaminase and malate synthase, meaning that the glyoxylate carboligase reaction was still by far (~350% glyoxylate usage) the most prevalent anabolic pathway. A closer look to the reactions proceeding the tartronate semialdehyde biosynthesis revealed that hydroxypyruvate was produce by the hydroxypyruvate isomerase and then reduced to D-glycerate, which was further mainly incorporated into glycolysis central pathway. In the unlimited energy system the observed maximum growth rate was 0.0070 $h^{-1}$ meaning that the main cause for the underestimated biomass growth was probably energy/redox potential production. In order to verify the cause of such energetic bottleneck, a FVA was performed to analyse, which flux values were reaching the predefined upper and lower bounds and/or were invariable, as displayed in Figure 3.7.

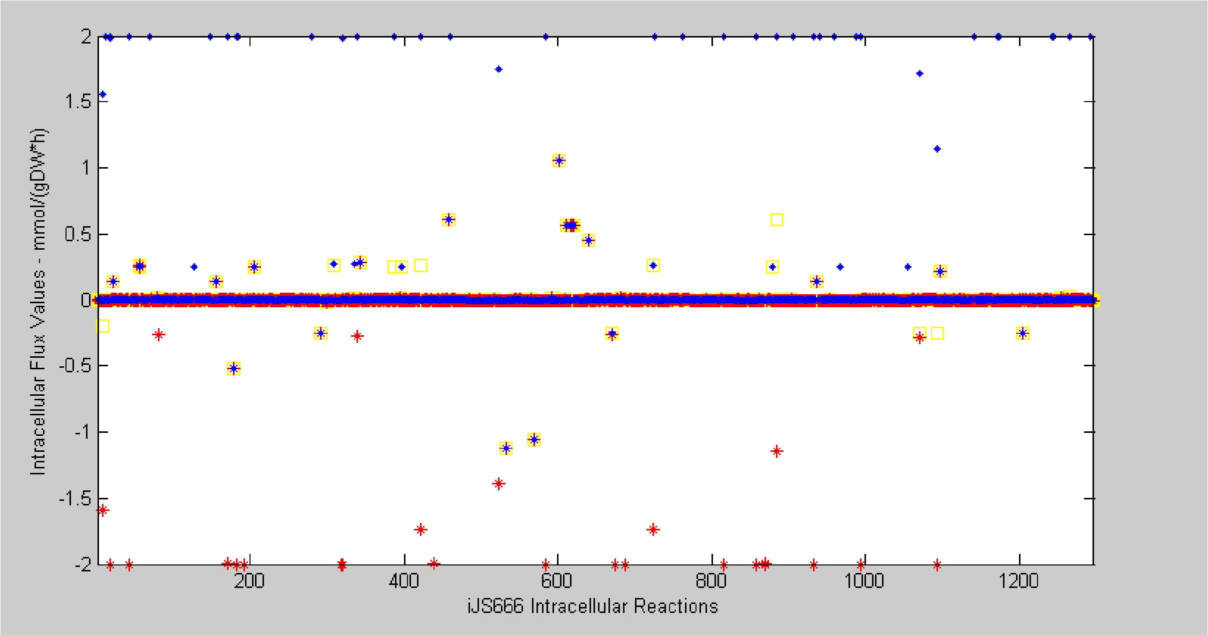


**Figure 3.7.** FBA and FVA of intracellular iJS666 reactions. Red stars and blue points represent minimal and maximal FVA predicted values, respectively. Yellow squares represent the flux values acquired from FBA with the biomass production set as objective function and with Rs=0.561 mmol gDW-1 h-1.

As displayed in Figure 3.7, no flux value reached the lower bound or upper bound of the defined system meaning there are no bottleneck reactions in the iJS666 model. Although, some intracellular reactions seemed to have invariable flux values (minimal and maximum values obtained in FVA are equal). Those reactions were easily identified by screening the Figure 3.7 for the 3 representation symbols overlapping each other and the reactions that had an absolute flux value superior to 0.1 mmol $gDW^{-1}$ $h^{-1}$ were represented in Supplementary Data 2. As suspected, those reactions are involved in redox potential and ATP synthesis as well in the main carbon metabolism. However, it was not possible to identify which reactions contributed the most to the underrated specific biomass growth rate prediction.

During the analysis of the intracellular fluxes, two reactions from the cDCE-epoxide degradation pathway, that recycle glutathione, were found to be incorrectly annotated by the PathwayTools software. As described in Cox, 2012, glutathione may have a major relevance not only in the degradation of the cDCE-epoxide, but also in detoxifying intracellular (di)chloroacetaldehydes by limiting their cytosolic concentration. As demonstrated in the Table 3.6, the re-annotated reactions for the glycolyl-glutathione lyase and hydroxyacyl-glutathione hydrolase had no influence on the iJS666 model predictions when biomass reaction was maximized.

**Tabela 3.6.** Re-annotated reactions for the glyoxylate synthesis from glyoxal and respective fluxes (μ=0.0046 $h^{-1}$, Rs=0.561 mmol $gDW^{-1}$ $h^{-1}$).

| ID | Name of Reaction | Reaction | Gene | Fluxes [mmol $gDW^{-1}$ $h^{-1}$] |
|---|---|---|---|---|
| **Deleted Reaction** | | | | |
| NEWRXN 66613 | glyoxylate synthase | glyoxal + $H_2O$ → glyoxylate + 3 $H^+$ | Bpro_0577 | 0 |
| **Re-annotated Reactions** | | | | |
| GLYOXI-RXN2 | glycolylglutathione lyase | glyoxal + glutathione → S-glycolyl-glutathione | Bpro_3549<br>Bpro_2168 | 0 |
| GLYOXII-RXN2 | Hydroxyacyl-glutathione hydrolase | S-glycolyl-glutathione + $H_2O$ → glutathione + glyoxylate + 3 $H^+$ | Bpro_2055 | 0 |

As displayed in the Table 2.1, the glutathione metabolite was not included in the iJS666 initial biomass composition. Due to the fact that many reactions may use this metabolite in *Polaromonas* sp. strain JS666 under the cDCE degradation and to the fact that a glutathione-S-transferase (Bpro_0645) was the most overexpressed gene on cDCE medium, it is very unlikely that this compound is not produced in the cytosol of this microorganism. In order to evaluate the influence of this metabolite on the predicted specific biomass growth rate, the biomass equation for the iJS666 model was altered in order to incorporate different final concentrations of glutathione in the biomass, as displayed in the Figure 3.8.

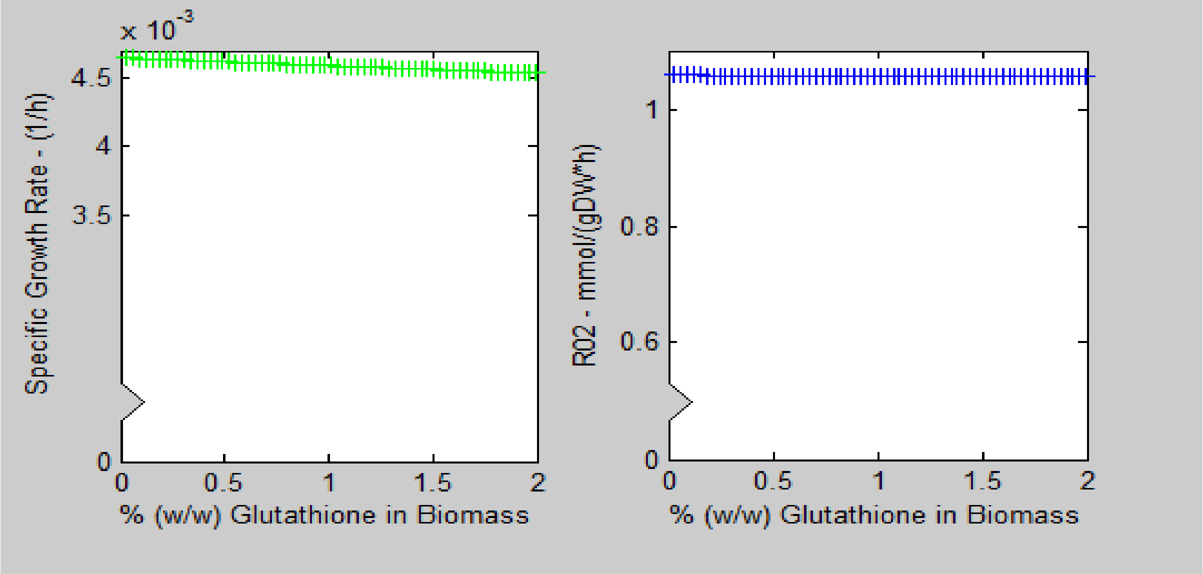


**Figure 3.8.** Glutathione influence on the specific growth rate (left) and specific oxygen consumption rate (right).

As presented in Figure 3.8, the inclusion of glutathione in the biomass reaction had a minor negative effect on the specific growth rate. The predicted specific growth rate and $RO_2$ values under these conditions may reflect the expensive cost for glutathione biosynthesis, since this molecule requires more ATP consumption than other biomass precursors due to the use of energetically expensive amino acid precursors as L-cysteine.

The maximum tested percentage of glutathione tripeptide in the biomass reaction was way above the normal concentrations when comparing to others cofactors present in the biomass (i.e. Coenzyme A have a final percentage of only 0.3% (w/w)). Since the yield of glutathione in the biomass was not experimentally measured, posterior analysis of the flux variation using different objective functions were assessed with 2% (w/w) glutathione and without glutathione. The overrated pre-set value of glutathione in the biomass allowed to easily identify differences between those cases. The results of the maximization of different objective functions are reported in Supplementary Data 3.

When the model iJS666 was tested for the degradation of cDCE (Rs = 0.561 mmol gDW$^{-1}$ h$^{-1}$) with the biomass equation as the objective function, the intracellular fluxes for the different cDCE degradation pathways remained practically the same. The main difference was that the model with 2% (w/w) glutathione in the biomass allowed a maximum value of 2.87x10$^{-5}$ mmol gDW$^{-1}$ h$^{-1}$ for all fluxes in the epoxide-derived pathways instead of 8.61x10$^{-6}$ mmol gDW$^{-1}$ h$^{-1}$ for the model without glutathione incorporated in the biomass. Since one of the pathway that degrades the cDCE-epoxide metabolite does not depend on the use of glutathione for further degradation (spontaneous dehalogenation), the predicted data reflect the effect of the glutathione in the redox potential/energy production. The model that included 2% (w/w) glutathione in the biomass equation allowed higher import fluxes through the sink reactions, suggesting that the system could use the carbon source to the biosynthesis of those compounds more efficiently. The ATP synthase flux value slightly decreased for the model incorporating glutathione however higher flux values in the electron transport chain and other cofactor interchangeable redox reactions were reported in the end of the list presented in Supplementary Data 3, meaning that glutathione as a biomass precursor had a positive effect on the redox state of the iJS666 model. As for the fate of the glyoxylate, the model incorporating glutathione in the biomass also predicted maximum flux values for serine-glyoxylate aminotransferase and malate synthase slightly increased. No alteration was detected in the glyoxylate carboxylase maximum flux rate under these conditions confirming that this reaction is highly demanded for ATP production as referred before.

Using the minimization of NADH and ATP consumption reactions fluxes as objective functions in the iJ666 model, a higher maximum flux rate for the epoxide-derived pathways reactions was predicted, 0.135 (~24% of the maximum substrate consumption flux) and 0.171 mmol gDW$^{-1}$ h$^{-1}$ (~30% of the maximum substrate consumption flux), respectively. Since the biomass equation was not subject to optimization in the later cases, it is important to refer that the specific growth rate was approximately null. Once again the presence of glutathione in the biomass equation had no effect on the fluxes predicted for the epoxide-derived pathway. Further analysis of the predicted fluxes and methodologies used suggested that the iJS666 model and used solver (using norm one minimization to block loops) were unable to predict the effect of glutathione in those reactions. Glutathione is regenerated to the initial redox state every time it is consumed in linear dependent reactions. A more suitable way to predict the epoxide-related flux values could be the use of ODE although the concentrations of those epoxide intermediates is almost impossible to quantify. Also, all intracellular flux for the reactions that used glyoxylate had a higher maximum value under minimizing ATP/NADH consumption. Even the serine-glyoxylate aminotransferase reaction changed directionality, having a maximum value of positive 0.602 and 0.514 mmol gDW$^{-1}$ h$^{-1}$ for the minimization of ATP and NADH, respectively. The maximum flux value for malate synthase's reaction was 0.882 mmol gDW$^{-1}$ h$^{-1}$ and constant for the different minimization objectives. Flux values superior to the observable substrate consumption rate may suggest that glyoxylate and the other metabolites used as substrates on those reactions are being produced downstream, since the minimization of the norm to block loops was used. Nevertheless, it was possible to contemplate that these fluxes may vary, depending on the cellular objectives and their experimental quantification may have the most important significance in the validation and comprehension of the metabolic state of the microorganism.

In order to understand how the ATP consumption rate influenced the predicted specific biomass growth, the iJS666 model was also tested using different NGAM and GAM flux values. Simultaneously, different transporter reactions were defined to infer the transport mechanism of cDCE and chlorine, as reported in the Figure 3.9. In all those iJS666 model predictions, the NGAM flux was the parameter that had the most influence in the biomass specific growth rate. Excluding active export of chlorine, where the solution was unfeasible, and all the other transport systems had a linear dependency to the NGAM flux value. Also, varying the GAM flux allowed the prediction of higher specific growth rates for the different models. In all the cases where active transport was assumed, either for cDCE or chloride, no specific growth rate had similar values to the reported experimental value of 0.0089h$^{-1}$. For the remaining tested transport systems, only the model that assumed cDCE symport and chloride antiport (three protons entering the cytoplasm for each for each cDCE degraded) had a minor decrease in the specific growth rate, indicating that the cause for the iJS666 model predicted values for specific growth rate may be mainly underrated due to an ATP production shortcoming. Jennings et al., 2009, findings revealed the upregulation of a putative ABC transport system and two sodium/solute symporters in cDCE grown Polaromonas sp. strain JS666 cells, meaning that further work should be done in order to assess the function of those transporters and refine the model predictions.

.

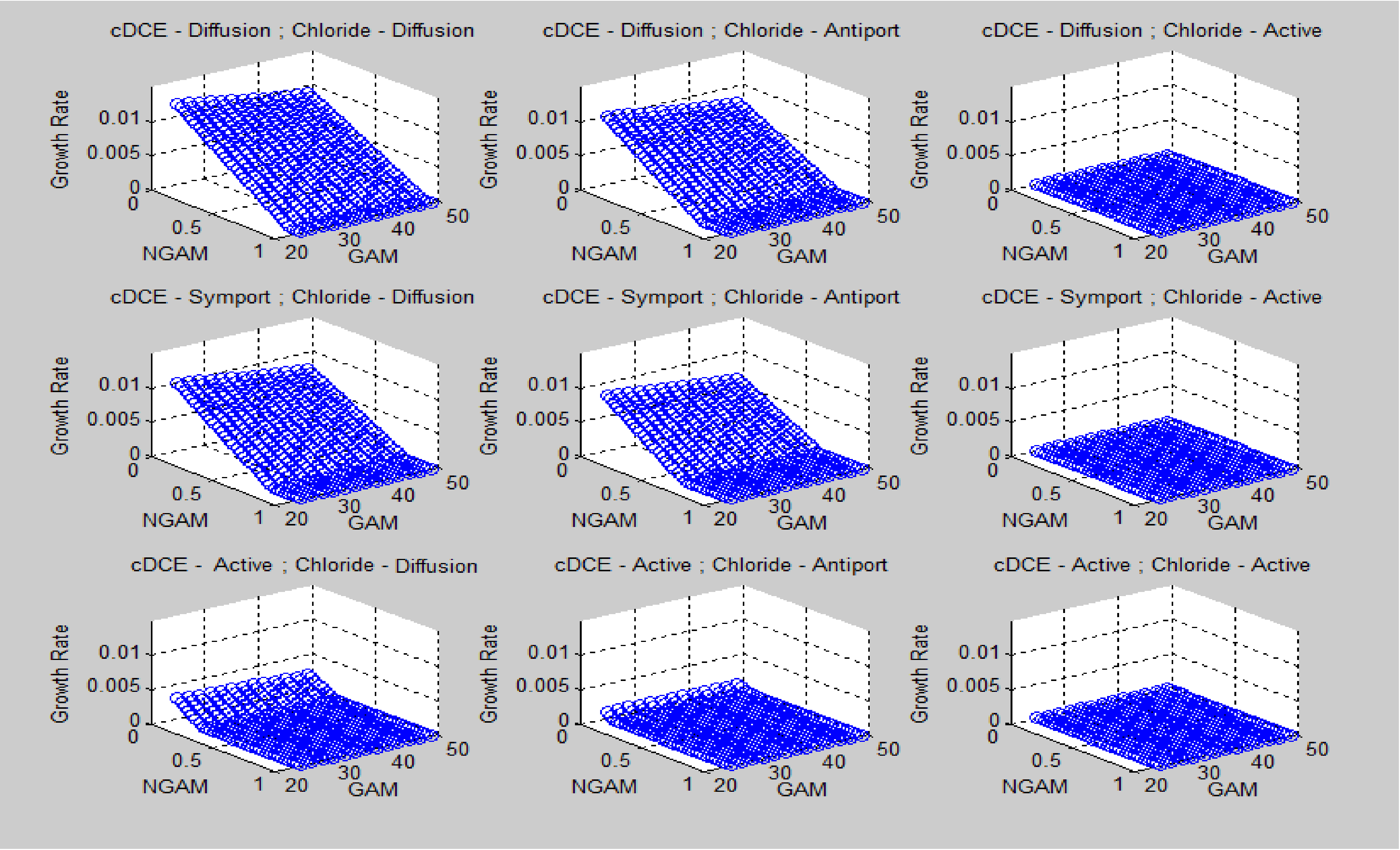


**Figure 3.9.** Influence of NGAM (mmol h-1), GAM(mmol gDW-1 h-1) and different types of cDCE and chloride transporter in the specific growth rate (Biomass as objective and Rs=0.561 mmol gDW-1 h-1).

As reported in the work of Giddings *et al*., 2010a, the growth of cDCE degrading bacteria in aerobic plumes is advantageous when byproducts of anaerobic biologic activity, like sulfides, are present in the groundwater. In order to assess the influence of different sulfur sources on the iJS666 model predicted specific growth rates, different sulfur substrates were allowed to enter the system by changing the respective exchange reactions, as represented in the Figure 3.10.

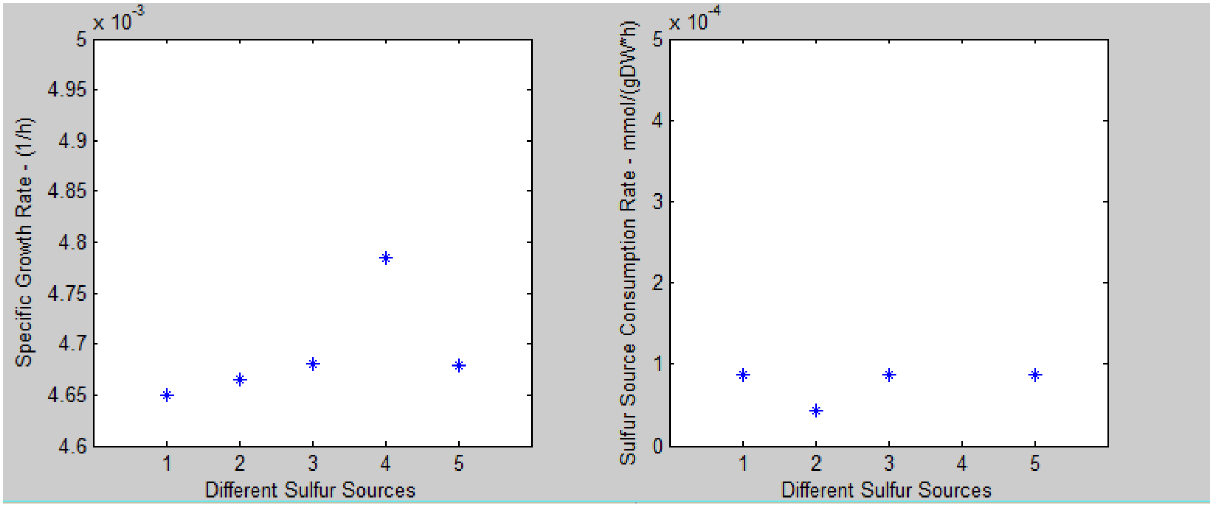


**Figure 3.10.** Predictions made by the iJS666 model for the specific growth rate (left) and sulfur consumption rates (right) using the following sulfur source: 1 - Sulfate; 2 - Thiosulfate; 3 – Sulfite; 4 - Hydrogen Sulfide; 5 - Elementar Sulfur. The predicted hydrogen sulfide consumption rate was 0.0128 mmol gDW-1 h-1 and is not displayed in the figure.

From the Figure 3.10, it is evident that the model is capable of predict the specific growth rate (positive growth) under the conditions described previously. Moreover, all the predictions of the iJS666 model for the specific growth rate using sulfur sources different than sulfate had superior rate values. This means that less oxidized sulfur species favored the redox potential of the model, allowing it to use the redox potential to catalyze other relevant intracellular reactions. Thiosulfate carries two sulfur atoms and that is the reason the sulfur consumption rate being lower than sulfate/sulfite but still the final specific growth rate is slightly increased. The hypothesis that *Polaromonas* sp. strain JS666 could grow chemolithotrophically by oxidizing thiosulfate was tested previously in Mattes *et al*., 2008 and the results were negative. However, the growth tests with thiosulfate (0 to 40 mM) were done assuming ethanol as carbon source and the model iJS666 was able to predict the absence of growth under this conditions, but not when cDCE carbon source was taken in consideration, as seen in Figure 3.10. By the other hand, hydrogen sulfide had an overrated consumption rate value compared to the other sulfur sources. It seems that a proton unbalanced was the cause for the reactions that used this metabolite but after exploiting the intracellular fluxes of those reactions, it was assessed that hydrogen sulfide react with acetyl-L-serine and acetyl-L-homoserine to generate L-cysteine and L-homocysteine without using any reductive cofactors.

Since the majority of the biomass content is composed of protein, as referred in Chapter 2.2.2.1, the nitrogen source may be a crucial factor in the prediction of specific biomass growth by the iJS666 model. With regard to the nitrogen influence in the specific growth rate, different nitrogen sources were assessed, as demonstrated in Figure 3.11. Similar to sulfur sources, the influence of the different nitrogen sources in the predicted specific growth rate was due to an increased intake nitrogen source consumption rate from the extracellular medium, as displayed in the Figure 3.11. All the different nitrogen sources tested had a higher predicted consumption rate than the prediction using ammonium as nitrogen source in the iJS666 model.  The maximum variation in the specific growth rate was greater when comparing to the maximum variation using any of the sulfur sources tested, because nitrogen atom is more prevalent in protein composition, and therefore, as in biomass equation. Although the iJS666 model predicts higher growth rates for nitrate, nitrite and hydroxylamine, those values may occur, because the exceeding consumption rates contribute to nitrate/nitrite/hydroxylamine respiration that generates proton motive force for energy, which as reported before is limiting the predicted growth rate of the iJS666 model. In a similar way, the model assumes that ammonia is transformed into ammonium with the release of one proton to cytosol that increases the NADH redox potential to the electron transport chain and, consequently, the generation of more energy although, due to the reasons reported before, these predictions may not be accurate.

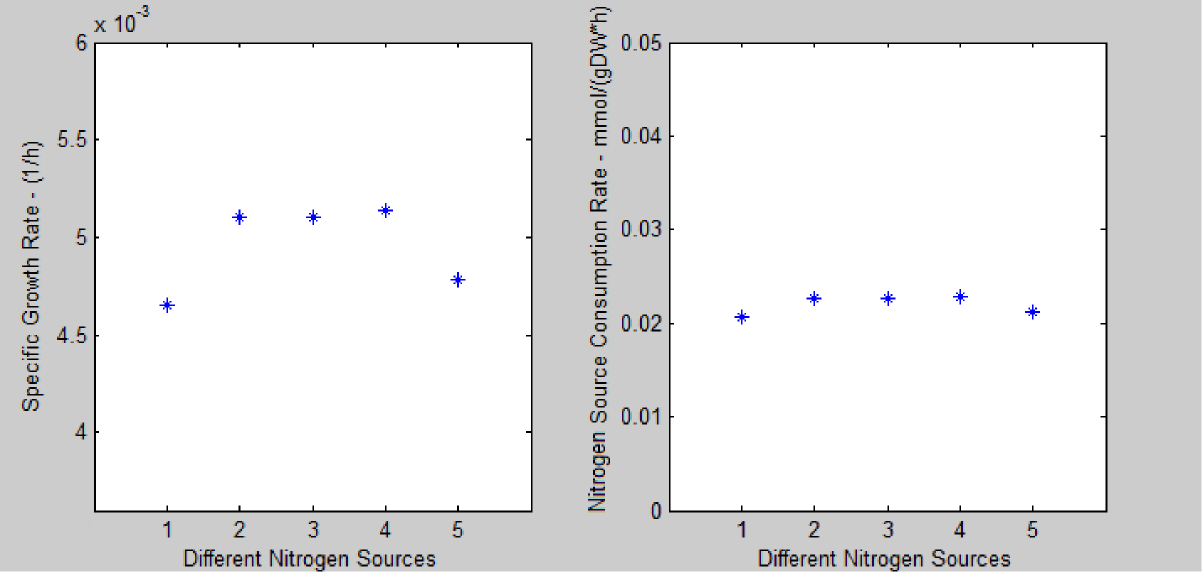


**Figure 3.11.** Predictions made by the iJS666 model for the specific growth rate (left) and nitrogen consumption rates (right) using the following nitrogen sources: 1 – Ammonium; 2 – Nitrate; 3 – Nitrite; 4 - Hydroxylamine; 5 - Ammonia.

## 3.3. Metabolites for Biostimulation

As reported in the Chapter 2.6, after the initial validation of the iJS666 model further calculations were made in order to predict biostimulant compounds to be used in the bioaugmentation of Polaromonas sp. Strain JS666 in the degradation of cDCE. From the initial 1073 tested metabolites only 315 had positive scores for the criteria described in Chapter 2.6, and only 74 of those had a predicted increment in the specific growth rate above 2%. The top 30 metabolites that had the most influence on specific growth rate are displayed in the Table 3.7.

**Table 3.7.** Predicted biostimulant compounds by iJS666 on cDCE medium, (Rs=0.561 mmol gDW$^{-1}$ h$^{-1}$). Rb – specifc biostimulant consumption rate (mmol gDW$^{-1}$ h$^{-1}$).

| Putative Biostimulant | Increment in % of μ | Rb [mmol gDW$^{-1}$ h$^{-1}$] | % of salvaged reactions | # of activated reactions |
|---|---|---|---|---|
| sarcosine | 29.6 | -2.7E-02 | 86.0 | 18 |
| histidinal | 29.4 | -1.5E-02 | 84.2 | 26 |
| L-histidine | 28.0 | -1.1E-02 | 84.4 | 29 |
| 3-phospho-L-serine | 27.6 | -2.6E-02 | 85.8 | 35 |
| L-citrulline | 27.4 | -8.8E-03 | 85.8 | 29 |
| L-serine | 25.9 | -2.6E-02 | 86.0 | 23 |
| 3-ureidopropionate | 24.5 | -1.4E-02 | 91.3 | 59 |
| L-asparagine | 22.5 | -1.4E-02 | 86.4 | 22 |
| 5,6-dihydrouracil | 20.8 | -2.0E-02 | 90.9 | 23 |
| N-formyl-N-5-phospho-β-D-ribosyl-glycinamide | 18.2 | -4.7E-03 | 85.6 | 35 |
| glycine | 17.6 | -2.4E-02 | 84.8 | 21 |
| N-5-phospho-β-D-ribosyl-glycinamide | 16.1 | -3.6E-03 | 85.8 | 15 |
| L,L-diaminopimelate | 16.0 | -2.7E-01 | 87.0 | 23 |
| 2-iminoacetate | 14.0 | -2.4E-02 | 87.8 | 13 |
| creatine | 13.5 | -7.8E-03 | 86.0 | 50 |
| CDP | 12.0 | -2.1E-03 | 85.4 | 23 |
| orotidine-5-phosphate | 12.0 | -2.1E-03 | 86.0 | 34 |
| UMP | 11.7 | -2.1E-03 | 90.9 | 35 |
| UDP | 11.6 | -2.1E-03 | 85.4 | 14 |
| uridine | 11.5 | -2.1E-03 | 85.8 | 17 |
| CTP | 11.4 | -2.1E-03 | 85.8 | 53 |
| UTP | 11.0 | -2.1E-03 | 86.0 | 26 |
| 5-phospho-β-D-ribosylamine | 10.8 | -3.4E-03 | 87.4 | 24 |
| cytosine | 10.2 | -7.6E-03 | 86.2 | 21 |
| GDP | 10.0 | -1.5E-03 | 87.2 | 35 |
| GTP | 10.0 | -1.5E-03 | 87.8 | 37 |
| ppGpp | 8.9 | -1.5E-03 | 86.6 | 30 |
| pppGpp | 8.4 | -1.5E-03 | 86.6 | 25 |
| L-tryptophan | 7.6 | -2.2E-02 | 87.8 | 36 |
| lipid-A-core | 6.5 | -4.5E-05 | 81.7 | 12 |

The majority of the putative biostimulants presented in the Table 3.7 are metabolites involved in the amino acid, cofactor and nucleotide metabolism maybe due to the fact that those compounds are used in the major cellular biomass components. Some amino acids that were predicted as putative biostimulants have a higher influence in the specific growth rate comparing to other cellular metabolites since the highest biomass coefficient is protein. The script *testsub.m* could screen those metabolites from their precursors due to higher increment on the specific biomass growth rate.

The hypothesis that metabolites which have a smaller number of inducted reactions are more suitable for biostimulation, due to the fact that the microorganism have to undertake minor adaptions to the pre-existent metabolic network (i.e. MOMA) (Palsson, 2009), were also taken in consideration. It's evident that using these metabolites as biostimulants the iJS666 model predicts that the majority of active metabolic pathways remains equal as seen by the percentage of salvaged reactions from cDCE degradation metabolism. Also, iJS666 model assumes that less energetical nucleotides give higher specific growth rate predictions (i.e. 12% growth rate increase with CDP and 11.4% increase using CTP). It is possible that those metabolites are used by the iJS666 model reactions in that particular state although, in reality, the more energetical metabolites are most appreciated by the cellular metabolism.

L-serine and glycine metabolites are direct inputs in some of the downstream degradation reactions of cDCE, more specifically in the degradation pathway of glyoxylate. The model iJS666 still predicted the activation of 23 and 21 reactions, respectively, that were not active on the model iJS666 set with the cDCE as solo carbon source, meaning that those metabolites may influence many more pathways than those previously reported. Although not displayed in the figure, the predictions that generate biostimulants wich activate the smallest number of not previously used reactions are those which had the smallest increment in the specific growth rate. This means that the model was very robust and that those biostimulants did not have a major impact in the system individually. A common feature to all the predicted biostimulants presented on Table 3.7 is that the biostimulant consumption rate (Rb) was always much smaller than the substrate consumption rate (only L,L-diaminopimelate had reported values in the same order of magnitude) implicating that in the final balanced system those biostimulants only contribute to a small amount of the biomass constitution and thus that only small amounts of those compounds would be needed on medium and putatively cDCE will still be degraded at the same rate with an improvement on the yield of biomass production.

# 4. Conclusion

As reported by Cox, 2012, "*The evolution, regulation and potential for improvement of the cDCE pathway are compelling mysteries that can be explored now that the basic biochemistry and molecular biology of the pathway has been established*". Hence the overall goal of this dissertation was to generate a comprehensive genome-scale metabolic model for the *Polaromonas* sp. strain JS666 to predict biostimulants for improving cDCE bioremediation and better understand the metabolic network and their relationship to the observable phenotype since the biodegradation capabilities of this microorganism, previously validated in field, did not met the desired requirements.

In particular, the major methodological objectives of this research were to build an updated genome-scale metabolic reconstruction, a functional and validated genome-scale metabolic model and, finally, the prediction of biostimulants using the validated GEM using FVA and FBA approaches. Also, the aerobic oxidation pathway for cDCE in Polaromonas sp. strain JS666 was studied for a system's biology point of view. The main downfalls of these methodologies are that the predicted intracellular fluxes and uptake reactions may not account for post-translational modifications, specific and divergent activity of enzymes or repression effects of some metabolites.

The developed model, iJS666, could only be partial validated since biochemical data for *Polaromonas* sp. strain JS666 was scarce. The model iJS666 predicted growth under the previously tested carbon sources, predict different growth rates in the same order of magnitude of the experimental reported values and yielded a reasonable prediction for other cellular uptake rates. Also, the model iJS666 was built in such way that it was possible to inquire the substrate consumption rates at the maximum specific growth rate for each of those substrate sources.

During this dissertation, fundamental key aspects needed for a high quality genome-scale metabolic network reconstruction were individually addressed and a confidence score was assigned to each reaction in order to facilitate the interpretation of the accuracy of the model during future experiments. Although the predicted specific growth rate for *Polaromonas* sp. strain JS666 was a sub optimal value comparing to the experimentally reported one, the deviousness was find to be manly dependent on the energy state of the system and FVA identified various chokepoints reactions related to the ATP production which when fully refined could give rise to a more accurated model since those reactions dictate the predicted growth rate.

Several other gaps of information, such as fatty acid pool, cofactor biosynthesis pathways and glutathione composition in biomass, were identified for future scientific research. The model works as a preliminary checkpoint for cyclic rounds of improvement in which genes will be accurately annotated to account for experimentally observed metabolic reactions (i.e. the test of the different predicted pathways), and contrariwise, metabolic processes predicted by genomic annotations could be experimentally validated. All the information integration in the iJS666 will allow that further improvements may easily be introduced by correcting preexistent information or by adding new data.

Finally, some biostimulant compounds were predicted using the iJS666 model and their predicted improvement values on the bioaugmentation were within reasonable magnitude. In the future, tests using these biostimulants could be accomplished simultaneously for confirming the improvement of the specific growth rate of Polaromonas sp. strain JS666 grown in cDCE and further validation of the iJS666 model.

## 4.1. Further Research

As referred on the previous chapter, some of the parameters inputed on the iJS666 model should be experimentally determined in order to refine the iJS666 model. Therefore, a complete list of the needed experiments and techniques that should be used in order to accomplish that final curation are displayed in the Table 4.1.

**Table 4.1.** Future relevant experiments to improve iJS666 model.

| Further Research | Method | References |
|---|---|---|
| Determine the NGAM and GAM values | Measure of 3 different carbon sources along growth, limiting the specific growth rate in the model and solving the system to obtain NGAM. | Risso et al, 2009 |
| Determine the secondary products during exponential growth | Mass spectrometry quantification | - |
| Determine the biomass composition experimentally | Mass spectometry quantification | - |
| Determine the glutathione pool | Assay for quantitative determination of glutathione and glutathione disulfide levels using enzymatic recycling method | Rahan, 2007 |
| Determine the cDCE/chloride Transport System | Crystallography and kinetic assays | - |
| Obtain Rs and Ro for the different carbon sources | Mass spectometry quantification | - |
| Test the intracellular glyoxylate reactions | Microarrays<br>Gene knock-out | Jennings et al., 2009 |
| Test the different putative biostimulants | Growth culture experiments | - |

# 6. References

Alexander, A. K. (2010). *Bioremediation and biocatalysis with Polaromonas sp. Strain JS666*. University of Iowa.

Alvarez, P., & Illman, W. (2006). Environmental contamination by hazardous substances: magnitude of the contamination problem. In *Bioremediation and natural attenuation: Process fundamentals and mathematical models.* (1st ed., p. 4). Hoboken, New Jersey: John Wiley & Sons, Inc. Retrieved from http://link.springer.com/chapter/10.1007/978-1-4614-5764-0_5\nhttp://link.springer.com/content/pdf/10.1007/978-1-4614-5764-0_5.pdf

Alvarez-Cohen, L., & Speitel, G. E. (2001). Kinetics of aerobic cometabolism of chlorinated solvents. *Biodegradation*, *12*(2), 105–126. http://doi.org/10.1023/A:1012075322466

Bradley, P. M. (2003). History and ecology of chlororethene biodegradation: a review. *Bioremediation Journal*, *7*(2), 81–109. http://doi.org/10.1080/10889860390246141

Bradley, P. M., & Chapelle, F. H. (2000). Aerobic microbial mineralization of dichloroethene as sole carbon substrate. *Environmental Science and Technology*, *34*(1), 221–223. http://doi.org/10.1021/es990785c

Branco, C. (2007). *Estudo da contaminação do aquífero superior na região de Estarreja*. Universidade de Coimbra.

Caspi, R., Foerster, H., Fulcher, C. a, Hopkinson, R., Ingraham, J., Kaipa, P., … Karp, P. D. (2006). MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Research*, *34*(Database issue), D511–D516. http://doi.org/10.1093/nar/gkj128

Christ, J. a., Ramsburg, C. A., Abriola, L. M., Pennell, K. D., & Löffler, F. E. (2005). Coupling aggressive mass removal with microbial reductive dechlorination for remediation of DNAPL source zones: A review and assessment. *Environmental Health Perspectives*, *113*(4), 465–477. http://doi.org/10.1289/ehp.6932

Coleman, N. V, Mattes, T. E., & Gossett, J. M. (2002a). Biodegradation of cis -Dichloroethene as the Sole Carbon Source by a β-Proteobacterium. *APPLIED AND ENVIRONMENTAL MICROBIOLOGY*, *68*(6), 2726–2730. http://doi.org/10.1128/AEM.68.6.2726

Coleman, N. V., Mattes, T. E., Gossett, J. M., & Spain, J. C. (2002b). Phylogenetic and kinetic diversity of aerobic vinyl chloride-assimilating bacteria from contaminated sites. *Applied and Environmental Microbiology*, *68*(12), 6162–6171. http://doi.org/10.1128/AEM.68.12.6162-6171.2002

Cox, E. (2012). Elucidation of the Mechanisms and Environmental Relevance of cis-Dichloroethene and Vinyl Chloride Biodegradation, (November). Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA581957

Darcy, J. L., Lynch, R. C., King, A. J., Robeson, M. S., & Schmidt, S. K. (2011). Global distribution of Polaromonas phylotypes - evidence for a highly successful dispersal capacity. *PLoS ONE*, *6*(8). http://doi.org/10.1371/journal.pone.0023742

De Lorenzo, V. (2008). Systems biology approaches to bioremediation. *Current Opinion in Biotechnology*, *19*(6), 579–589. http://doi.org/10.1016/j.copbio.2008.10.004

De Wildeman, S., & Verstraete, W. (2003). The quest for microbial reductive dechlorination of C (2) to C (4) chloroalkanes is warranted. *Applied Microbiology and Biotechnology*, *61*(2), 94–102. http://doi.org/10.1007/s00253-002-1174-6

DiStefano, T. D., Gossett, J. M., & Zinder, S. H. (1991). Reductive dechlorination of high concentrations of tetrachloroethene to ethene by an anaerobic enrichment culture in the absence of methanogenesis. *Applied and Environmental Microbiology*, *57*(8), 2287–2292.

Durot, M., Le Fèvre, F., de Berardinis, V., Kreimeyer, A., Vallenet, D., Combe, C., … Schachter, V. (2008). Iterative reconstruction of a global metabolic model of Acinetobacter baylyi ADP1 using high-throughput growth phenotype and gene essentiality data. *BMC Systems Biology*, *2*, 85. http://doi.org/10.1186/1752-0509-2-85

Dworkin, M., Harder, W., & Tru, H. (2001). Enrichment, Cultivation, and Detection of Reductively Dechlorinating Bacteria. *Methods in Enzimology*, *397*(1996), 1018–1020. http://doi.org/10.1016/S0076-6879(05)97005-5

Esteve-Núñez, A., Rothermich, M., Sharma, M., & Lovley, D. (2005). Growth of Geobacter sulfurreducens under nutrient-limiting conditions in continuous culture. *Environmental Microbiology*, *7*(5), 641–648. http://doi.org/10.1111/j.1462-2920.2005.00731.x

Feist, A. M., Henry, C. S., Reed, J. L., Krummenacker, M., Joyce, A. R., Karp, P. D., … Palsson, B. Ø. (2007). A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Molecular Systems Biology*, *3*(121), 121. http://doi.org/10.1038/msb4100155

Giddings, C. G. S., Jennings, L. K., & Gossett, J. M. (2010a). Microcosm Assessment of a DNA Probe Applied to Aerobic Degradation of cis-1,2-Dichloroethene by Polaromonas sp. Strain JS666. *Ground Water Monitoring & Remediation*, *30*(2), 97–105. http://doi.org/10.1111/j.1745-6592.2010.01280.x

Giddings, C. G. S., Liu, F., & Gossett, J. M. (2010b). Microcosm Assessment of Polaromonas sp. JS666 as a Bioaugmentation Agent for Degradation of cis-1,2-dichloroethene in Aerobic, Subsurface Environments. *Ground Water Monitoring & Remediation*, *30*(2), 106–113. http://doi.org/10.1111/j.1745-6592.2010.01283.x

Green, M. L., & Karp, P. D. (2006). The outcomes of pathway database computations depend on pathway ontology. *Nucleic Acids Research*, *34*(13), 3687–3697. http://doi.org/10.1093/nar/gkl438

Gudmundsson, S., & Thiele, I. (2010). Computationally efficient flux variability analysis. *BMC Bioinformatics*, *11*(1), 489. http://doi.org/10.1186/1471-2105-11-489

Hamilton, J. J., & Reed, J. L. (2014). Software platforms to facilitate reconstructing genome-scale metabolic networks. *Environmental Microbiology*, *16*(1), 49–59. http://doi.org/10.1111/1462-2920.12312

Hartmans, S., Bont, J. A. M. de, Tramper, J., & Luyben, K. C. A. M. (1985). Bacterial degradation of Vinyl Chloride. *Biotechnologg Letters*, *7*(6), 383–388.

Hopkins, G. D., & McCarty, P. L. (1995). Field Evaluation of in Situ Aerobic Cometabolism of Trichloroethylene and Three Dichloroethylene Isomers Using Phenol and Toluene as the Primary Substrates. *Environmental Science & Technology*, *29*(6), 1628–1637. http://doi.org/10.1021/es00006a029

Irgens, R. L., Gosink, J. J., & Staley, J. T. (1996). Polaromonas vacuolata gen. nov., sp. nov., a psychrophilic, marine, gas vacuolate bacterium from Antarctica. *International Journal of Systematic Bacteriology*, *46*(3), 822–826. http://doi.org/10.1099/00207713-46-3-822

Jennings, L. K. (2008). *Proteomic Analysis for the Determination of Biodegradation Pathways in Polaromonas sp . JS666*.

Jennings, L. K., Chartrand, M. M. G., Lacrampe-Couloume, G., Lollar, B. S., Spain, J. C., & Gossett, J. M. (2009). Proteomic and transcriptomic analyses reveal genes upregulated by cis-Dichloroethene in Polaromonas sp. strain JS666. *Applied and Environmental Microbiology*, *75*(11), 3733–3744. http://doi.org/10.1128/AEM.00031-09

Jeon, C. O., Park, M., Ro, H., Park, W., & Madsen, E. L. (2006). The naphthalene catabolic (*nag*) genes of *Polaromonas naphthalenivorans* CJ2: evolutionary implications for two gene clusters and novel regulatory control. *Applied and Environmental Microbiology*, *72*(2), 1086–1095. http://doi.org/10.1128/AEM.72.2.1086

Kämpfer, P., Busse, H. J., & Falsen, E. (2006). Polaromonas aquatica sp. nov., isolated from tap water. *International Journal of Systematic and Evolutionary Microbiology*, *56*(3), 605–608. http://doi.org/10.1099/ijs.0.63963-0

Karp, P. D., Latendresse, M., & Caspi, R. (2011). The Pathway Tools Pathway Prediction Algorithm. *Standards in Genomic Sciences*, *5*(3), 424–429. http://doi.org/10.4056/sigs.1794338

Karp, P. D., Paley, S. M., Krummenacker, M., Latendresse, M., Dale, J. M., Lee, T. J., … Caspi, R. (2009). Pathway Tools version 13.0: Integrated software for pathway/genome informatics and systems biology. *Briefings in Bioinformatics*, *11*(1), 40–79. http://doi.org/10.1093/bib/bbp043

Kielhorn, J., Melber, C., Wahnschaffe, U., Aitio, a, & Mangelsdorf, I. (2000). Vinyl chloride: still a cause for concern. *Environmental Health Perspectives*, *108*(7), 579–588.

Kurt, Z., Mack, E. E., & Spain, J. C. (2014). Biodegradation of cis -Dichloroethene and Vinyl Chloride in the Capillary Fringe. *Environmental Science & Technology*, *48*(22), 13350–13357. http://doi.org/10.1021/es503071m

Latendresse, M. (2014). Efficiently gap-filling reaction networks. *BMC Bioinformatics*, *15*(1), 225. http://doi.org/10.1186/1471-2105-15-225

Löffter, F., Ritalahti, K., & Zinder, S. (2013). Anaerobic Microbial Degradation of Chlorinated Ethenes. In *Bioaugmentation for Groundwater Remediation* (pp. 39–54). New York: Springer.

Loy, A., Beisker, W., & Meier, H. (2005). Diversity of Bacteria Growing in Natural Mineral Water after Bottling Diversity of Bacteria Growing in Natural Mineral Water after Bottling, *71*(7), 3624–3632. http://doi.org/10.1128/AEM.71.7.3624

Lyon, D., & Vogel, T. (2013). Development of Bioaugmentation for Groundwater Bioremediation. In *Bioaugmentation for Groundwater Remediation* (pp. 3–4). New York: Springer. http://doi.org/10.1007/978-1-4614-4115-1

Magic-Knezev, A., Wullings, B., & Van der Kooij, D. (2009). Polaromonas and Hydrogenophaga species are the predominant bacteria cultured from granular activated carbon filters in water treatment. *Journal of Applied Microbiology*, *107*(5), 1457–1467. http://doi.org/10.1111/j.1365-2672.2009.04337.x

Mahadevan, R., Bond, D. R., Butler, J. E., Coppi, V., Palsson, B. O., Schilling, C. H., & Lovley, D. R. (2006). Characterization of Metabolism in the Fe ( III ): Reducing Organism Geobacter sulfurreducens by Constraint-Based Modeling Characterization of Metabolism in the Fe ( III ) - Reducing Organism Geobacter sulfurreducens by Constraint-Based Modeling †. *Applied and Environmental Microbiology*, *72*(2), 1558–1568. http://doi.org/10.1128/AEM.72.2.1558

Mahendra, S., Petzold, C. J., Baidoo, E. E., Keasling, J. D., & Alvarez-Cohen, L. (2007). Identification of the intermediates of in vivo oxidation of 1,4-dioxane by monooxygenase-containing bacteria. *Environmental Science and Technology*, *41*(21), 7330–7336. http://doi.org/10.1021/es0705745

Margesin, R., Spröer, C., Zhang, D. C., & Busse, H. J. (2012). Polaromonas glacialis sp. nov. and Polaromonas cryoconiti sp. nov., isolated from alpine glacier cryoconite. *International Journal of Systematic and Evolutionary Microbiology*, *62*(11), 2662–2668. http://doi.org/10.1099/ijs.0.037556-0

Masip, L., Veeravalli, K., & Georgiou, G. (2006). The many faces of glutathione in bacteria. *Antioxidants & Redox Signaling*, *8*(5-6), 753–762. http://doi.org/10.1089/ars.2006.8.753

Mattes, T. E., Alexander, A. K., Richardson, P. M., Munk, a. C., Han, C. S., Stothard, P., & Coleman, N. V. (2008). The genome of Polaromonas sp. strain JS666: Insights into the evolution of a hydrocarbon- and xenobiotic-degrading bacterium, and features of relevance to biotechnology. *Applied and Environmental Microbiology*, *74*(20), 6405–6416. http://doi.org/10.1128/AEM.00197-08

McCarty, P. L., Goltz, M. N., Hopkins, G. D., Dolan, M. E., Allan, J. P., Kawakami, B. T., & Carrothers, T. J. (1998). Full-scale evaluation of in situ cometabolic degradation of trichloroethylene in groundwater through toluene injection. *Environmental Science and Technology*, *32*(1), 88–100. http://doi.org/10.1021/es970322b

Nakamura, Y. (2007). Kazusa Codon Usage Database. Acessed August 28, 2015, from http://www.kazusa.or.jp/codon/

Nishino, S. F., Shin, K. a., Gossett, J. M., & Spain, J. C. (2013). Cytochrome P450 initiates degradation of cis-dichloroethene by polaromonas sp. strain JS666. *Applied and Environmental Microbiology*, *79*(7), 2263–2272. http://doi.org/10.1128/AEM.03445-12

Nogales, J., Palsson, B. Ø., & Thiele, I. (2008). A genome-scale metabolic reconstruction of Pseudomonas putida KT2440: iJN746 as a cell factory. *BMC Systems Biology*, *2*, 79. http://doi.org/10.1186/1752-0509-2-79

Oberhardt, M. a, Palsson, B. Ø., & Papin, J. a. (2009). Applications of genome-scale metabolic reconstructions. *Molecular Systems Biology*, *5*(320), 320. http://doi.org/10.1038/msb.2009.77

Orth, J. D., Thiele, I., & Palsson, B. Ø. (2010). What is flux balance analysis? *Nature Biotechnology*, *28*(3), 245–248. http://doi.org/10.1038/nbt.1614

Osborne, T. H., Jamieson, H. E., Hudson-Edwards, K. a, Nordstrom, D. K., Walker, S. R., Ward, S. a, & Santini, J. M. (2010). Microbial oxidation of arsenite in a subarctic environment: diversity of arsenite oxidase genes and identification of a psychrotolerant arsenite oxidiser. *BMC Microbiology*, *10*, 205. http://doi.org/10.1186/1471-2180-10-205

Overbeek, R., Olson, R., Pusch, G. D., Olsen, G. J., Davis, J. J., Disz, T., … Stevens, R. (2014). The SEED and the Rapid Annotation of microbial genomes using Subsystems Technology (RAST). *Nucleic Acids Research*, *42*(D1), 206–214. http://doi.org/10.1093/nar/gkt1226

Page, K. A., Connon, S. A., & Giovannoni, S. J. (2004). Representative Freshwater Bacterioplankton Isolated from Crater Lake, Oregon. *Applied and Environmental Microbiology*, *70*(11), 6542–6550. http://doi.org/10.1128/AEM.70.11.6542-6550.2004

Palsson, B. (2009). Metabolic systems biology. *FEBS Letters*, *583*(24), 3900–3904. http://doi.org/10.1016/j.febslet.2009.09.031

Palsson, B. Ø. (2006). *Systems Biology: Properties of Reconstructed Networks*. Cambridge: Cambridge University Press.

Risso, C., Sun, J., Zhuang, K., Mahadevan, R., DeBoy, R., Ismail, W., … Methé, B. a. (2009). Genome-scale comparison and constraint-based metabolic reconstruction of the facultative anaerobic Fe(III)-reducer Rhodoferax ferrireducens. *BMC Genomics*, *10*, 447. http://doi.org/10.1186/1471-2164-10-447

Saha, R., Chowdhury, A., & Maranas, C. D. (2014). Recent advances in the reconstruction of metabolic models and integration of omics data. *Current Opinion in Biotechnology*, *29*(1), 39–45. http://doi.org/10.1016/j.copbio.2014.02.011

Scheer, M., Grote, A., Chang, A., Schomburg, I., Munaretto, C., Rother, M., … Schomburg, D. (2011). BRENDA, the enzyme information system in 2011. *Nucleic Acids Research*, *39*(SUPPL. 1), 670–676. http://doi.org/10.1093/nar/gkq1089

Schellenberger, J., Lewis, N. E., & Palsson, B. (2011). Elimination of thermodynamically infeasible loops in steady-state metabolic models. *Biophysical Journal*, *100*(3), 544–553. http://doi.org/10.1016/j.bpj.2010.12.3707

Schilling, C. H., Letscher, D., & Palsson, B. O. (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, *203*(3), 229–248. http://doi.org/10.1006/jtbi.2000.1073

Sun, J., Sayyar, B., Butler, J. E., Pharkya, P., Fahland, T. R., Famili, I., … Mahadevan, R. (2009). Genome-scale constraint-based modeling of Geobacter metallireducens. *BMC Systems Biology*, *3*, 15. http://doi.org/10.1186/1752-0509-3-15

Sun, W., Xie, S., Luo, C., & Cupples, A. M. (2010). Direct link between Toluene degradation in contaminated-site microcosms and a polaromonas strain. *Applied and Environmental Microbiology*, *76*(3), 956–959. http://doi.org/10.1128/AEM.01364-09

Taffi, M., Paoletti, N., Angione, C., Pucciarelli, S., Marini, M., & LiÃ², P. (2014). Bioremediation in marine ecosystems: a computational study combining ecological modeling and flux balance analysis. *Frontiers in Genetics*, *5*(September), 1–12. http://doi.org/10.3389/fgene.2014.00319

Thiele, I., & Palsson, B. Ø. (2010). A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature Protocols*, *5*(1), 93–121. http://doi.org/10.1038/nprot.2009.203

Usepa. (2000). Engineered approaches to in situ bioremediation of chlorinated solvents: Fundamentals and field applications, 1–144. http://doi.org/EPA 542-R-00-008

Van Hylckama Vlieg, J. E., Kingma, J., Van Den Wijngaard, a J., & Janssen, D. B. (1998). A glutathione S-transferase with activity towards cis-1, 2-dichloroepoxyethane is involved in isoprene utilization by Rhodococcus sp. strain AD45. *Applied and Environmental Microbiology*, *64*(8), 2800–2805.

Van Hylckama Vlieg, J. E. T., De Koning, W., & Janssen, D. B. (1996). Transformation kinetics of chlorinated ethenes by Methylosinus trichosporium OB3b and detection of unstable epoxides by on-line gas chromatography. *Applied and Environmental Microbiology*, *62*(9), 3304–3312.

Verce, M. F., Ulrich, R. L., & Freedman, D. L. (2000). Characterization of an isolate that uses vinyl chloride as a growth substrate under aerobic conditions. *Applied and Environmental Microbiology*, *66*(8), 3535–3542. http://doi.org/10.1128/AEM.66.8.3535-3542.2000

Verce, M. F., Ulrich, R. L., & Freedman, D. L. (2001). Transition from cometabolic to growth-linked biodegradation of vinyl chloride by a Pseudomonas sp. isolated on ethene. *Environmental Science and Technology*, *35*(21), 4242–4251. http://doi.org/10.1021/es002064f

Wetterstrand, K. (2015). DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP). Retrieved August 28, 2015, from www.genome.gov/sequencingcosts

Wittig, U., & De Beuckelaer, a. (2001). Analysis and comparison of metabolic pathway databases. *Briefings in Bioinformatics*, *2*(2), 126–142. http://doi.org/10.1093/bib/2.2.126

Zhuang, K., Izallalen, M., Mouser, P., Richter, H., Risso, C., Mahadevan, R., & Lovley, D. R. (2011). Genome-scale dynamic modeling of the competition between Rhodoferax and Geobacter in anoxic subsurface environments. *The ISME Journal*, *5*(2), 305–316. http://doi.org/10.1038/ismej.2010.117

# 7. Supplementary Data

## 7.1. Supplementary Data 1 – MATLAB functions and scripts.

***Total.m***

```
%% Retrieving data...
clear all;
clear all;
clc;
h=waitbar(0,'Retrieving data...' );
[~, ~, raw] = xlsread('\Work\RECON\Stage2\6-
22\rxnformation2.xlsx','REVIEWD','A2:V2000');
%% Cleaning non rxns lines...
waitbar(0.1,h,'Cleaning non rxns lines...');
u=[];
for i=1:length(raw);
    if ~isnan(cell2mat(raw(i,3)));
        u(length(u)+1,1)=i;
    end
end
raw=raw(u,:);
[~,idx]=sort(raw(:,1));
raw=raw(idx,:);
ind=[];
for i=1:length(raw)-1
    if ~strcmp(raw{i,1},raw{i+1,1})
       ind(length(ind)+1,1)=i;
    end
end
ind(end+1,1)=length(raw);
raw=raw(ind,:);
%% Loading other data files...
waitbar(0.2,h,'Loading other data files...');
all=raw(:,1:21);
ID1=all(:,1);
%Info from base file
file2='\Work\RECON\Stage1\2-4.PathwayTools Data\FULLREACTIONS.xlsx';
sheet2='FULLREACTIONS';
id2=['A','1',':','A','2000'];
spon2=['X','1',':','X','2000'];
[~,~,ID2]=xlsread(file2,sheet2,id2);
[~,~,SPON2]=xlsread(file2,sheet2,spon2);
%% Spontaneous check
waitbar(0.3,h,'Cheking sptnx rxns...');
 for i=1:length(ID2)
    if ~isnan(ID2{i,1})
        if strcmp(SPON2{i,1}(1,1),'T')
           for j=1:length(ID1)
               if strcmp(ID1{j,1},ID2{i,1})
                  all{j,21}='Y';
               end
           end
        end
    end
end
```

```
%% Scoring CS rxns...
waitbar(0.4,h,'Scoring CS rxns...');
%Accordingly to Palson et al;
% Evidence type  |  Confidence score  |  Examples
% Biochemical data  |  4  |  Direct evidence for gene product function and
biochemical reaction: Protein purification, biochemical assays,
experimentally solved protein structures, and comparative gene-expression
studies (e.g., Chhabra et al. 95).
% Genetic data  |  3  |  Direct and indirect evidence for gene function:
Knock-out characterization, knock-in characterization, and over-expression.
% Physiological data  |  2  |  Indirect evidence for biochemical reactions
based on physiological data: secretion products or defined medium
components serve as evidence for transport and metabolic reactions.
% Sequence data  |  2  |  Evidence for gene function: Genome annotation,
SEED annotation32.
% Modeling data  |  1  |  No evidence is available but reaction is required
for modeling. The included function is a hypothesis and needs experimental
verification. The reaction mechanism may be different from the included
reaction(s).
% Not evaluated 0
% Accordingly to Guerra et al;
% Evidence type  |  Confidence score  |  Examples
% Biochemical data  |  4  |  Direct evidence for gene product function and
% biochemical reaction: Protein purification, biochemical assays,
experimentally solved protein structures, and comparative gene-expression
studies.
% Annotation Strong Evidence  |  3  |  Direct and indirect evidence for
gene function: Predited by PathwayTools/Transport Predicter and present on
R.ferrireducans. Spontaneus rxns go here.
% Annotation Weak Evidence  |  2  |  Predicted only by
PathwayTools/Transport Predicter.
% Weak Evidence  |  2  |  No evidence is available but reaction is required
for modeling and similar rxn is present at ferroreducans.
% Modeling data  |  1  |  No evidence is available but reaction is required
for modeling. The included function is a hypothesis and needs experimental
verification. The reaction mechanism may be different from the included
reaction(s).
% Not evaluated  |  0  |  Not applied to this case since all rxns were
scored. Serves as an scoring error identifier.
%SCORE 4 will be manually introduced since the cases when happen are
very,very few.
for i=1:length(all);
    %If its not biochemical/Logical evidenced
    if isnan(all{i,5});
        %score spontaneous
        if findstr('Y',all{i,21})
        all{i,5}='3';
        %SCORE NON SPONT
        elseif ~isnan(all{i,18}) %with genes
            if ~isnan(all{i,13}) %and R ferri equivalent
            all{i,5}='3';
            elseif isnan(all{i,13}) %no R ferri equivalent
            all{i,5}='2';
            end
        elseif isnan(all{i,18}) %no GPR
            if ~isnan(all{i,13}) % with R ferri equi
            all{i,5}='2';
            elseif isnan(all{i,13})%no GPR OR equiv
            all{i,5}='1';
            end
        end
    else
        all{i,5}=num2str(all{i,5});
    end


end
%% Converting to COBRA
waitbar(0.5,h,'Extracting META & RNX independent data...');
%import independent data to work on.
rxn=all(:,3);
left=all(:,14);
right=all(:,15);
%% join metabolites
join={left{:,1},right{:,1}};
join=join';
join(any(cellfun(@(x) numel(x)==1 & isnumeric(x) && isnan(x),join),2),:) =
[];
%% rearranjments of metabolites list
x={};
for i=1:length(join);
    pos=findstr(', ',join{i,1});
       if ~isempty(pos)
            tmp = regexp(join{i,1},'\, ','split');
            x=[x;tmp'];
        else
            x((length(x)+1),1)=join(i,1);
        end
end
xx=strtrim(x);
xxx=deblank(xx);
xxxx=sort(xxx);
[~,ii]=unique(xxxx);
metabolite=xxxx(ii);
s=cellfun(@size,metabolite,'uniform',false);
[trash is]=sortrows(cat(1,s{:}),-[1 2]);
metabolites=metabolite(is);
%% Converte rxns
waitbar(0.6,h,'Directionality & reversibility correction...');
%conversor of direccionality (ATENTION: CAN NOT BE EQUAL! CONVERT &harr;
AND <==>)
a={};
for i=1:length(rxn);
    if findstr('  &harr;  ',rxn{i,1})
        a{i,1}=strrep(rxn{i,1},'  &harr;  ',' <--> ');
        elseif findstr('  &rarr;  ',rxn{i,1})
        a{i,1}=strrep(rxn{i,1},'  &rarr;  ',' --> ');
        elseif findstr('  &larr;  ',rxn{i,1})
        a{i,1}=strrep(rxn{i,1},'  &larr;  ',' <-- ');
        else
            a{i,1}=rxn{i,1};
        end
    end

all(:,3)=a(:,1);
 %% Convert direccionality para 1 e 0.
 for k=1:length(all)
     if findstr(' <--> ',all{k,3})
        all{k,4}='1';
     else
        all{k,4}='0';
     end
```

```
end

%% remove unnecessary spaces in the names of metabolites
 waitbar(0.7,h,'Metabolites deblanking, sorting and fixing');
%remove spcaes
for i=1:length(metabolites);
    if findstr('[c]',metabolites{i,1});
        metabolites{i,2}=metabolites{i,1};
    elseif findstr('[e]',metabolites{i,1})
        metabolites{i,2}=metabolites{i,1};
    else
    metabolites{i,2}=[metabolites{i,1} '[c]'];
    end
end
for i=1:length(metabolites);
    metabolites{i,2}=strrep(metabolites{i,2},' ','');
    metabolites{i,2}=strrep(metabolites{i,2},'-','');
    metabolites{i,2}=strrep(metabolites{i,2},'<','(');
    metabolites{i,2}=strrep(metabolites{i,2},'>',')');
    metabolites{i,2}=strrep(metabolites{i,2},'&','');
    metabolites{i,2}=strrep(metabolites{i,2},';','');
    metabolites{i,2}=strrep(metabolites{i,2},',','');
    metabolites{i,2}=strrep(metabolites{i,2},'[','(');
    metabolites{i,2}=strrep(metabolites{i,2},']',')');
    metabolites{i,2}=strrep(metabolites{i,2},'(c)','[c]');
    metabolites{i,2}=strrep(metabolites{i,2},'(e)','[e]');
end
 %% adds localization
 waitbar(0.9,h,'Adding localization to rxns...');
 for i=1:length(metabolites)
    oldmet=[metabolites{i,1} ' '];
    newmet=[metabolites{i,2} ' '];
    for j=1:length(all);
        all{j,3}=[all{j,3} ' '];
        all{j,3}=strrep(all{j,3},oldmet,newmet);
    end
 end
 %% adds coeficient stoich
 for i=1:length(metabolites)
    for j=1:length(all)
        pos=findstr(metabolites{i,2},all{j,3});
        if ~isempty(pos)
            if pos(1,1)==1;
                all{j,3}=['1 ' all{j,3}];
            end
            try
                all{j,3}=strrep(all{j,3},['> ' metabolites{i,2}],['> 1 '
metabolites{i,2}]);
            catch
            end
            try
                all{j,3}=strrep(all{j,3},['- ' metabolites{i,2}],['- 1 '
metabolites{i,2}]);
            catch
            end
            try
                all{j,3}=strrep(all{j,3},['+ ' metabolites{i,2}],['+ 1 '
metabolites{i,2}]);
            catch
            end
        end
    end
end
%% resort dos metabolitos
waitbar(0.95,h,'Resorting...');
 met=sort(metabolites(:,2));
ind=[];
 for i=1:length(met)-1
    if ~strcmp(met{i},met{i+1})
       ind(length(ind)+1,1)=i;
    end
end
 ind(end+1,1)=length(met);
 met=met(ind);
 s=cellfun(@size,met,'uniform',false);
[trash is]=sortrows(cat(1,s{:}),-[1 2]);
met=met(is);
%%
%clears preexistent data
Filename='G:\Work\RECON\Stage3\38\EXCELTOCOBRA\MODEL2.xls';
for SheetNum=1:2
     [N, T, Raw]=xlsread(Filename, SheetNum);
     [Raw{:, :}]=deal(NaN);
     xlswrite(Filename, Raw, SheetNum);
end
%% saving mets
waitbar(1,h,'Final arrangments and Saving...');
met(:,2)=met(:,1);
met(:,3)={'C1'};
met(:,4)=met(:,3);
met(:,5)={'0'};
for i=1:length(met)
    if findstr('[c]',met{i,2})
        met{i,6}='cytosol';
    elseif findstr('[e]',met{i,2})
        met{i,6}='extracellular';
    end
end
met(:,7:12)={'NaN'};
metheader={'Abbreviation', 'Description', 'Neutral formula', 'Charged
formula', 'Charge', 'Compartment', 'KEGG ID', 'PubChem ID', 'ChEBI ID',
'InChI string', 'SMILES', 'HMDB ID'};
meta=metheader;
for i=1:length(met)
    meta(i+1,:)=met(i,:);
end
[n,m]=size(meta);
for nn=1:n;
    for mm=1:m;
        meta{nn,mm}=deblank(meta{nn,mm});
    end
end
%%
%saves mets
xlswrite('G:\Work\RECON\Stage3\38\EXCELTOCOBRA\MODEL2.xls',meta,'Metabolite
List')
%% Creating exchange rxns.
metext={};
for i=1:length(met)
    if findstr('[e]',met{i,1})
        metext{length(metext)+1,1}=met{i,1};
    end
```

```
end
for j=1:length(metext);
exchange{j,1}=['EX_' metext{j,1}];
exchange{j,2}=['Exchange rxn for ' metext{j,1}];
exchange{j,3}=['1 ' metext{j,1} ' <--> '];
exchange{j,4}={'NaN'};
exchange{j,5}={'NaN'};
exchange{j,6}={'NaN'};
exchange{j,7}='EXCHANGE';
exchange{j,8}='1';
exchange{j,9}='-1000';
exchange{j,10}='1000';
exchange{j,11}='0';
exchange{j,12}='3';
exchange{j,13}={'NaN'};
exchange{j,14}={'NaN'};
exchange{j,15}={'NaN'};

end

%% saving rxns

rxn(:,1:3)=all(:,1:3);
rxn(:,4)=all(:,18);
rxn(:,5)=rxn(:,4);
rxn(:,6)={'NaN'};
rxn(:,7)=all(:,6);
rxn(:,8)=all(:,4);
for j=1:length(rxn);
    if findstr(' <--> ',rxn{j,3})
        rxn{j,9}='-1000'; %LB
        rxn{j,10}='1000'; %UB
    elseif findstr(' --> ',rxn{j,3})
        rxn{j,9}='0'; %LB
        rxn{j,10}='1000'; %UB
    else
        rxn{j,9}='-1000'; %LB
        rxn{j,10}='0'; %UB
    end
end
rxn(:,11)={'0'};
rxn(:,12)=all(:,5);
rxn(:,13)=all(:,8);
rxn(:,14)={'NaN'};
rxn(:,15)={'NaN'};
rxnheader={'Abbreviation', 'Description', 'Reaction', 'GPR', 'Genes',
'Proteins', 'Subsystem', 'Reversible', 'Lower bound', 'Upper bound',
'Objective', 'Confidence Score', 'EC Number', 'Notes', 'References'};
rxns=rxnheader;
for i=1:length(rxn)
    rxns(i+1,:)=rxn(i,:);
end
% adds exchange rxns
rxns=[rxns;exchange];
%deblanks;
[n,m]=size(rxns);
for nn=1:n;
    for mm=1:m;
        rxns{nn,mm}=deblank(rxns{nn,mm}(1:end));
        if isempty(rxns{nn,mm})
           rxns{nn,mm}={'NaN'};
        end
    end
end
%%
xlswrite('G:\Work\RECON\Stage3\38\EXCELTOCOBRA\MODEL2.xls',rxns,'Reaction
List')
close(h);
%%
clearvars -except rxns meta exchange;
% xls2model Writes a model from Excel spreadsheet.
%
% model = xls2model(fileName,biomassRxnEquation)
%
% INPUT
% fileName      xls spreadsheet, with one 'Reaction List' and one
'Metabolite List' tab
%
% 'Reaction List' tab: Required headers (case sensitive):
%   'Abbreviation'      HEX1
%   'Description'       Hexokinase
%   'Reaction'          1 atp[c] + 1 glc-D[c] --> 1 adp[c] + 1 g6p[c] + 1
h[c]
%   'GPR'               (3098.3) or (80201.1) or (2645.3) or ...
%   'Genes'             2645.1,2645.2,2645.3,...  (optional)
%   'Proteins'          Flj22761.1, Hk1.3, Gck.2,...  (optional)
%   'Subsystem'         Glycolysis
%   'Reversible'        0 (false) or 1 (true)
%   'Lower bound'       0
%   'Upper bound'       1000
%   'Objective'         0   (optional)
%   'Confidence Score'  0,1,2,3,4
%   'EC Number'         2.7.1.1,2.7.1.2
%   'Notes'             'Reaction also associated with EC 2.7.1.2'
(optional)
%   'References'        PMID:2043117,PMID:7150652,...   (optional)
%
% 'Metabolite List' tab: Required headers (case sensitive): (needs to be
complete list of metabolites, i.e., if a metabolite appears in multiple
compartments it has to be represented in multiple rows. Abbreviations need
to overlap with use in Reaction List
%   'Abbreviation'      glc-D or glc-D[c]
%   'Description'       D-glucose
%   'Neutral formula'   C6H12O6
%   'Charged formula'   C6H12O6
%   'Charge'            0
%   'Compartment'       cytosol
%   'KEGG ID'           C00031
%   'PubChem ID'        5793
%   'ChEBI ID'          4167
%   'InChI string'      InChI=1/C6H12O6/c7-1-2-3(8)4(9)5(10)6(11)12-2/h2-
11H,1H2/t2-,3-,4+,5-,6?/m1/s1
%   'SMILES'            OC[C@H]1OC(O)[C@H](O)[C@@H](O)[C@@H]1O
%   'HMDB ID'           HMDB00122
%
% OPTIONAL INPUT (may be required for input on unix macines)
% biomassRxnEquation        .xls may have a 255 character limit on each
cell,
%                           so pass the biomass reaction separately if it
hits this maximum.
%
% OUTPUT
% model         COBRA Toolbox model

%% loading data

% assumes that one has an xls file with two tabs
fileName='G:\Work\RECON\Stage3\38\EXCELTOCOBRA\MODEL2.xls';
[~, Strings, rxnInfo] = xlsread(fileName,'Reaction List');
[~, MetStrings, metInfo] = xlsread(fileName,'Metabolite List');
rxnInfo = rxnInfo(1:size(Strings,1),:);
metInfo = metInfo(1:size(MetStrings,1),:);
rxnHeaders = rxnInfo(1,:);
% Assuming first row is header row
rxnAbrList =
Strings(2:end,strmatch('Abbreviation',rxnHeaders,'exact'));%correct loading
of abreviation of reactions
rxnNameList =
Strings(2:end,strmatch('Description',rxnHeaders,'exact'));%correct loading
of name of reactions
rxnList = Strings(2:end,strmatch('Reaction',rxnHeaders,'exact'));%correct
loading of formula of reactions
grRuleList = Strings(2:end,strmatch('GPR',rxnHeaders,'exact'));%correct
loading of associated genes
Protein = Strings(2:end,strmatch('Proteins',rxnHeaders,'exact'));%correct
loading of protein name
subSystemList =
Strings(2:end,strmatch('Subsystem',rxnHeaders,'exact'));%correct loading of
subsystem
if ~isempty(strmatch('Reversible',rxnHeaders,'exact'))
    revFlagList =
cell2mat(rxnInfo(2:end,strmatch('Reversible',rxnHeaders,'exact')));%Matrix
else
    revFlagList = []; %if is empty
end
if ~isempty(strmatch('Lower bound',rxnHeaders,'exact'))
    lowerBoundList = cell2mat(rxnInfo(2:end,strmatch('Lower
bound',rxnHeaders,'exact')));%Matrix
else
    lowerBoundList = 1000*ones(length(rxnAbrList),1);%if is empty
end
if ~isempty(strmatch('Upper bound',rxnHeaders,'exact'))
    upperBoundList = cell2mat(rxnInfo(2:end,strmatch('Upper
bound',rxnHeaders,'exact')));%Matrix
else
    upperBoundList = 1000*ones(length(rxnAbrList),1);%if is empty
end
if ~isempty(strmatch('Objective',rxnHeaders,'exact'))
    Objective =
cell2mat(rxnInfo(2:end,strmatch('Objective',rxnHeaders,'exact')));% matrix
else
    Objective = zeros(length(rxnAbrList),1);% if is empty
end
%%
%createModel Create a COBRA model from inputs or an empty model
%structure if no inputs are provided.
%
% model = createModel(rxnAbrList,rxnNameList,rxnList,revFlagList,...
%
lowerBoundList,upperBoundList,subSystemList,grRuleList,geneNameList,...
%    systNameList)
%
%INPUTS
```

```
% rxnAbrList            List of names of the new reactions
% rxnNameList           List of names of the new reactions
% rxnList               List of reactions: format: {'A -> B + 2 C'}
%                       If the compartment of a metabolite is not
%                       specified, it is assumed to be cytoplasmic, i.e.
[c]
%
%OPTIONAL INPUTS
% revFlagList           List of reversibility flag (opt, default = 1)
% lowerBoundList        List of lower bound (Default = 0 or -vMax)
% upperBoundList        List of upper bound (Default = vMax)
% subSystemList         List of subsystem (Default = '')
% grRuleList            List of gene-reaction rule in boolean format
(and/or allowed)
%                       (Default = '');
% geneNameList          List of gene names (used only for translation
%                       from common gene names to systematic gene names)
% systNameList          List of systematic names
%
%OUTPUT
% model                 COBRA model structure
%
%create blank model
model = struct();
model.mets=cell(0,1);model.metNames=cell(0,1);model.metFormulas=cell(0,1);
model.rxns=cell(0,1);model.rxnNames=cell(0,1);model.subSystems=cell(0,1);
model.lb=zeros(0,1);model.ub=zeros(0,1);model.rev=zeros(0,1);
model.c=zeros(0,1);model.b=zeros(0,1);
model.S=sparse(0,0);
model.rxnGeneMat=sparse(0,0);
model.rules=cell(0,1);
model.grRules=cell(0,1);
model.genes=cell(0,1);
lbGivenFlag = true; %reversibility implied by lower bound
revGivenFlag = true; %reversibility implied by revFlag
nRxns = length(rxnNameList); %número de rxns
geneNameList=grRuleList; %assumes GPR genes
systNameList(1:nRxns,1) = {''}; %empty vector for sys names
kk=waitbar(0,'Starting Loading');
for i = 1 : nRxns
    waitbar(i/nRxns,kk)
    if i==nRxns
        pause(eps)
    end
    % decomposoes GPRs
    if ~isempty(grRuleList{i});
        if ~isempty(strfind(grRuleList{i},','))
          grRuleList{i}=(regexprep(grRuleList{i},',',' or '));
        end
        if ~isempty(strfind(grRuleList{i},'&'))
           grRuleList{i}=(regexprep(grRuleList{i},'&',' and '));
        end
       if ~isempty(strfind(grRuleList{i},'+'))
          grRuleList{i}=(regexprep(grRuleList{i},'+',' and '));
       end
    end
    %FORMULA DECOMPOSITION!!!!!!!!!!!!
    [metaboliteList,stoichCoeffList] = parseRxnFormula2(rxnList{i});
    %adds reaction to model
    [model,~,metasxx] =
addReaction2(model,{rxnAbrList{i},rxnNameList{i}},metaboliteList,stoichCoef
fList,revFlagList(i),lowerBoundList(i),upperBoundList(i),0,subSystemList{i}
,grRuleList{i},geneNameList,systNameList,false);
end
close(kk);
%%
if ~isempty(strmatch('Confidence Score',rxnHeaders,'exact'))
    model.confidenceScores = rxnInfo(2:end,strmatch('Confidence
Score',rxnHeaders,'exact'));
else
    model.confidenceScores = cell(length(model.rxns),1); %empty cell
instead of NaN
end
if ~isempty(strmatch('EC Number',rxnHeaders,'exact'))
    model.rxnECNumbers = Strings(2:end,strmatch('EC
Number',rxnHeaders,'exact'));
end
if ~isempty(strmatch('Notes',rxnHeaders,'exact'))
    model.rxnNotes = Strings(2:end,strmatch('Notes',rxnHeaders,'exact'));
end
if ~isempty(strmatch('References',rxnHeaders,'exact'))
    model.rxnReferences =
Strings(2:end,strmatch('References',rxnHeaders,'exact'));
end
%% fill in opt info for rxns
if ~isempty(Objective) && length(Objective) == length(model.rxns)
    model.c = (Objective);
end
model.proteins = Protein;
metHeaders = metInfo(1,:);
for n = 1:length(metHeaders)
    if isnan(metHeaders{n})
        metHeaders{n} = '';
    end
end
%% case 1: all metabolites in List have a compartment assignement
metCol = strmatch('Abbreviation',metHeaders,'exact');
    for i = 2 : length(MetStrings(:,metCol))% assumes that first row is
header
        % finds metabolites in model structure
        MetLoc =  strmatch(MetStrings{i,metCol},model.mets,'exact');
        if ~isempty(MetLoc)
            model.metNames{MetLoc} =
MetStrings{i,strmatch('Description',metHeaders,'exact')};
            model.metFormulasNeutral{MetLoc} =
MetStrings{i,strmatch('Neutral formula',metHeaders,'exact')};
            model.metFormulas{MetLoc} = MetStrings{i,strmatch('Charged
formula',metHeaders,'exact')};
            model.metCompartment{MetLoc} =
metInfo{i,strmatch('Compartment',metHeaders,'exact')};

            model.metCharges(MetLoc) =
metInfo{i,strmatch('Charge',metHeaders,'exact')};
            model.metKEGGID{MetLoc} = MetStrings{i,strmatch('KEGG
ID',metHeaders,'exact')};
            model.metInChIString{MetLoc} = MetStrings{i,strmatch('InChI
string',metHeaders,'exact')};
            model.metHMDBID{MetLoc} = MetStrings{i,strmatch('HMDB
ID',metHeaders,'exact')};
            model.metSmiles{MetLoc} =
MetStrings{i,strmatch('SMILES',metHeaders,'exact')};
```

```
            model.metPubChemID{MetLoc} = MetStrings{i,strmatch('PubChem
ID',metHeaders,'exact')};
            model.metChEBIID{MetLoc} = MetStrings{i,strmatch('ChEBI
ID',metHeaders,'exact')};
        else
            warning(['Metabolite ' metInfo{i,metCol} ' not in model']);
        end
        MetLoc=[];
    end
%% Verify all vectors are column Vectors
model.lb = columnVector(model.lb);
model.ub = columnVector(model.ub);
model.rev = columnVector(model.rev);
model.c = columnVector(model.c);
model.b = columnVector(model.b);
model.rxns = columnVector(model.rxns);
model.rxnNames = columnVector(model.rxnNames);
model.mets = columnVector(model.mets);
model.metNames = columnVector(model.metNames);
model.metFormulas = columnVector(model.metFormulas);
model.metCharges = columnVector(model.metCharges); % all others have plural
for vector
model.metFormulasNeutral = columnVector(model.metFormulasNeutral);
model.subSystems = columnVector(model.subSystems);
model.rules = columnVector(model.rules);
model.grRules = columnVector(model.grRules);
model.genes = columnVector(model.genes);
model.confidenceScores = columnVector(model.confidenceScores);
model.rxnECNumbers = columnVector(model.rxnECNumbers);
model.rxnNotes = columnVector(model.rxnNotes);
model.rxnReferences = columnVector(model.rxnReferences);
model.proteins = columnVector(model.proteins);
model.metPubChemID = columnVector(model.metPubChemID);
model.metChEBIID = columnVector(model.metChEBIID);
%%
if isfield(model,'metCompartment')
    model.metCompartment = columnVector(model.metCompartment);
end
if isfield(model,'metKEGGID')
    model.metKEGGID = columnVector(model.metKEGGID);
end
if isfield(model,'metInChIString')
    model.metInChIString = columnVector(model.metInChIString);
end
if isfield(model,'metSmiles')
    model.metSmiles = columnVector(model.metSmiles);
end
if isfield(model,'metHMDBID')
    model.metHMDBID = columnVector(model.metHMDBID);
end
```

***GapFind.m***

```
function [allGaps,rootGaps,downstreamGaps] =
gapFind(model,findNCgaps,verbFlag)
%gapFind Identifies all blocked metabolites (anything downstream of a gap)
%in a model.  MILP algorithm that finds gaps that may be missed by simple
%inspection of the S matrix. To find every gap in a model, change the rxn
%bounds on all exchange reactions to allow uptake of every metabolite.
%% [allGaps,rootGaps,downstreamGaps] = gapFind(model,findNCgaps,verbFlag)
```

```
%%INPUT
% model               a COBRA model
%%OPTIONAL INPUTS
% findNCgaps          find no consupmption gaps as well as no production gaps
%                     (default false)
% verbFlag            verbose flag (default false)
%%OUTPUTS
% allGaps             all gaps found by GapFind
% rootGaps            all root no production (and consumption) gaps
% downstreamGaps      all downstream gaps
%% based on:
% Kumar, V. et al. BMC Bioinformatics. 2007 Jun 20;8:212.
%% solve problem:
%   max ||xnp||
%       s.t. S(i,j)*v(j) >= e*w(i,j)        S(i,j) > 0, j in IR
%            S(i,j)*v(j) <= M*w(i,j)        S(i,j) > 0, j in IR
%            S(i,j)*v(j) >= e - M(1-w(i,j)) S(i,j) ~= 0, j in R
%            S(i,j)*v(j) <= M*w(i,j)        S(i,j) ~= 0, j in R
%            ||w(i,j)|| >= xnp(i)
%            lb <= v <= ub
%            S*v >= 0
%            xnp(i) = {0,1}
%            w(i,j) = {0,1}
%% reformulated for COBRA MILP as:
%   max sum(xnp(:))
%       s.t. S*v >= 0   (or = 0 if findNCgaps = true)               (1)
%            S(i,j)*v(j) - e*w(i,j) >= 0     S(i,j) > 0, j in IR    (2)
%            S(i,j)*v(j) - M*w(i,j) <= 0     S(i,j) > 0, j in IR    (3)
%            S(i,j)*v(j) - M*w(i,j) >= e-M   S(i,j) ~= 0, j in R    (4)
%            S(i,j)*v(j) - M*w(i,j) <= 0     S(i,j) ~= 0, j in R    (5)
%            sum(w(i,:)) - xnp(i) >= 0                              (6)
%            lb <= v <= ub
%            xnp and w are binary variables, v are continuous
%
%
% Jeff Orth 7/6/09

if nargin < 2
    findNCgaps = false;
end
if nargin < 3
    verbFlag = false;
end
 M = length(model.rxns); %this was set to 100 in GAMS GapFind
implementation
N = length(model.mets);
R = model.rev ~= 0; %reversible reactions
R_index = find(R);
IR = model.rev == 0; %irreversible reactions
IR_index = find(IR);
e = 0.0001;
S = model.S;
lb = model.lb;
ub = model.ub;
 % MILPproblem
%  A      LHS matrix
%  b      RHS vector
%  c      Objective coeff vector
%  lb     Lower bound vector
%  ub     Upper bound vector
%  osense Objective sense (-1 max, +1 min)
%  csense Constraint senses, a string containting the constraint sense for
%         each row in A ('E', equality, 'G' greater than, 'L' less than).
%  vartype Variable types
%  x0      Initial solution
 % initialize MILP fields
 % get number of rows and cols for each constraint
%rows
m_c1 = N; %number of metabolites
m_c2 = length(find(S(:,IR) > 0)); %number of Sij>0 in irreverisible
reactions
m_c3 = m_c2;
m_c4 = length(find(S(:,R))); %number of Sij>0 in reversible reactions
m_c5 = m_c4;
m_c6 = N; %number of xnp (metabolites)
%columns
n_v = M; %number of reactions
n_wij_IR = m_c2;
n_wij_R = m_c4;
n_xnp = N;
 % LHS matrix A
 % constraint 1
A = [S sparse(m_c1,(n_wij_IR+n_wij_R+n_xnp))];
 % constraint 2
% create Sij IR matrix and wij IR matrix
Sij_IR = sparse(m_c2,n_v);
wij_IR = sparse(m_c6,n_wij_IR);
row = 1;
for i = 1:length(IR_index)
    rxn_index = IR_index(i);
    met_index = find(S(:,rxn_index) > 0);
    for j = 1:length(met_index)
        Sij_IR(row,rxn_index) = S(met_index(j),rxn_index);
        wij_IR(met_index(j),row) = 1;
        row = row + 1;
    end
end
 A = [A ; Sij_IR -e*speye(m_c2,n_wij_IR) sparse(m_c2,(n_wij_R+n_xnp))];
 % constraint 3
A = [A ; Sij_IR -M*speye(m_c3,n_wij_IR) sparse(m_c3,(n_wij_R+n_xnp))];
 % constraint 4
% create Sij R matrix
Sij_R = sparse(m_c4,n_v);
wij_R = sparse(m_c6,n_wij_R);
row = 1;
for i = 1:length(R_index)
    rxn_index = R_index(i);
    met_index = find(S(:,rxn_index) ~= 0);
    for j = 1:length(met_index)
        Sij_R(row,rxn_index) = S(met_index(j),rxn_index);
        wij_R(met_index(j),row) = 1;
        row = row + 1;
    end
end
 A = [A ; Sij_R sparse(m_c4,n_wij_IR) -M*speye(m_c4,n_wij_R)
sparse(m_c4,n_xnp)];
 % constraint 5
A = [A ; Sij_R sparse(m_c5,n_wij_IR) -M*speye(m_c5,n_wij_R)
sparse(m_c5,n_xnp)];
 % constraint 6
A = [A ; sparse(m_c6,n_v) wij_IR wij_R -1*speye(m_c6,n_xnp)];
 % RHS vector b
```

```
b = [zeros(m_c1+m_c2+m_c3,1);(e-M)*ones(m_c4,1);zeros(m_c5+m_c6,1)];
 % objective coefficient vector c
c = [zeros(n_v+n_wij_IR+n_wij_R,1);ones(n_xnp,1)];
 % upper and lower bounds on variables (v,w,xnp)
lb = [lb;zeros(n_wij_IR+n_wij_R+n_xnp,1)];
ub = [ub;ones(n_wij_IR+n_wij_R+n_xnp,1)];
 % objective sense osense
osense = -1; %want to maximize objective
 % constraint senses csense
if findNCgaps
    csense(1:m_c1) = 'E';
else
    csense(1:m_c1) = 'G';
end
csense((m_c1+1):(m_c1+m_c2)) = 'G';
csense((m_c1+m_c2+1):(m_c1+m_c2+m_c3)) = 'L';
csense((m_c1+m_c2+m_c3+1):(m_c1+m_c2+m_c3+m_c4)) = 'G';
csense((m_c1+m_c2+m_c3+m_c4+1):(m_c1+m_c2+m_c3+m_c4+m_c5)) = 'L';
csense((m_c1+m_c2+m_c3+m_c4+m_c5+1):(m_c1+m_c2+m_c3+m_c4+m_c5+m_c6)) = 'G';
 % variable types vartype
vartype(1:n_v) = 'C';
vartype((n_v+1):(n_v+n_wij_IR+n_wij_R+n_xnp)) = 'B';
 % inital solution x0
x0 = [];
 % run COBRA MILP solver
gapFindMILPproblem.A = A;
gapFindMILPproblem.b = b;
gapFindMILPproblem.c = c;
gapFindMILPproblem.lb = lb;
gapFindMILPproblem.ub = ub;
gapFindMILPproblem.osense = osense;
gapFindMILPproblem.csense = csense;
gapFindMILPproblem.vartype = vartype;
gapFindMILPproblem.x0 = x0;
 if verbFlag
    parameters.printLevel = 3;
else
    parameters.printLevel = 0;
end
 solution = solveCobraMILP(gapFindMILPproblem,parameters);
 % get the list of gaps from MILP solution
metsProduced =
solution.full((n_v+n_wij_IR+n_wij_R+1):(n_v+n_wij_IR+n_wij_R+n_xnp),1);
allGaps = model.mets(~metsProduced);
rootGaps = findrootgaps(model); %identify root gaps using findRootNPmets
downstreamGaps = allGaps(~ismember(allGaps,rootGaps));
```

### ***FindRootGaps.m***

```
function [rootGaps] = findrootgaps(model)
%based on findRootNPmets and findGaps
%Initially finds the root no production (NP) and no consumption(NC)
%metabolites in a model.
%PART1;
%INPUT
%model (COBRA model)
%OUTPUT
%NP (non-production root's)
%NC (non-comsuption root's)
%PART2;
```

```
%After identifying root mets, adds sink rxns for those mets to the model
%and run gapFind to reveal non-topologycal gaps (NT);
%INPUT
%newmodel (new COBRA model with sink rxns)
%OUTPUT
%NT (non-topologycal unbalances)
 % Andre Guerra 05/05/2015
 %% Finding root gaps.
% Finds metabolites that only have one entry on S matrix...meaning that
% they are root gaps (non-production, non-consumption or a single
% reversible rxn)
 S=full(model.S);
ind=find(S~=0);
S(ind)=1;
MetSum=sum(S');
rootGaps=find(MetSum==1);
rootGaps=model.mets(rootGaps);
 end
```

***Similarity.m***

```
%% - infeasibility -
% This script looks for similar reactions in the model that may cause
infeasibilities.
% André Guerra: 07 - 05 - 1015
%% Join rxns that use same metabolites.
 [l,c]=size(model.S);
a=cell([l,c]);
for i=1:l
 for j=1:c
     if model.S(i,j)~=0
         a(i,j)=model.mets(i,1);
     end
 end
end
 %% clear empty spaces with empty strings
 for i=1:l
 for j=1:c
     if isempty(a{i,j})
         a{i,j}='';
     end
 end
end
 %% Sort columns of mets
 for i=1:length(model.S)
a(:,i)=sortrows(a(:,i));
 end
 %% find columns that use NADPH and NAD
 [s,n]=size(a);
b={};
for nn=1:n
    if strmatch('NADPH[c]',a(:,nn))
    b(:,nn)=a(:,nn);
    elseif strmatch('NADH[c]',a(:,nn))
    b(:,nn)=a(:,nn);
    else
    b(:,nn)=cell(s,1);
    end
end
 %% clear empty spaces with empty strings
[n,m]=size(b);
```

```
for i=1:n
 for j=1:m
     if isempty(b{i,j})
         b{i,j}='';
     end
 end
end
 %% removing NADPH AND NADH
 [~,m]=size(b);
 for mm=1:m
    posnadh=strmatch('NADH[c]',b(:,mm));
    if ~isempty(posnadh)
    b(posnadh,mm)={''};
    end
    posnadph=strmatch('NADPH[c]',b(:,mm));
    if ~isempty(posnadph)
    b(posnadph,mm)={''};
    end
    posnadhm=strmatch('NAD(sup)+(/sup)[c]',b(:,mm));
    if ~isempty(posnadhm)
    b(posnadhm,mm)={''};
    end
    posnadphm=strmatch('NADP(sup)+(/sup)[c]',b(:,mm));
    if ~isempty(posnadphm)
    b(posnadphm,mm)={''};
    end
end
 %% Sort columns of mets
 [~,m]=size(b);
 for i=1:m
b(:,i)=sortrows(b(:,i));
 end
 %%
[s,m]=size(b);
 for ii=1:m
    if length(strmatch('',b(:,m)))~=s
        k=[];
        for aa=1:m
            if isequal(b(:,ii),b(:,iinm));
                k(length(k)+1,1)=iinm;
            end
        end
        if ~isempty(k)
            disp(['rxn' num2str(ii) 'have the following similar
reactions;'])
            disp(k)
        end
    end
end
```

***Blank.m***

```
function [blankmodel]=blank(model)
%Blankmodel by Guerra 25/05/2015
 % iJS666 was tested to grow in carbon-free
% minimal salts medium (MSM) modified from Hartmans et al.
% (1985) to contain 20 mM phosphate, 10 mM ammonium, and
% 0.02 mM chloride.
 % MSM contained per litre deionized water:
% 3.88 g K2HPO4,
% 2.13 g NaH2PO4'2H2O,
% 2.0 g (NH4)2SO4,
% 0.1 g MoCl2'6H20,
% 10 mg ethylenediaminetetraacetic acid (EDTA),
% 2 mg ZnSO4" 7Hz,
% 1 mg CaCl2" 2HzO,
% 5 mg FeSO4"7HzO,
% 0.2mg Na2MoO4"2H20,
% 0.2mg CuSO4"5H20,
% 0.4mg CoCl2-6H20
% 1 mg MnCI2"2H2O
 %NO CARBON CAN ENTER IN MEDIUM!!!
 %% Change all bounds
 a=find(model.rev==0);
b=find(model.rev==1);
blankmodel=changeRxnBounds(model,model.rxns(a),0,'l');
blankmodel=changeRxnBounds(blankmodel,model.rxns(a),1000,'u');
blankmodel=changeRxnBounds(blankmodel,model.rxns(b),-1000,'l');
blankmodel=changeRxnBounds(blankmodel,model.rxns(b),1000,'u');
  %% Getting the exchange rxns:
 indices=strmatch('EX_',model.rxns);
emedium=model.rxns(indices);



%% The exchange reactions are in the format: A[e] <-->    , that means
positive values means production and negative value means uptake.
%% Metabolites that usually are consumed by the microorganism have negative
exchange reactions associated Ex:O2,glucose, sais...
%% Uptake
 pos1=strmatch('EX_Co(sup)2+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos1),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos1),0,'u');
 pos2=strmatch('EX_Fe(sup)3+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos2),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos2),0,'u');
 pos3=strmatch('EX_Mg(sup)2+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos3),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos3),0,'u');
 pos4=strmatch('EX_Mo(sup)2+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos4),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos4),0,'u');
 pos5=strmatch('EX_Ni(sup)2+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos5),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos5),0,'u');
 pos6=strmatch('EX_Zn(sup)2+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos6),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos6),0,'u');
 pos7=strmatch('EX_Na(sup)+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos7),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos7),0,'u');
 pos8=strmatch('EX_K(sup)+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos8),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos8),0,'u');
 pos9=strmatch('EX_phosphate[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos9),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos9),0,'u');
 pos10=strmatch('EX_ammonium[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos10),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos10),0,'u');
 pos11=strmatch('EX_chloride[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos11),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos11),0,'u');
```

```
 pos12=strmatch('EX_selenate[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos12),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos12),0,'u');
 pos13=strmatch('EX_sulfate[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos13),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos13),0,'u');
 pos14=strmatch('EX_oxygen[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(pos14),-100,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(pos14),0,'u');
 inlist=[pos1 pos2 pos3 pos4 pos5 pos6 pos7 pos8 pos9 pos10 pos11 pos12
pos13 pos14];
  %% Metabolites that exit the system have positive values for the exchange
flux. Ex:CO2 e H2
 otherslist=setdiff((1:length(emedium))',inlist);
otherslistname=emedium(otherslist);
blankmodel=changeRxnBounds(blankmodel,otherslistname,0,'l');
blankmodel=changeRxnBounds(blankmodel,otherslistname,100,'u');
 %% H2O
posagua=strmatch('EX_H(sub)2(/sub)O[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(posagua),0,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(posagua),0,'u');
 %% H+
posh=strmatch('EX_H(sup)+(/sup)[e]',emedium);
blankmodel=changeRxnBounds(blankmodel,emedium(posh),0,'l');
blankmodel=changeRxnBounds(blankmodel,emedium(posh),0,'u');
 %% Disp name + bounds (just a simple display)
for i=1:length(emedium);
    if length(emedium{i,1})>19
    emedium{i,1}=emedium{i,1}(1:19);
    end
end
disp([emedium,num2cell(blankmodel.lb(indices)),num2cell(blankmodel.ub(indic
es))])
end
```

***cDCE.m* (equal to the previous *blank.m* script but with the following adaptation)**

```
%% Defining NGAM
 cDCEmodel=changeRxnBounds(cDCEmodel,'NGAM',0.45,'b');
 %% FONTE CARBONO
 pos16=strmatch('EX_cDCE[e]',emedium);
cDCEmodel=changeRxnBounds(cDCEmodel,emedium(pos16),-0.561,'l');
cDCEmodel=changeRxnBounds(cDCEmodel,emedium(pos16),0,'u');
inlist=[inlist pos16];
```

***Validationpositive.m***

```
%% validationpositive
%% Test Growth for the experimentally tested substrates tha had an positive
outcome. André Guerra - 10/07/2015

benzoatemodel=benzoate(model);
benzoateresult=optimizeCbModel(benzoatemodel,'max','one',1);
 catecholmodel=catechol(model);
catecholresult=optimizeCbModel(catecholmodel,'max','one',1);
 DCAmodel=DCA(model);
DCAresult=optimizeCbModel(DCAmodel,'max','one',1);
 glucosemodel=glucose(model);
glucoseresult=optimizeCbModel(glucosemodel,'max','one',1);
```

```
 glycolatemodel=glycolate(model);
glycolateresult=optimizeCbModel(glycolatemodel,'max','one',1);
 hydroxyquinolmodel=hydroxyquinol(model);
hydroxyquinolresult=optimizeCbModel(hydroxyquinolmodel,'max','one',1);
pos=[benzoateresult.f;catecholresult.f;DCAresult.f;glucoseresult.f;glycolat
eresult.f;hydroxyquinolresult.f];
disp(pos);
```

***Validationnegative.m***

```
%% validationnegative
%% Test Growth for the experimentally tested substrates that had a negative
outcome. André Guerra - 12/07/2015

chloroacetaldehydemodel=chloroacetaldehyde(model);
chloroacetaldehyderesult=optimizeCbModel(chloroacetaldehydemodel,'max','one
',1);
 cyclohexanemodel=cyclohexane(model);
cyclohexaneresult=optimizeCbModel(cyclohexanemodel,'max','one',1);
 cyclohexaneacetatemodel=cyclohexaneacetate(model);
cyclohexaneacetateresult=optimizeCbModel(cyclohexaneacetatemodel,'max','one
',1);
 DCPmodel=DCP(model);
DCPresult=optimizeCbModel(DCPmodel,'max','one',1);
 ethanemodel=ethane(model);
ethaneresult=optimizeCbModel(ethanemodel,'max','one',1);

ethanolnitrateanoxicmodel=ethanolnitrateanoxic(model);
ethanolnitrateanoxicresult=optimizeCbModel(ethanolnitrateanoxicmodel,'max',
'one',1);
 ethylcyclohexanemodel=ethylcyclohexane(model);
ethylcyclohexaneresult=optimizeCbModel(ethylcyclohexanemodel,'max','one',1)
;
 hexanemodel=hexane(model);
hexaneresult=optimizeCbModel(hexanemodel,'max','one',1);
 hydroxy4benzoatemodel=hydroxy4benzoate(model);
hydroxy4benzoateresult=optimizeCbModel(hydroxy4benzoatemodel,'max','one',1)
;
 naphthalenemodel=naphthalene(model);
naphthaleneresult=optimizeCbModel(naphthalenemodel,'max','one',1);
 amino2benzoatemodel=amino2benzoate(model);
amino2benzoateresult=optimizeCbModel(amino2benzoatemodel,'max','one',1);
 amino4benzoatemodel=amino4benzoate(model);
amino4benzoateresult=optimizeCbModel(amino4benzoatemodel,'max','one',1);
 phthalatemodel=phthalate(model);
phthalateresult=optimizeCbModel(phthalatemodel,'max','one',1);
 propanemodel=propane(model);
propaneresult=optimizeCbModel(propanemodel,'max','one',1);
 thiosulfatemodel=thiosulfate(model);
thiosulfateresult=optimizeCbModel(thiosulfatemodel,'max','one',1);
 thiosulfateethanolmodel=thiosulfateethanol(model);
thiosulfateethanolresult=optimizeCbModel(thiosulfateethanolmodel,'max','one
',1);
neg=[chloroacetaldehyderesult.f;cyclohexaneresult.f;cyclohexaneacetateresul
t.f;DCPresult.f;ethaneresult.f;ethanolnitrateanoxicresult.f;ethylcyclohexan
eresult.f;hexaneresult.f;hydroxy4benzoateresult.f;naphthaleneresult.f;amino
2benzoateresult.f;amino4benzoateresult.f;
phthalateresult.f;propaneresult.f;thiosulfateresult.f;thiosulfateethanolres
ult.f;]
```

***Validationvariation.m***

```
%% Validationvariation.m
%% This script was done in order to assess the influence of Rs values in
the specific growth rate of microorganism.
%% André Guerra - 28/07/2015
 %% Positive Growth (with substrate consuming rate associated)
 acetatemodel=acetate(model);
i=0;
acetatevariation=zeros(1,2);
while acetatevariation(end,2)<0.0375
    i=i+0.1
acetatemodel=changeRxnBounds(acetatemodel,'EX_acetate[e]',-(i),'l');
acetateresult=optimizeCbModel(acetatemodel,'max','one',1);
for j=length(acetatevariation)
    acetatevariation(j+1,1)=i;
if isfield(acetateresult,'f')
acetatevariation(j+1,2)=acetateresult.f;
else
acetatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,1), plot(acetatevariation(:,1),acetatevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...

title('Acetate');
 %%
 cDCEmodel=cDCE(model);
i=0;
cDCEvariation=zeros(1,2);
while cDCEvariation(end,2)<0.0089
     i=i+0.01;
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_cDCE[e]',-(i),'l');
cDCEresult=optimizeCbModel(cDCEmodel,'max','one',1);
for j=length(cDCEvariation)
    cDCEvariation(j+1,1)=i;
if isfield(cDCEresult,'f')
cDCEvariation(j+1,2)=cDCEresult.f;
else
cDCEvariation(j+1,2)=0;
end
end
end
 subplot(3,5,2), plot(cDCEvariation(:,1),cDCEvariation(:,2)), xlabel('Rs -
mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('cDCE');
 %%
 chloroacetatemodel=chloroacetate(model);
i=0;
chloroacetatevariation=zeros(1,2);
while chloroacetatevariation(end,2)<0.0018
     i=i+0.01
chloroacetatemodel=changeRxnBounds(chloroacetatemodel,'EX_chloroacetate[e]'
,-(i),'l');
chloroacetateresult=optimizeCbModel(chloroacetatemodel,'max','one',1);
for j=length(chloroacetatevariation)
    chloroacetatevariation(j+1,1)=i;
if isfield(chloroacetateresult,'f')
chloroacetatevariation(j+1,2)=chloroacetateresult.f;
```

```
else
chloroacetatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,3),
plot(chloroacetatevariation(:,1),chloroacetatevariation(:,2)), xlabel('Rs -
mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Chloroacetate');
 %%
 cyclohexanecarboxylatemodel=cyclohexanecarboxylate(model);
i=0;
cyclohexanecarboxylatevariation=zeros(1,2);
while cyclohexanecarboxylatevariation(end,2)<0.0190
     i=i+0.01;
cyclohexanecarboxylatemodel=changeRxnBounds(cyclohexanecarboxylatemodel,'EX
_cyclohexane1carboxylate[e]',-(i),'l');
cyclohexanecarboxylateresult=optimizeCbModel(cyclohexanecarboxylatemodel,'m
ax','one',1);
for j=length(cyclohexanecarboxylatevariation)
    cyclohexanecarboxylatevariation(j+1,1)=i;
if isfield(cyclohexanecarboxylateresult,'f')
cyclohexanecarboxylatevariation(j+1,2)=cyclohexanecarboxylateresult.f;
else
cyclohexanecarboxylatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,4),
plot(cyclohexanecarboxylatevariation(:,1),cyclohexanecarboxylatevariation(:
,2)), xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate -
1/(h)'),...
title('Cyclohexanecarboxylate');
 %%
 cyclohexanolmodel=cyclohexanol(model);
i=0;
cyclohexanolvariation=zeros(1,2);
while cyclohexanolvariation(end,2)<0.0097
     i=i+0.01;
cyclohexanolmodel=changeRxnBounds(cyclohexanolmodel,'EX_cyclohexanol[e]',-
(i),'l');
cyclohexanolresult=optimizeCbModel(cyclohexanolmodel,'max','one',1);
for j=length(cyclohexanolvariation)
    cyclohexanolvariation(j+1,1)=i;
if isfield(cyclohexanolresult,'f')
cyclohexanolvariation(j+1,2)=cyclohexanolresult.f;
else
cyclohexanolvariation(j+1,2)=0;
end
end
end
 subplot(3,5,5),
plot(cyclohexanolvariation(:,1),cyclohexanolvariation(:,2)), xlabel('Rs -
mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Cyclohexanol');
%%
 ethanolmodel=ethanol(model);
i=0;
ethanolvariation=zeros(1,2);
while ethanolvariation(end,2)<0.0090
     i=i+0.01;
```

```
ethanolmodel=changeRxnBounds(ethanolmodel,'EX_ethanol[e]',-(i),'l');
ethanolresult=optimizeCbModel(ethanolmodel,'max','one',1);
for j=length(ethanolvariation)
    ethanolvariation(j+1,1)=i;
if isfield(ethanolresult,'f')
ethanolvariation(j+1,2)=ethanolresult.f;
else
ethanolvariation(j+1,2)=0;
end
end
end
 subplot(3,5,6), plot(ethanolvariation(:,1),ethanolvariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Ethanol');
 %%
ferulatemodel=ferulate(model);
i=0;
ferulatevariation=zeros(1,2);
while ferulatevariation(end,2)<0.0035
     i=i+0.01;
ferulatemodel=changeRxnBounds(ferulatemodel,'EX_ferulate[e]',-(i),'l');
ferulateresult=optimizeCbModel(ferulatemodel,'max','one',1);
for j=length(ferulatevariation)
    ferulatevariation(j+1,1)=i;
if isfield(ferulateresult,'f')
ferulatevariation(j+1,2)=ferulateresult.f;
else
ferulatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,7), plot(ferulatevariation(:,1),ferulatevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Ferulate');
 %%
 gentisatemodel=gentisate(model);
i=0;
gentisatevariation=zeros(1,2);
while gentisatevariation(end,2)<0.0079
     i=i+0.01;
gentisatemodel=changeRxnBounds(gentisatemodel,'EX_gentisate[e]',-(i),'l');
gentisateresult=optimizeCbModel(gentisatemodel,'max','one',1);
for j=length(gentisatevariation)
    gentisatevariation(j+1,1)=i;
if isfield(gentisateresult,'f')
gentisatevariation(j+1,2)=gentisateresult.f;
else
gentisatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,8), plot(gentisatevariation(:,1),gentisatevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Gentisate');
 %%
 heptanemodel=heptane(model);
i=0;
heptanevariation=zeros(1,2);
while heptanevariation(end,2)<0.0047
     i=i+0.01;
heptanemodel=changeRxnBounds(heptanemodel,'EX_nheptane[e]',-(i),'l');
```

```
heptaneresult=optimizeCbModel(heptanemodel,'max','one',1);
for j=length(heptanevariation)
    heptanevariation(j+1,1)=i;
if isfield(heptaneresult,'f')
heptanevariation(j+1,2)=heptaneresult.f;
else
heptanevariation(j+1,2)=0;
end
end
end
 subplot(3,5,9), plot(heptanevariation(:,1),heptanevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Heptane');
 %%
 hydroxybenzoatemodel=hydroxybenzoate(model);
i=0;
hydroxybenzoatevariation=zeros(1,2);
while hydroxybenzoatevariation(end,2)<0.0080
     i=i+0.01;
hydroxybenzoatemodel=changeRxnBounds(hydroxybenzoatemodel,'EX_3hydroxybenzo
ate[e]',-(i),'l');
hydroxybenzoateresult=optimizeCbModel(hydroxybenzoatemodel,'max','one',1);
for j=length(hydroxybenzoatevariation)
    hydroxybenzoatevariation(j+1,1)=i;
if isfield(hydroxybenzoateresult,'f')
hydroxybenzoatevariation(j+1,2)=hydroxybenzoateresult.f;
else
hydroxybenzoatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,10),
plot(hydroxybenzoatevariation(:,1),hydroxybenzoatevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('3-hydroxybenzoate');
 %%
octanemodel=octane(model);
i=0;
octanevariation=zeros(1,2);
while octanevariation(end,2)<0.0025
     i=i+0.01;
octanemodel=changeRxnBounds(octanemodel,'EX_noctane[e]',-(i),'l');
octaneresult=optimizeCbModel(octanemodel,'max','one',1);
for j=length(octanevariation)
    octanevariation(j+1,1)=i;
if isfield(octaneresult,'f')
octanevariation(j+1,2)=octaneresult.f;
else
octanevariation(j+1,2)=0;
end
end
end
 subplot(3,5,11), plot(octanevariation(:,1),octanevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Octane');
 %%
protocatechuatemodel=protocatechuate(model);
i=0;
protocatechuatevariation=zeros(1,2);
while protocatechuatevariation(end,2)<0.0094
     i=i+0.01;
```

```
protocatechuatemodel=changeRxnBounds(protocatechuatemodel,'EX_protocatechua
te[e]',-(i),'l');
protocatechuateresult=optimizeCbModel(protocatechuatemodel,'max','one',1);
for j=length(protocatechuatevariation)
    protocatechuatevariation(j+1,1)=i;
if isfield(protocatechuateresult,'f')
protocatechuatevariation(j+1,2)=protocatechuateresult.f;
else
protocatechuatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,12),
plot(protocatechuatevariation(:,1),protocatechuatevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Protocatechuate');
 %%
 salicylatemodel=salicylate(model);
i=0;
salicylatevariation=zeros(1,2);
while salicylatevariation(end,2)<0.0121
     i=i+0.01;
salicylatemodel=changeRxnBounds(salicylatemodel,'EX_salicylate[e]',-
(i),'l');
salicylateresult=optimizeCbModel(salicylatemodel,'max','one',1);
for j=length(salicylatevariation)
    salicylatevariation(j+1,1)=i;
if isfield(salicylateresult,'f')
salicylatevariation(j+1,2)=salicylateresult.f;
else
salicylatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,13), plot(salicylatevariation(:,1),salicylatevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Salicylate');
 %%
 succinatemodel=succinate(model);
i=0;
succinatevariation=zeros(1,2);
while succinatevariation(end,2)<0.0231
    i=i+0.01;
succinatemodel=changeRxnBounds(succinatemodel,'EX_succinate[e]',-(i),'l');
succinateresult=optimizeCbModel(succinatemodel,'max','one',1);
for j=length(succinatevariation)
    succinatevariation(j+1,1)=i;
if isfield(succinateresult,'f')
succinatevariation(j+1,2)=succinateresult.f;
else
succinatevariation(j+1,2)=0;
end
end
end
 subplot(3,5,14), plot(succinatevariation(:,1),succinatevariation(:,2)),
xlabel('Rs - mmol/(gDW*h)'), ylabel('Specific Growth Rate - 1/(h)'),...
title('Succinate');
```

***Validationox.m***

```
%% Positive Growth (with substrate consuming rate associated) for test of
oxygen uptake
%% 30/07/2015 - André Guerra
 n=1
acetatemodel=acetate(model);
i=0;
acetatevariation=zeros(1,3);
while acetatevariation(end,2)<0.0375
    i=i+0.1;
acetatemodel=changeRxnBounds(acetatemodel,'EX_acetate[e]',-(i),'l');
acetateresult=optimizeCbModel(acetatemodel,'max','one',1);
for j=length(acetatevariation)
    acetatevariation(j+1,1)=i;
if isfield(acetateresult,'f')
acetatevariation(j+1,2)=acetateresult.f;
else
acetatevariation(j+1,2)=0;
end
 if isempty(acetateresult.x)
acetatevariation(j+1,3)=0;
else
acetatevariation(j+1,3)=-
acetateresult.x(strmatch('EX_oxygen',acetatemodel.rxns));
end
end
end
acetatevariation(all(acetatevariation==0,2),:)=[]; %removing zeros for
faster ploting.
figure;subplot(3,5,1),...
plot(acetatevariation(:,1),acetatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 acetatevariation(end,1) 0 acetatevariation(end,3)]),...
title('RO2 vs Rs in Acetate');
 %%
n=2
cDCEmodel=cDCE(model);
i=0;
cDCEvariation=zeros(1,3);
while cDCEvariation(end,2)<0.0089
     i=i+0.01;
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_cDCE[e]',-(i),'l');
cDCEresult=optimizeCbModel(cDCEmodel,'max','one',1);
for j=length(cDCEvariation)
    cDCEvariation(j+1,1)=i;
if isfield(cDCEresult,'f')
cDCEvariation(j+1,2)=cDCEresult.f;
else
cDCEvariation(j+1,2)=0;
end
if isempty(cDCEresult.x)
cDCEvariation(j+1,3)=0;
else
cDCEvariation(j+1,3)=-cDCEresult.x(strmatch('EX_oxygen',cDCEmodel.rxns));
end
end
end
 cDCEvariation(all(cDCEvariation==0,2),:)=[]; %removing zeros for faster
ploting.
subplot(3,5,2),...
plot(cDCEvariation(:,1),cDCEvariation(:,3)),...
```

```
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 cDCEvariation(end,1) 0 cDCEvariation(end,3)]),...
title('RO2 vs Rs in cDCE');
%%
n=3
chloroacetatemodel=chloroacetate(model);
i=0;
chloroacetatevariation=zeros(1,3);
while chloroacetatevariation(end,2)<0.0018
     i=i+0.01;
chloroacetatemodel=changeRxnBounds(chloroacetatemodel,'EX_chloroacetate[e]'
,-(i),'l');
chloroacetateresult=optimizeCbModel(chloroacetatemodel,'max','one',1);
for j=length(chloroacetatevariation)
    chloroacetatevariation(j+1,1)=i;
if isfield(chloroacetateresult,'f')
chloroacetatevariation(j+1,2)=chloroacetateresult.f;
else
chloroacetatevariation(j+1,2)=0;
end
if isempty(chloroacetateresult.x)
chloroacetatevariation(j+1,3)=0;
else
chloroacetatevariation(j+1,3)=-
chloroacetateresult.x(strmatch('EX_oxygen',chloroacetatemodel.rxns));
end
end
end
 chloroacetatevariation(all(chloroacetatevariation==0,2),:)=[]; %removing
zeros for faster ploting.
subplot(3,5,3),...
plot(chloroacetatevariation(:,1),chloroacetatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 chloroacetatevariation(end,1) 0 chloroacetatevariation(end,3)]),...
title('RO2 vs Rs in Chloroacetate');
 %%
n=4
cyclohexanecarboxylatemodel=cyclohexanecarboxylate(model);
i=0;
cyclohexanecarboxylatevariation=zeros(1,3);
while cyclohexanecarboxylatevariation(end,2)<0.0190
     i=i+0.01;
cyclohexanecarboxylatemodel=changeRxnBounds(cyclohexanecarboxylatemodel,'EX
_cyclohexane1carboxylate[e]',-(i),'l');
cyclohexanecarboxylateresult=optimizeCbModel(cyclohexanecarboxylatemodel,'m
ax','one',1);
for j=length(cyclohexanecarboxylatevariation)
    cyclohexanecarboxylatevariation(j+1,1)=i;
if isfield(cyclohexanecarboxylateresult,'f')
cyclohexanecarboxylatevariation(j+1,2)=cyclohexanecarboxylateresult.f;
else
cyclohexanecarboxylatevariation(j+1,2)=0;
end
if isempty(cyclohexanecarboxylateresult.x)
cyclohexanecarboxylatevariation(j+1,3)=0;
else
cyclohexanecarboxylatevariation(j+1,3)=-
cyclohexanecarboxylateresult.x(strmatch('EX_oxygen',cyclohexanecarboxylatem
odel.rxns));
```

```
end
end
end
cyclohexanecarboxylatevariation(all(cyclohexanecarboxylatevariation==0,2),:
)=[]; %removing zeros for faster ploting.
subplot(3,5,4),...
plot(cyclohexanecarboxylatevariation(:,1),cyclohexanecarboxylatevariation(:
,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 cyclohexanecarboxylatevariation(end,1) 0
cyclohexanecarboxylatevariation(end,3)]),...
title('RO2 vs Rs in Cyclohexanecarboxylate');
%%
n=5
cyclohexanolmodel=cyclohexanol(model);
i=0;
cyclohexanolvariation=zeros(1,3);
while cyclohexanolvariation(end,2)<0.0097
     i=i+0.01;
cyclohexanolmodel=changeRxnBounds(cyclohexanolmodel,'EX_cyclohexanol[e]',-
(i),'l');
cyclohexanolresult=optimizeCbModel(cyclohexanolmodel,'max','one',1);
for j=length(cyclohexanolvariation)
    cyclohexanolvariation(j+1,1)=i;
if isfield(cyclohexanolresult,'f')
cyclohexanolvariation(j+1,2)=cyclohexanolresult.f;
else
cyclohexanolvariation(j+1,2)=0;
end
if isempty(cyclohexanolresult.x)
cyclohexanolvariation(j+1,3)=0;
else
cyclohexanolvariation(j+1,3)=-
cyclohexanolresult.x(strmatch('EX_oxygen',cyclohexanolmodel.rxns));
end
end
end
 cyclohexanolvariation(all(cyclohexanolvariation==0,2),:)=[]; %removing
zeros for faster ploting.
subplot(3,5,5),...
plot(cyclohexanolvariation(:,1),cyclohexanolvariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 cyclohexanolvariation(end,1) 0 cyclohexanolvariation(end,3)]),...
title('RO2 vs Rs in Cyclohexanol');
 %%
n=6
ethanolmodel=ethanol(model);
i=0;
ethanolvariation=zeros(1,3);
while ethanolvariation(end,2)<0.0090
     i=i+0.01;
ethanolmodel=changeRxnBounds(ethanolmodel,'EX_ethanol[e]',-(i),'l');
ethanolresult=optimizeCbModel(ethanolmodel,'max','one',1);
for j=length(ethanolvariation)
    ethanolvariation(j+1,1)=i;
if isfield(ethanolresult,'f')
ethanolvariation(j+1,2)=ethanolresult.f;
else
ethanolvariation(j+1,2)=0;
```

```
end
if isempty(ethanolresult.x)
ethanolvariation(j+1,3)=0;
else
ethanolvariation(j+1,3)=-
ethanolresult.x(strmatch('EX_oxygen',ethanolmodel.rxns));
end
end
end
 ethanolvariation(all(ethanolvariation==0,2),:)=[]; %removing zeros for
faster ploting.
subplot(3,5,6),...
plot(ethanolvariation(:,1),ethanolvariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 ethanolvariation(end,1) 0 ethanolvariation(end,3)]),...
title('RO2 vs Rs in Ethanol');
%%
n=7
ferulatemodel=ferulate(model);
i=0;
ferulatevariation=zeros(1,3);
while ferulatevariation(end,2)<0.0035

    i=i+0.01;
ferulatemodel=changeRxnBounds(ferulatemodel,'EX_ferulate[e]',-(i),'l');
ferulateresult=optimizeCbModel(ferulatemodel,'max','one',1);
for j=length(ferulatevariation)
    ferulatevariation(j+1,1)=i;
if isfield(ferulateresult,'f')
ferulatevariation(j+1,2)=ferulateresult.f;
else
ferulatevariation(j+1,2)=0;
end
if isempty(ferulateresult.x)
ferulatevariation(j+1,3)=0;
else
ferulatevariation(j+1,3)=-
ferulateresult.x(strmatch('EX_oxygen',ferulatemodel.rxns));
end
end
end
 ferulatevariation(all(ferulatevariation==0,2),:)=[]; %removing zeros for
faster ploting.
subplot(3,5,7),...
plot(ferulatevariation(:,1),ferulatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 ferulatevariation(end,1) 0 ferulatevariation(end,3)]),...
title('RO2 vs Rs in Ferulate');
 %%
n=8
gentisatemodel=gentisate(model);
i=0;
gentisatevariation=zeros(1,3);
while gentisatevariation(end,2)<0.0079
     i=i+0.01;
gentisatemodel=changeRxnBounds(gentisatemodel,'EX_gentisate[e]',-(i),'l');
gentisateresult=optimizeCbModel(gentisatemodel,'max','one',1);
for j=length(gentisatevariation)
    gentisatevariation(j+1,1)=i;
```

```
if isfield(gentisateresult,'f')
gentisatevariation(j+1,2)=gentisateresult.f;
else
gentisatevariation(j+1,2)=0;
end
if isempty(gentisateresult.x)
gentisatevariation(j+1,3)=0;
else
gentisatevariation(j+1,3)=-
gentisateresult.x(strmatch('EX_oxygen',gentisatemodel.rxns));
end
end
end
 gentisatevariation(all(gentisatevariation==0,2),:)=[]; %removing zeros for
faster ploting.
subplot(3,5,8),...
plot(gentisatevariation(:,1),gentisatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 gentisatevariation(end,1) 0 gentisatevariation(end,3)]),...
title('RO2 vs Rs in Gentisate');
 %%
n=9
heptanemodel=heptane(model);
i=0;
heptanevariation=zeros(1,3);
while heptanevariation(end,2)<0.0047
     i=i+0.01;
heptanemodel=changeRxnBounds(heptanemodel,'EX_nheptane[e]',-(i),'l');
heptaneresult=optimizeCbModel(heptanemodel,'max','one',1);
for j=length(heptanevariation)
    heptanevariation(j+1,1)=i;
if isfield(heptaneresult,'f')
heptanevariation(j+1,2)=heptaneresult.f;
else
heptanevariation(j+1,2)=0;
end
if isempty(heptaneresult.x)
heptanevariation(j+1,3)=0;
else
heptanevariation(j+1,3)=-
heptaneresult.x(strmatch('EX_oxygen',heptanemodel.rxns));
end
end
end
 heptanevariation(all(heptanevariation==0,2),:)=[]; %removing zeros for
faster ploting.
subplot(3,5,9),...
plot(heptanevariation(:,1),heptanevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 heptanevariation(end,1) 0 heptanevariation(end,3)]),...
title('RO2 vs Rs in Heptane');
%%
n=10
hydroxybenzoatemodel=hydroxybenzoate(model);
i=0;
hydroxybenzoatevariation=zeros(1,3);
while hydroxybenzoatevariation(end,2)<0.0080
     i=i+0.01;
```

```
hydroxybenzoatemodel=changeRxnBounds(hydroxybenzoatemodel,'EX_3hydroxybenzo
ate[e]',-(i),'l');
hydroxybenzoateresult=optimizeCbModel(hydroxybenzoatemodel,'max','one',1);
for j=length(hydroxybenzoatevariation)
    hydroxybenzoatevariation(j+1,1)=i;
if isfield(hydroxybenzoateresult,'f')
hydroxybenzoatevariation(j+1,2)=hydroxybenzoateresult.f;
else
hydroxybenzoatevariation(j+1,2)=0;
end
if isempty(hydroxybenzoateresult.x)
hydroxybenzoatevariation(j+1,3)=0;
else
hydroxybenzoatevariation(j+1,3)=-
hydroxybenzoateresult.x(strmatch('EX_oxygen',hydroxybenzoatemodel.rxns));
end
end
end
 hydroxybenzoatevariation(all(hydroxybenzoatevariation==0,2),:)=[];
%removing zeros for faster ploting.
subplot(3,5,10),...
plot(hydroxybenzoatevariation(:,1),hydroxybenzoatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 hydroxybenzoatevariation(end,1) 0
hydroxybenzoatevariation(end,3)]),...
title('RO2 vs Rs in 3-Hydroxybenzoate');

%%
n=11
octanemodel=octane(model);
i=0;
octanevariation=zeros(1,3);
while octanevariation(end,2)<0.0015
     i=i+0.01;
octanemodel=changeRxnBounds(octanemodel,'EX_noctane[e]',-(i),'l');
octaneresult=optimizeCbModel(octanemodel,'max','one',1);
for j=length(octanevariation)
    octanevariation(j+1,1)=i;
if isfield(octaneresult,'f')
octanevariation(j+1,2)=octaneresult.f;
else
octanevariation(j+1,2)=0;
end
if isempty(octaneresult.x)
octanevariation(j+1,3)=0;
else
octanevariation(j+1,3)=-
octaneresult.x(strmatch('EX_oxygen',octanemodel.rxns));
end
end
end
 octanevariation(all(octanevariation==0,2),:)=[]; %removing zeros for
faster ploting.
subplot(3,5,11),...
plot(octanevariation(:,1),octanevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 octanevariation(end,1) 0 octanevariation(end,3)]),...
title('RO2 vs Rs in Octane');
 %%
```

```
n=12
protocatechuatemodel=protocatechuate(model);
i=0;
protocatechuatevariation=zeros(1,3);
while protocatechuatevariation(end,2)<0.0094
     i=i+0.01;
protocatechuatemodel=changeRxnBounds(protocatechuatemodel,'EX_protocatechua
te[e]',-(i),'l');
protocatechuateresult=optimizeCbModel(protocatechuatemodel,'max','one',1);
for j=length(protocatechuatevariation)
    protocatechuatevariation(j+1,1)=i;
if isfield(protocatechuateresult,'f')
protocatechuatevariation(j+1,2)=protocatechuateresult.f;
else
protocatechuatevariation(j+1,2)=0;
end
if isempty(protocatechuateresult.x)
protocatechuatevariation(j+1,3)=0;
else
protocatechuatevariation(j+1,3)=-
protocatechuateresult.x(strmatch('EX_oxygen',protocatechuatemodel.rxns));
end
end
end
 protocatechuatevariation(all(protocatechuatevariation==0,2),:)=[];
%removing zeros for faster ploting.
subplot(3,5,12),...
plot(protocatechuatevariation(:,1),protocatechuatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 protocatechuatevariation(end,1) 0
protocatechuatevariation(end,3)]),...
title('RO2 vs Rs in Protocatechuate');
 %%
n=13
salicylatemodel=salicylate(model);
i=0;
salicylatevariation=zeros(1,3);
while salicylatevariation(end,2)<0.0121
     i=i+0.01;
salicylatemodel=changeRxnBounds(salicylatemodel,'EX_salicylate[e]',-
(i),'l');
salicylateresult=optimizeCbModel(salicylatemodel,'max','one',1);
for j=length(salicylatevariation)
    salicylatevariation(j+1,1)=i;
if isfield(salicylateresult,'f')
salicylatevariation(j+1,2)=salicylateresult.f;
else
salicylatevariation(j+1,2)=0;
end
if isempty(salicylateresult.x)
salicylatevariation(j+1,3)=0;
else
salicylatevariation(j+1,3)=-
salicylateresult.x(strmatch('EX_oxygen',salicylatemodel.rxns));
end
end
end
 salicylatevariation(all(salicylatevariation==0,2),:)=[]; %removing zeros
for faster ploting.
subplot(3,5,13),...
```

```
plot(salicylatevariation(:,1),salicylatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 salicylatevariation(end,1) 0 salicylatevariation(end,3)]),...
title('RO2 vs Rs in Salicylate');
%%
n=14
succinatemodel=succinate(model);
i=0;
succinatevariation=zeros(1,3);
while succinatevariation(end,2)<0.0231
     i=i+0.01;
succinatemodel=changeRxnBounds(succinatemodel,'EX_succinate[e]',-(i),'l');
succinateresult=optimizeCbModel(succinatemodel,'max','one',1);
for j=length(succinatevariation)
    succinatevariation(j+1,1)=i;
if isfield(succinateresult,'f')
succinatevariation(j+1,2)=succinateresult.f;
else
succinatevariation(j+1,2)=0;
end
if isempty(succinateresult.x)
succinatevariation(j+1,3)=0;
else
succinatevariation(j+1,3)=-
succinateresult.x(strmatch('EX_oxygen',succinatemodel.rxns));
end
end
end
 succinatevariation(all(succinatevariation==0,2),:)=[]; %removing zeros for
faster ploting.
subplot(3,5,14),...
plot(succinatevariation(:,1),succinatevariation(:,3)),...
xlabel('Rs - mmol/(gDW*h)'),...
ylabel('RO2 - mmol/(gDW*h)'),...
axis([0 succinatevariation(end,1) 0 succinatevariation(end,3)]),...
title('RO2 vs Rs in Succinate');
```

### *FVAcDCE.m*

```
%%FVAcDCE.m - objective: analyse bottleneck reactions in common FBA.
%% André Guerra 13/08/2015
cDCEmodel=cDCE(model);
 %% excludes exchange
 cDCEex=cDCEmodel.rxns(strmatch('EX_',cDCEmodel.rxns));
diff=setdiff(cDCEmodel.rxns,cDCEex);
a=[];
for i=1:length(diff);
    b=strmatch(diff(i),cDCEmodel.rxns,'exact');
    if ~isempty(b)
        a(length(a)+1,1)=b;
    end
end
 posintra=a;
 %% Runs FBA and FVA
 FBA=optimizeCbModel(cDCEmodel,'max','one');
 [FVAmin,FVAmax]=fluxVariability(cDCEmodel,100,'max',cDCEmodel.rxns);
 %% calculates the flux variability (normalized to max)
 fbaintra=FBA.x(posintra);
varmin=FVAmin(posintra);
```

```
varmax=FVAmax(posintra);
 figure;plot(1:1:length(varmax),fbaintra,'sqy');hold on;
plot(1:length(varmin),varmin,'*r');hold on;
plot(1:1:length(varmax),varmax,'.b'),
axis([1 length(varmin) -2 2]),xlabel('iJS666 Intracellular
Reactions'),ylabel('Intracellular Flux Values - mmol/(gDW*h)');hold off;
 indbottle=[];
for i=1:length(posintra)
    if abs(FBA.x(posintra(i)))>0.1 && abs(FBA.x(posintra(i))-
FVAmin(posintra(i)))<0.001 && abs(FBA.x(posintra(i))-
FVAmax(posintra(i)))<0.001;
        indbottle(length(indbottle)+1,1)=posintra(i);
    end
end
 bottlerxn=cDCEmodel.rxns(indbottle);
 printRxnFormula(cDCEmodel,bottlerxn);
```

### *Gluta.m*

```
%% gluta.m
%% See the influence of glutathione 0 and 2% of biomass content on growth
rate.
%% André Guerra - 10-08-2015
i=0.001:0.001:0.033*2;%mmol gdw
per=0.30732*(i);%*g mmol = g/gw
perper=100*per;% in %
val=ones(size(per))-per;
%%
cDCEmodel=cDCE(model);
 %%
res=[];
ox=[];
m=strmatch('glutathione[c]',cDCEmodel.mets);
r=strmatch('NEWRXN18975418598',cDCEmodel.rxns);
 for j=1:1:length(i)
    cDCEmodel=cDCE(model);
    cDCEmodel.S(:,r)=(val(j)).*(cDCEmodel.S(:,r));
    cDCEmodel.S(strmatch('BiomassJS666[c]',model.mets),r)=1;
    cDCEmodel.S(m,r)=-(i(j));
    FBA=optimizeCbModel(cDCEmodel,'max');
    if isfield(FBA,'f')
        res(j)=FBA.f;
    else
        res(j)=0;
    end

    if isempty(FBA.x)
        ox(j)=0;
    else
        ox(j)=-FBA.x(strmatch('EX_oxygen',cDCEmodel.rxns));
    end
  end
 a=[perper',res'];
maxres=max(res);
maxresx=perper(find(res'==maxres));
figure;
subplot(1,2,1);plot(a(:,1),a(:,2),'+g'),...
xlabel('% (w/w) Glutathione in Biomass'),ylabel('Specific Growth Rate -
(1/h)'),...
axis([0 2 0 0.0047]);
    b=[perper',ox'];
```

```
maxox=max(ox);
subplot(1,2,2);plot(b(:,1),b(:,2),'+b'),...
xlabel('% (w/w) Glutathione in Biomass'),ylabel('R02 - mmol/(gDW*h)'),...
axis([0 2 0 1.1]);
```

***Testobj.m***

```
%%Influencia da função objectivo nos fluxos medidos
cDCEmodel=cDCE(model);

li={'NEWRXN6661b';'NEWRXN6662';'NEWRXN6663';'NEWRXN6664';...
    'NEWRXN6665b';'NEWRXN6666';'NEWRXN6667';'NEWRXN6668';'NEWRXN6669';...
    'NEWRXN66610';'GLYOCARBOLIG-RXN';...
    'SERINE-GLYOXYLATE-AMINOTRANSFERASE-RXN';'MALSYN-RXN'};
indices=[strmatch('EX_',model.rxns);strmatch('SINK',model.rxns)];
emedium=model.rxns(indices);
%other reactions redox
la={'ATPSYN-RXN';'TRANS-RXN0-277';'RXN0-5266';'NADH-DEHYDROG-A-RXN';'NADH-
DEHYDROG-A-RXN';'RXN-14107';'1.10.2.2-RXN';'CYTOCHROME-C-OXIDASE-RXN';'RXN-
9510';'RXN-12444';'RXN0-271';'1.5.5.1-RXN';}
% important reactions names
access=[li;emedium;la];
% get indices for access
ind=[];
for i=1:length(access)
    ind(i,1)=strmatch(access(i),cDCEmodel.rxns);
end
%% FVA under max biomass (rs constant) - evolution drives system for
maximum biomass yield bt substrate uptake.
% Without glutathione in biomass
FBA=optimizeCbModel(cDCEmodel,'max','one');
[minimo,maximo] = fluxVariability(cDCEmodel,100,'max',access);
% minimo=(1/FBA.x(strmatch('SINK001',model.rxns))).*(minimo);
% maximo=(1/FBA.x(strmatch('SINK001',model.rxns))).*(maximo);
minimo=num2cell(minimo);
maximo=num2cell(maximo);
% With glutathione in biomass
i=0.033;%mmol gdw
per=0.30732*(i);%*g mmol = g/gw
val=ones(size(per))-per;
m=strmatch('glutathione[c]',cDCEmodel.mets);
r=strmatch('NEWRXN18975418598',cDCEmodel.rxns);
for j=1:length(i)
    cDCEmodel=cDCE(model);
    cDCEmodel.S(:,r)=(val(j)).*(cDCEmodel.S(:,r));
    cDCEmodel.S(strmatch('BiomassJS666[c]',model.mets),r)=1;
    cDCEmodel.S(m,r)=-(i(j));
end
[minimo2,maximo2] = fluxVariability(cDCEmodel,100,'max',access);
minimo2=num2cell(minimo2);
maximo2=num2cell(maximo2);




%% FVA under min redox potencial (NADH)- cells reduce the number of
oxidizing reactions thus conserving their energy of using it in the most
efficient way.
cDCEmodel=cDCE(model);
%changing obj
```

```
%descobrir todas as reacções que utilizam NADH, esteq e sinal.
idmet=strmatch('NADH[c]',cDCEmodel.mets);
estsinal=cDCEmodel.S(idmet,:);
num=length(find(estsinal~=0));
%nome das reacções
names=cDCEmodel.rxns(estsinal~=0,1);
cDCEmodel = changeObjective(cDCEmodel,names,(1/num).*estsinal');
%making biomass constant and equal to maximum
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_cDCE[e]',-0.561,'b');
% Without glutathione in biomass
[minimo3,maximo3] = fluxVariability(cDCEmodel,100,'min',access);
minimo3=num2cell(minimo3);
maximo3=num2cell(maximo3);
% With glutathione in biomass
i=0.033;%mmol gdw
per=0.30732*(i);%*g mmol = g/gw
val=ones(size(per))-per;
m=strmatch('glutathione[c]',cDCEmodel.mets);
r=strmatch('NEWRXN18975418598',cDCEmodel.rxns);
for j=1:1:length(i)
    cDCEmodel.S(:,r)=(val(j)).*(cDCEmodel.S(:,r));
    cDCEmodel.S(strmatch('BiomassJS666[c]',model.mets),r)=1;
    cDCEmodel.S(m,r)=-(i(j));
end
[minimo4,maximo4] = fluxVariability(cDCEmodel,100,'min',access);
minimo4=num2cell(minimo4);
maximo4=num2cell(maximo4);
%% FVA under min ATP consumption (rs constante) - cells grow with the
minimal usage of energy, thus conserving the most energy possible.
cDCEmodel=cDCE(model);
%changing obj
%descobrir todas as reacções que utilizam NADH, esteq e sinal.
idmet=strmatch('ATP[c]',cDCEmodel.mets);
estsinal=cDCEmodel.S(idmet,:);
num=length(find(estsinal~=0));
%nome das reacções
names=cDCEmodel.rxns(estsinal~=0,1);
cDCEmodel = changeObjective(cDCEmodel,names,(1/num).*estsinal');
%making biomass constant and equal to maximum
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_cDCE[e]',-0.561,'b');
% Without glutathione in biomass
[minimo5,maximo5] = fluxVariability(cDCEmodel,100,'min',access);
minimo5=num2cell(minimo5);
maximo5=num2cell(maximo5);
% With glutathione in biomass
i=0.033;%mmol gdw
per=0.30732*(i);%*g mmol = g/gw
val=ones(size(per))-per;
m=strmatch('glutathione[c]',cDCEmodel.mets);
r=strmatch('NEWRXN18975418598',cDCEmodel.rxns);

for j=1:1:length(i)
    cDCEmodel.S(:,r)=(val(j)).*(cDCEmodel.S(:,r));
    cDCEmodel.S(strmatch('BiomassJS666[c]',model.mets),r)=1;
    cDCEmodel.S(m,r)=-(i(j));
end
[minimo6,maximo6] = fluxVariability(cDCEmodel,100,'min',access);
minimo6=num2cell(minimo6);
maximo6=num2cell(maximo6);
%LIST
names = printRxnFormula(cDCEmodel,access,0,0,0,'FID666',0,0);
```

```
bind=[access names minimo maximo minimo2 maximo2 minimo3 maximo3 minimo4
maximo4 minimo5 maximo5 minimo6 maximo6];
```

***TNGAM.m***

```
%%FVAcDCE.m - objective: analyse bottleneck reactions in common FBA.
%% André Guerra 17/08/2015
cDCEmodel=cDCE(model);
 %% excludes exchange
 cDCEex=cDCEmodel.rxns(strmatch('EX_',cDCEmodel.rxns));
diff=setdiff(cDCEmodel.rxns,cDCEex);
a=[];
for i=1:length(diff);
    b=strmatch(diff(i),cDCEmodel.rxns,'exact');
    if ~isempty(b)
        a(length(a)+1,1)=b;
    end
end
 posintra=a;
 %% Runs FBA and FVA
 FBA=optimizeCbModel(cDCEmodel,'max','one');
 [FVAmin,FVAmax]=fluxVariability(cDCEmodel,100,'max',cDCEmodel.rxns);
 %% calculates the flux variability (normalized to max)
 fbaintra=FBA.x(posintra);
varmin=FVAmin(posintra);
varmax=FVAmax(posintra);

figure;plot(1:1:length(varmax),fbaintra,'sqy');hold on;
plot(1:length(varmin),varmin,'*r');hold on;
plot(1:1:length(varmax),varmax,'.b'),
axis([1 length(varmin) -2 2]),xlabel('iJS666 Intracellular
Reactions'),ylabel('Intracellular Flux Values - mmol/(gDW*h)');hold off;
 indbottle=[];
for i=1:length(posintra)
    if abs(FBA.x(posintra(i)))>0.1 && abs(FBA.x(posintra(i))-
FVAmin(posintra(i)))<0.001 && abs(FBA.x(posintra(i))-
FVAmax(posintra(i)))<0.001;
        indbottle(length(indbottle)+1,1)=posintra(i);
    end
end
 bottlerxn=cDCEmodel.rxns(indbottle);
 printRxnFormula(cDCEmodel,bottlerxn);
```

***Nitrogen.m***

```
%% Nitrogen.m
%% Tests the ingluence of different nitrogen sources in the specific
biomass growth.
%% 15-08-2015 - André Guerra
 %% Nitrogen Sources
 %% Ammonium
 cDCEmodel=cDCE(model);
ammoniumresult=optimizeCbModel(cDCEmodel,'max','one',1);
 %% Nitrate
 cDCEmodel=cDCE(model);
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_ammonium[e]',0,'b');
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_nitrate[e]',-100,'l');
nitrateresult=optimizeCbModel(cDCEmodel,'max','one',1);
 %% Nitrite
 cDCEmodel=cDCE(model);
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_ammonium[e]',0,'b');
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_nitrite[e]',-100,'l');
nitriteresult=optimizeCbModel(cDCEmodel,'max','one',1);
 %% Hydroxilamine
 cDCEmodel=cDCE(model);
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_ammonium[e]',0,'b');
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_hydroxylamine[e]',-100,'l');
hydroxilamineresult=optimizeCbModel(cDCEmodel,'max','one',1);
 %% Ammonia
 cDCEmodel=cDCE(model);
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_ammonium[e]',0,'b');
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_ammonia[e]',-100,'l');
ammoniaresult=optimizeCbModel(cDCEmodel,'max','one',1);
 %%
i=1:5;
figure;subplot(1,2,1); plot(i,[ammoniumresult.f, nitrateresult.f,
nitriteresult.f hydroxilamineresult.f ammoniaresult.f],'*');axis([0 6
0.0036 0.0060]),...
xlabel('Different Nitrogen Sources'),...
ylabel('Specific Growth Rate - (1/h)');
subplot(1,2,2); plot(i,-
[ammoniumresult.x(strmatch('EX_ammonium[e]',cDCEmodel.rxns))
nitrateresult.x(strmatch('EX_nitrate[e]',cDCEmodel.rxns))...
    nitriteresult.x(strmatch('EX_nitrite[e]',cDCEmodel.rxns))
hydroxilamineresult.x(strmatch('EX_hydroxylamine[e]',cDCEmodel.rxns))...
    ammoniaresult.x(strmatch('EX_ammonia[e]',cDCEmodel.rxns))],'*');axis([0
6 0 0.05]),...
xlabel('Different Nitrogen Sources'),...
ylabel('Specific Growth Rate - (1/h)');
```

***Sulfur.m***

```
%% sulfur.m
%% Influence of different sulfur sources in cDCE degradation
%% 15-08-2015 - André Guerra
 cDCEmodel=cDCE(model);
 %% Sulfur
 %% Sulfate
 cDCEmodel=cDCE(model);
% cDCEmodel = changeRxnBounds(cDCEmodel,rxnNameList,value,boundType)
%
%
% pos13=strmatch('EX_sulfate[e]',emedium);
% cDCEmodel=changeRxnBounds(cDCEmodel,emedium(pos13),-100,'l');
% cDCEmodel=changeRxnBounds(cDCEmodel,emedium(pos13),0,'u');
sulfateresult=optimizeCbModel(cDCEmodel,'max','one',1);
%% Thiosulfate
 cDCEmodel=cDCE(model);
 cDCEmodel=changeRxnBounds(cDCEmodel,'EX_sulfate[e]',0,'b');
 cDCEmodel=changeRxnBounds(cDCEmodel,'EX_thiosulfate[e]',-100,'l');
 thiosulfateresult=optimizeCbModel(cDCEmodel,'max','one',1);
 %% Sulfite
cDCEmodel=cDCE(model);
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_sulfate[e]',0,'b');
cDCEmodel=changeRxnBounds(cDCEmodel,'EX_sulfite[e]',-100,'l');
sulfiteresult=optimizeCbModel(cDCEmodel,'max','one',1);
%% Hydrogen Sulfide
 cDCEmodel=cDCE(model);
 cDCEmodel=changeRxnBounds(cDCEmodel,'EX_sulfate[e]',0,'b');
 cDCEmodel=changeRxnBounds(cDCEmodel,'EX_hydrogensulfide[e]',-100,'l');
 hydrogensulresult=optimizeCbModel(cDCEmodel,'max','one',1);
%% Sulfur
 cDCEmodel=cDCE(model);
 cDCEmodel=changeRxnBounds(cDCEmodel,'EX_sulfate[e]',0,'b');
 cDCEmodel=changeRxnBounds(cDCEmodel,'EX_S(sup)0(/sup)[e]',-100,'l');
 sulfuresult=optimizeCbModel(cDCEmodel,'max','one',1);
i=1:5
figure;subplot(1,2,1); plot(i,[sulfateresult.f thiosulfateresult.f
sulfiteresult.f hydrogensulresult.f sulfuresult.f],'*');axis([0 6 0.0046
0.0050]);
xlabel('Different Sulfur Sources'),...
ylabel('Specific Growth Rate - (1/h)');
subplot(1,2,2); plot(i,-
[sulfateresult.x(strmatch('EX_sulfate[e]',cDCEmodel.rxns))
thiosulfateresult.x(strmatch('EX_thiosulfate[e]',cDCEmodel.rxns))
sulfiteresult.x(strmatch('EX_sulfite[e]',cDCEmodel.rxns))
hydrogensulresult.x(strmatch('EX_hydrogensulfide[e]',cDCEmodel.rxns))
sulfuresult.x(strmatch('EX_S(sup)0(/sup)[e]',cDCEmodel.rxns))],'*');axis([0
6 0 0.0005]);
xlabel('Different Sulfur Sources'),...
ylabel('Sulfur Source Consumption Rate - mmol/(gDW*h)');
 %%
```

***Testsub.m***

```
%% testsub.m
%% This script produces a list of putative biostimulants.
%% André Guerra - 7-09-2015

testsub=cell(length(cDCEmodel.mets),6);
cDCEmodel=cDCE(model);
cDCEinitial=optimizeCbModel(cDCEmodel,'max','one',1);
 %reacçoes utilizadas
usedrxns=[];
for i=1:length(cDCEinitial.x)
    if cDCEinitial.x(i)~=0
        usedrxns(i)=1;
    else
        usedrxns(i)=0;
    end
end
%%

for i=1:length(cDCEmodel.mets)
    %% Define se o metabolito é intracellular
    if findstr('[c]', char(cDCEmodel.mets(i)));
        cDCEmodel=cDCE(model);
        % Adicina transport
        [cDCEmodel,~]=addReaction(cDCEmodel,
strcat('TEST_',char(cDCEmodel.mets(i))), {'ATP[c]',
strrep(char(cDCEmodel.mets(i)),'[c]','[e]'),'H(sub)2(/sub)O[e]',
char(cDCEmodel.mets(i)),'ADP[c]','phosphate[c]','H(sup)+(/sup)[c]'},[-1 -1
-1 1 1 1 1]);
        %Define constraints no transport
        %
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TEST_',char(cDCEmodel.mets(i)))
,0,'l');%
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TEST_',char(cDCEmodel.mets(i)))
,0.561,'u');
```

```
        % Adiciona exchange
[cDCEmodel,~]=addReaction(cDCEmodel,strcat('TESTEX_',char(cDCEmodel.mets(i)
)), {strrep(char(cDCEmodel.mets(i)),'[c]','[e]')} ,-1);
         %Define constraints na exchange
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TESTEX_',char(cDCEmodel.mets(i)
)),0,'u');
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TESTEX_',char(cDCEmodel.mets(i)
)),-0.561,'l');
        %Run
        res=optimizeCbModel(cDCEmodel,'max','one');
        if res.f>cDCEinitial.f && res.f<(cDCEinitial.f*1.30)
                if
res.x(strmatch('EX_cDCE[e]',cDCEmodel.rxns))<cDCEinitial.x(strmatch('EX_cDC
E[e]',cDCEmodel.rxns))
                disp('não está a usar todo o cDCE')
                else
                    DCEmodel=changeRxnBounds(cDCEmodel,'EX_cDCE[e]',0,'b');
%
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TEST_',char(cDCEmodel.mets(i)))
,0,'l');
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TEST_',char(cDCEmodel.mets(i)))
,10,'u');
%
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TESTEX_',char(cDCEmodel.mets(i)
)),0,'u');
cDCEmodel=changeRxnBounds(cDCEmodel,strcat('TESTEX_',char(cDCEmodel.mets(i)
)),-10,'l');
                    cat=optimizeCbModel(cDCEmodel,'max','one');
                           if cat.f==0

                            %test new used rxs
                            nusedrxns=[];
                            for j=1:length(cDCEinitial.x)
                                if res.x(j)==0
                                        nusedrxns(j)=0;
                                else
                                        nusedrxns(j)=1;
                                end
                            end
   %Finds the percentage of reactions useb on both simulations.
                            equal=[];
                            for k=1:length(cDCEinitial.x)
                                if usedrxns(k)==1 && nusedrxns(k)==1
                                    equal(k)=1;
                                end
                            end

    %Finds newly activated rxns
                            newly=[];
                            for m=1:length(cDCEinitial.x)
                                if usedrxns(m)==0 && nusedrxns(m)==1
                                    newly(m)=1;
                                end
                            end
                       testsub(i,1)=cDCEmodel.mets(i);
                       testsub(i,2)={num2str(((res.f/cDCEinitial.f)-
1)*100)};
testsub(i,3)={num2str(res.x(end,1))};
testsub(i,4)={num2str(res.x(strmatch('EX_cDCE[e]',cDCEmodel.rxns)))};
testsub(i,5)={num2str((sum(equal)/sum(usedrxns))*100)};
testsub(i,6)={num2str(sum(newly))};
```

```
else
    disp('Catabolic repression')
```

## 7.2. Supplementary Data 2 - Flux values for the constrained reactions in Figure 3.7.

| Reaction ID | Reaction Formula | Flux |
|---|---|---|
| 1.7.2.2-RXN | 7 $H^+$[c] + 6 reduced c-cytochrome[c] + nitrite[c] → 6 oxidized c-cytochrome[c] + 2 $H_2O$[c] + ammonia[c] | 0.1373 |
| 2OXOGLUTARATEDEH-RXN | NAD$^+$[c] + coenzyme A[c] + 2-oxoglutarate[c] → NADH[c] + $CO_2$[c] + succinyl-CoA[c] | 0.2534 |
| 2PGADEHYDRAT-RXN | 2-phospho-D-glycerate[c] <=> $H_2O$[c] + phosphoenolpyruvate[c] | 0.263 |
| AMONITRO-RXN | NADH[c] + oxygen[c] + ammonia[c] → NAD$^+$[c] + $H^+$[c] + $H_2O$[c] + hydroxylamine[c] | 0.1373 |
| ATPSYN-RXN | 3 $H^+$[c] + $H_2O$[c] + ATP[c] <=> phosphate[c] + ADP[c] + 4 H+[e] | -0.5222 |
| CITSYN-RXN | $H_2O$[c] + oxaloacetate[c] + acetyl CoA[c] → H+[c] + coenzyme A[c] + citrate[c] | 0.2544 |
| FUMHYDR-RXN | (S)-malate[c] <=> $H_2O$[c] + fumarate[c] | -0.2565 |
| GLYOCARBOLIG-RXN | $H^+$[c] + 2 glyoxylate[c] → CO2[c] + tartronate semialdehyde[c] | 0.2798 |
| NADH-DEHYDROG-A-RXN | NADH[c] + 3 $H^+$[c] + menaquinone-7[c] → NAD$^+$[c] + 2 $H^+$[e] + menaquinol-7[c] | 0.6042 |
| NEWRXN1363457344 | chloride[e] <=> chloride[c] | -1.122 |
| NEWRXN23601623232501156 | $CO_2$[e] <=> $CO_2$[c] | -1.0525 |
| NEWRXN56135355317 | oxygen[e] <=> oxygen[c] | 1.0584 |
| NEWRXN66523692141213412 | cDCE[e] → cDCE[c] | 0.561 |
| NEWRXN6661b | NADH[c] + $H^+$[c] + oxygen[c] + cDCE[c] → NAD$^+$[c] + $H_2O$[c] + 2,2-dichloroacetaldehyde[c] | 0.561 |
| NEWRXN6662 | NAD$^+$[c] + $H_2O$[c] + 2,2-dichloroacetaldehyde[c] → NADH[c] + $H^+$[c] + 2,2-dichloroacetate[c] | 0.561 |
| NEWRXN6663 | $H_2O$[c] + 2,2-dichloroacetate[c] → $H^+$[c] + chloride[c] + chloroglycolate[c] | 0.561 |
| NEWRXN6664 | chloroglycolate[c] → $H^+$[c] + chloride[c] + glyoxylate[c] | 0.561 |
| NGAM | $H_2O$[c] + ATP[c] → $H^+$[c] + phosphate[c] + ADP[c] | 0.45 |
| RXN-3482 | pyruvate[c] + hydroxylamine[c] → $H_2O$[c] + pyruvic oxime[c] | 0.1373 |
| RXN-3483 | oxygen[c] + pyruvic oxime[c] → $H^+$[c] + pyruvate[c] + nitrite[c] | 0.1373 |
| RXN0-5266 | 4 $H^+$[c] + 2 ubiquinol-8[c] + oxygen[c] → 2 ubiquinone-8[c] + 2 $H_2O$[c] + 4 $H^+$[e] | 0.2227 |
| SUCCCOASYN-RXN | coenzyme A[c] + ATP[c] + succinate[c] <=> phosphate[c] + ADP[c] + succinyl CoA[c] | -0.2531 |

## 7.3 Supplementary Data 3 - List of FVA on iJS666 model using different objectives.

| | MAX BIOMASS | | | | MIN $\Sigma$ (NADH) | | | | MIN $\Sigma$ (ATP) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reaction ID | NO GSH | | GSH | | NO GSH | | GSH | | NO GSH | | GSH | |
| | MIN | MAX | MIN | MAX | MIN | MAX | MIN | MAX | MIN | MAX | MIN | MAX |
| 'NEWRXN6661b' | 5.61E-01 | 5.61E-01 | 5.61E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 |
| 'NEWRXN6662' | 5.61E-01 | 5.61E-01 | 5.61E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 |
| 'NEWRXN6663' | 5.61E-01 | 5.61E-01 | 5.61E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 |
| 'NEWRXN6664' | 5.61E-01 | 5.61E-01 | 5.61E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 4.26E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 | 3.90E-01 | 5.61E-01 |
| 'NEWRXN6665b' | 0.00E+00 | 8.61E-06 | 0.00E+00 | 2.87E-05 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.71E-01 | 0.00E+00 | 1.71E-01 |
| 'NEWRXN6666' | 0.00E+00 | 8.61E-06 | 0.00E+00 | 2.87E-05 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.71E-01 | 0.00E+00 | 1.71E-01 |
| 'NEWRXN6667' | 0.00E+00 | 8.61E-06 | 0.00E+00 | 2.87E-05 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.71E-01 | 0.00E+00 | 1.71E-01 |
| 'NEWRXN6668' | 0.00E+00 | 8.61E-06 | 0.00E+00 | 2.87E-05 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.71E-01 | 0.00E+00 | 1.71E-01 |
| 'NEWRXN6669' | 0.00E+00 | 8.61E-06 | 0.00E+00 | 2.87E-05 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.71E-01 | 0.00E+00 | 1.71E-01 |
| 'NEWRXN66610' | 0.00E+00 | 8.61E-06 | 0.00E+00 | 2.87E-05 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.35E-01 | 0.00E+00 | 1.71E-01 | 0.00E+00 | 1.71E-01 |
| 'GLYOCARBOLIG-RXN' | 2.80E-01 | 2.80E-01 | 2.80E-01 | 2.80E-01 | 0.00E+00 | 7.22E-01 | 0.00E+00 | 7.22E-01 | 0.00E+00 | 7.22E-01 | 0.00E+00 | 7.22E-01 |
| SERINE-GLYOXYLATE-AMINO… | -2.26E-03 | -2.14E-03 | -2.40E-03 | -2.00E-03 | -7.22E-01 | 6.02E-01 | -7.22E-01 | 6.02E-01 | -7.22E-01 | 5.14E-01 | -7.22E-01 | 5.14E-01 |
| 'MALSYN-RXN' | 3.51E-03 | 3.62E-03 | 3.52E-03 | 3.89E-03 | 0.00E+00 | 8.82E-01 | 0.00E+00 | 8.82E-01 | 0.00E+00 | 8.82E-01 | 0.00E+00 | 8.82E-01 |
| 'EX_(3E5Z)tetradecadienoate[e]' | 0.00E+00 | 2.93E-07 | 0.00E+00 | 9.75E-07 | 0.00E+00 | 5.15E-03 | 0.00E+00 | 5.15E-03 | 0.00E+00 | 5.03E-03 | 0.00E+00 | 5.03E-03 |
| 'EX_Dglucono15lactone[e]' | 0.00E+00 | -2.22E-22 | 0.00E+00 | 0.00E+00 | 0.00E+00 | -1.71E-27 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 6.84E-25 |
| 'EX_Co(sup)2+(/sup)[e]' | -5.06E-08 | 0.00E+00 | -1.69E-07 | 0.00E+00 | -7.92E-04 | 0.00E+00 | -7.92E-04 | 2.57E-50 | -8.22E-04 | -8.49E-27 | -8.22E-04 | 6.78E-25 |
| 'EX_Fe(sup)3+(/sup)[e]' | -2.85E-07 | -1.58E-07 | -5.77E-07 | -1.54E-07 | -2.11E-03 | 5.42E-26 | -2.11E-03 | 4.29E-25 | -2.17E-03 | -7.61E-27 | -2.17E-03 | 1.72E-26 |
| 'EX_Mg(sup)2+(/sup)[e]' | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 |
| 'EX_Mo(sup)2+(/sup)[e]' | -1.41E-07 | 0.00E+00 | -4.70E-07 | 0.00E+00 | -1.92E-03 | 0.00E+00 | -1.92E-03 | 0.00E+00 | -2.01E-03 | 0.00E+00 | -2.01E-03 | 0.00E+00 |
| 'EX_Ni(sup)2+(/sup)[e]' | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 | -1.00E+02 | 0.00E+00 |
| 'EX_Zn(sup)2+(/sup)[e]' | -8.18E-05 | 0.00E+00 | -2.72E-04 | 0.00E+00 | -1.29E+00 | 0.00E+00 | -1.29E+00 | 0.00E+00 | -1.29E+00 | 0.00E+00 | -1.29E+00 | 0.00E+00 |
| 'EX_CO(sub)2(/sub)[e]' | 1.05E+00 | 1.05E+00 | 1.05E+00 | 1.05E+00 | 9.99E-01 | 1.12E+00 | 9.99E-01 | 1.12E+00 | 9.98E-01 | 1.12E+00 | 9.98E-01 | 1.12E+00 |
| 'EX_Na(sup)+(/sup)[e]' | 3.64E-20 | 3.64E-20 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | -2.28E-22 | -2.28E-22 | 3.80E-24 | 3.80E-24 |
| 'EX_carbonmonoxide[e]' | 1.58E-06 | 1.58E-06 | 1.54E-06 | 1.55E-06 | 0.00E+00 | 1.14E-06 | 0.00E+00 | 1.12E-06 | 0.00E+00 | 1.16E-06 | 0.00E+00 | 1.14E-06 |
| 'EX_indole3acetate[e]' | 0.00E+00 | 5.42E-07 | 0.00E+00 | 1.80E-06 | 0.00E+00 | 9.12E-03 | 0.00E+00 | 9.12E-03 | 0.00E+00 | 9.67E-03 | 2.81E-26 | 9.67E-03 |
| 'EX_H(sub)2(/sub)[e]' | 0.00E+00 | 1.17E-05 | 0.00E+00 | 3.89E-05 | 0.00E+00 | 1.84E-01 | 0.00E+00 | 1.84E-01 | 0.00E+00 | 2.57E-01 | 0.00E+00 | 2.57E-01 |
| 'EX_K(sup)+(/sup)[e]' | -8.18E-05 | 0.00E+00 | -2.72E-04 | 0.00E+00 | -1.29E+00 | 0.00E+00 | -1.29E+00 | 0.00E+00 | -1.29E+00 | 0.00E+00 | -1.29E+00 | 0.00E+00 |
| 'EX_epoxyethylene[e]' | 0.00E+00 | 1.41E-06 | 0.00E+00 | 4.70E-06 | 0.00E+00 | 2.18E-02 | 0.00E+00 | 2.18E-02 | 0.00E+00 | 2.38E-02 | 0.00E+00 | 2.38E-02 |
| 'EX_betaDglucose[e]' | 0.00E+00 | -2.22E-22 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 1.28E-28 | 0.00E+00 | -3.57E-28 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 7.35E-25 |
| 'EX_Larabitol[e]' | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 4.10E-28 | 0.00E+00 | 2.45E-29 | 0.00E+00 | -4.86E-24 | 0.00E+00 | 0.00E+00 |
| 'EX_hippurate[e]' | 0.00E+00 | 6.34E-07 | 0.00E+00 | 2.11E-06 | 0.00E+00 | 1.05E-02 | 0.00E+00 | 1.05E-02 | 0.00E+00 | 1.15E-02 | 0.00E+00 | 1.15E-02 |
| 'EX_phosphate[e]' | -3.99E-03 | -3.97E-03 | -4.24E-03 | -4.18E-03 | -2.57E-01 | 0.00E+00 | -2.57E-01 | 0.00E+00 | -2.57E-01 | 0.00E+00 | -2.57E-01 | 0.00E+00 |
| 'EX_ammonium[e]' | -2.07E-02 | -2.07E-02 | -2.14E-02 | -2.14E-02 | -6.68E-02 | 0.00E+00 | -6.68E-02 | 0.00E+00 | -6.86E-02 | 0.00E+00 | -6.86E-02 | 0.00E+00 |
| 'EX_chloride[e]' | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 | 1.12E+00 |
| 'EX_selenate[e]' | -7.49E-06 | -7.48E-06 | -7.32E-06 | -7.31E-06 | -5.42E-06 | 0.00E+00 | -5.30E-06 | 0.00E+00 | -5.51E-06 | 0.00E+00 | -5.39E-06 | 0.00E+00 |
| 'EX_ethanol[e]' | 0.00E+00 | 2.34E-06 | 0.00E+00 | 7.78E-06 | 0.00E+00 | 3.90E-02 | 0.00E+00 | 3.90E-02 | 0.00E+00 | 4.43E-02 | 0.00E+00 | 4.43E-02 |
| 'EX_sulfate[e]' | -8.70E-05 | -8.65E-05 | -2.38E-04 | -2.36E-04 | -7.63E-03 | 1.35E-14 | -7.63E-03 | -4.12E-30 | -7.81E-03 | -1.60E-13 | -7.82E-03 | -7.65E-25 |
| 'EX_oxygen[e]' | -1.06E+00 | -1.06E+00 | -1.06E+00 | -1.06E+00 | -1.15E+00 | -1.01E+00 | -1.15E+00 | -1.01E+00 | -1.16E+00 | -9.89E-01 | -1.16E+00 | -9.89E-01 |
| 'EX_cDCE[e]' | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 | -5.61E-01 |
| 'SINK001' | 4.65E-03 | 4.65E-03 | 4.59E-03 | 4.59E-03 | 0.00E+00 | 3.37E-03 | 0.00E+00 | 3.32E-03 | 0.00E+00 | 3.42E-03 | 0.00E+00 | 3.38E-03 |
| 'SINK002' | 3.16E-06 | 3.55E-06 | 3.09E-06 | 3.86E-06 | 0.00E+00 | 2.49E-03 | 0.00E+00 | 2.49E-03 | 0.00E+00 | 2.55E-03 | 0.00E+00 | 2.55E-03 |
| 'SINK003' | 2.32E-04 | 2.33E-04 | 2.27E-04 | 2.28E-04 | 0.00E+00 | 3.34E-03 | 0.00E+00 | 3.33E-03 | 1.41E-16 | 3.44E-03 | 0.00E+00 | 3.44E-03 |
| 'SINK004' | 0.00E+00 | 9.86E-07 | 0.00E+00 | 3.28E-06 | 0.00E+00 | 1.65E-02 | 0.00E+00 | 1.65E-02 | 0.00E+00 | 1.74E-02 | 0.00E+00 | 1.74E-02 |
| 'SINK008' | 0.00E+00 | 2.39E-07 | 0.00E+00 | 7.95E-07 | 0.00E+00 | 4.17E-03 | 0.00E+00 | 4.17E-03 | 0.00E+00 | 4.11E-03 | 0.00E+00 | 4.11E-03 |
| 'SINK010' | 0.00E+00 | 5.06E-08 | 0.00E+00 | 1.69E-07 | 0.00E+00 | 7.92E-04 | 0.00E+00 | 7.92E-04 | 0.00E+00 | 8.22E-04 | 0.00E+00 | 8.22E-04 |
| 'SINK012' | 3.79E-05 | 3.95E-05 | 3.70E-05 | 4.23E-05 | 0.00E+00 | 1.67E-02 | 0.00E+00 | 1.67E-02 | 0.00E+00 | 1.71E-02 | 0.00E+00 | 1.71E-02 |
| 'SINK013' | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 |
| 'SINK014' | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 | 0.00E+00 | 1.00E+02 |
| 'SINK015' | 0.00E+00 | 8.18E-05 | 0.00E+00 | 2.72E-04 | 0.00E+00 | 1.29E+00 | 0.00E+00 | 1.29E+00 | 0.00E+00 | 1.29E+00 | 0.00E+00 | 1.29E+00 |
| 'SINK016' | 0.00E+00 | 8.18E-05 | 0.00E+00 | 2.72E-04 | 0.00E+00 | 1.29E+00 | 0.00E+00 | 1.29E+00 | 0.00E+00 | 1.29E+00 | 0.00E+00 | 1.29E+00 |
| 'SINK018' | 0.00E+00 | 8.14E-08 | 0.00E+00 | 2.71E-07 | 0.00E+00 | 1.24E-03 | 0.00E+00 | 1.24E-03 | 0.00E+00 | 1.27E-03 | 0.00E+00 | 1.27E-03 |
| 'SINK020' | 0.00E+00 | 8.14E-08 | 0.00E+00 | 2.71E-07 | 0.00E+00 | 1.24E-03 | 0.00E+00 | 1.24E-03 | 0.00E+00 | 1.27E-03 | 0.00E+00 | 1.27E-03 |
| 'SINK021' | 0.00E+00 | 5.06E-08 | 0.00E+00 | 1.69E-07 | 0.00E+00 | 7.92E-04 | 0.00E+00 | 7.92E-04 | 0.00E+00 | 8.22E-04 | 0.00E+00 | 8.22E-04 |
| 'SINK022' | 0.00E+00 | 6.25E-07 | 0.00E+00 | 2.08E-06 | 0.00E+00 | 8.87E-03 | 0.00E+00 | 8.87E-03 | 0.00E+00 | 9.12E-03 | 0.00E+00 | 9.12E-03 |
| 'SINK023' | 0.00E+00 | 1.06E-06 | 0.00E+00 | 3.52E-06 | 2.02E-26 | 1.80E-02 | 0.00E+00 | 1.80E-02 | 2.90E-25 | 1.96E-02 | 0.00E+00 | 1.96E-02 |
| 'SINK024' | 0.00E+00 | 1.41E-07 | 0.00E+00 | 4.70E-07 | 0.00E+00 | 1.92E-03 | 0.00E+00 | 1.92E-03 | 0.00E+00 | 2.01E-03 | 0.00E+00 | 2.01E-03 |
| 'SINK025' | 0.00E+00 | 1.27E-07 | 0.00E+00 | 4.22E-07 | 0.00E+00 | 2.11E-03 | 0.00E+00 | 2.11E-03 | 0.00E+00 | 2.17E-03 | 0.00E+00 | 2.17E-03 |
| 'SINK026' | 0.00E+00 | 9.98E-07 | 0.00E+00 | 3.32E-06 | 0.00E+00 | 1.65E-02 | 0.00E+00 | 1.65E-02 | 0.00E+00 | 1.89E-02 | -1.75E-24 | 1.89E-02 |
| 'SINK027' | 0.00E+00 | 4.99E-07 | 0.00E+00 | 1.66E-06 | 0.00E+00 | 8.24E-03 | 0.00E+00 | 8.24E-03 | 0.00E+00 | 9.45E-03 | 0.00E+00 | 9.46E-03 |
| 'SINK028' | 0.00E+00 | 1.02E-06 | 0.00E+00 | 3.41E-06 | 0.00E+00 | 1.69E-02 | 0.00E+00 | 1.69E-02 | 0.00E+00 | 1.89E-02 | 0.00E+00 | 1.89E-02 |
| 'SINK029' | 0.00E+00 | 3.75E-07 | 0.00E+00 | 1.25E-06 | 0.00E+00 | 6.30E-03 | 0.00E+00 | 6.30E-03 | 0.00E+00 | 6.70E-03 | 0.00E+00 | 6.70E-03 |
| 'SINK030' | 0.00E+00 | 4.04E-07 | 0.00E+00 | 1.35E-06 | 0.00E+00 | 6.71E-03 | 0.00E+00 | 6.71E-03 | 0.00E+00 | 6.78E-03 | 0.00E+00 | 6.79E-03 |
| 'SINK031' | 0.00E+00 | 4.15E-07 | 0.00E+00 | 1.38E-06 | 0.00E+00 | 6.91E-03 | 0.00E+00 | 6.91E-03 | 0.00E+00 | 6.99E-03 | 0.00E+00 | 6.99E-03 |
| 'SINK032' | 8.65E-05 | 8.71E-05 | 2.36E-04 | 2.38E-04 | 0.00E+00 | 8.66E-03 | 0.00E+00 | 8.66E-03 | 1.60E-13 | 8.90E-03 | 0.00E+00 | 8.90E-03 |
| 'SINK033' | 0.00E+00 | 2.41E-07 | 0.00E+00 | 8.03E-07 | 0.00E+00 | 4.21E-03 | 0.00E+00 | 4.21E-03 | 0.00E+00 | 4.21E-03 | 0.00E+00 | 4.21E-03 |
| 'SINK034' | 0.00E+00 | 3.61E-07 | 0.00E+00 | 1.20E-06 | 0.00E+00 | 6.29E-03 | 0.00E+00 | 6.29E-03 | 0.00E+00 | 6.35E-03 | 0.00E+00 | 6.35E-03 |
| 'SINK035' | 0.00E+00 | 2.67E-07 | 0.00E+00 | 8.89E-07 | 0.00E+00 | 4.65E-03 | 0.00E+00 | 4.65E-03 | 0.00E+00 | 4.68E-03 | 0.00E+00 | 4.68E-03 |
| 'SINK036' | 0.00E+00 | 2.73E-07 | 0.00E+00 | 9.10E-07 | 0.00E+00 | 4.77E-03 | 0.00E+00 | 4.77E-03 | 0.00E+00 | 4.77E-03 | 0.00E+00 | 4.77E-03 |
| 'SINK037' | 0.00E+00 | 2.36E-07 | 0.00E+00 | 7.86E-07 | 0.00E+00 | 4.11E-03 | 0.00E+00 | 4.11E-03 | 0.00E+00 | 4.14E-03 | 0.00E+00 | 4.14E-03 |
| 'SINK038' | 0.00E+00 | 6.99E-07 | 0.00E+00 | 2.33E-06 | 0.00E+00 | 1.20E-02 | 0.00E+00 | 1.20E-02 | 0.00E+00 | 1.16E-02 | 0.00E+00 | 1.16E-02 |
| 'SINK039' | 0.00E+00 | 6.34E-07 | 0.00E+00 | 2.11E-06 | 0.00E+00 | 8.19E-03 | 0.00E+00 | 8.19E-03 | 0.00E+00 | 8.40E-03 | 0.00E+00 | 8.40E-03 |
| 'SINK040' | 0.00E+00 | 5.55E-07 | 0.00E+00 | 1.85E-06 | 0.00E+00 | 7.63E-03 | 0.00E+00 | 7.63E-03 | 0.00E+00 | 7.81E-03 | 0.00E+00 | 7.82E-03 |
| 'SINK041' | 0.00E+00 | 1.98E-07 | 0.00E+00 | 6.59E-07 | -1.33E-25 | 3.34E-03 | -1.64E-27 | 3.33E-03 | 0.00E+00 | 3.44E-03 | 0.00E+00 | 3.44E-03 |
| 'ATPSYN-RXN' | -5.22E-01 | -5.22E-01 | -5.19E-01 | -5.19E-01 | -1.16E+00 | -3.10E-01 | -1.16E+00 | -3.10E-01 | -1.16E+00 | -4.50E-01 | -1.16E+00 | -4.50E-01 |
| 'TRANS-RXN0-277' | 0.00E+00 | 4.09E-05 | 0.00E+00 | 1.36E-04 | 0.00E+00 | 6.43E-01 | 0.00E+00 | 6.43E-01 | 0.00E+00 | 6.43E-01 | 0.00E+00 | 6.43E-01 |
| 'RXN0-5266' | 2.23E-01 | 2.23E-01 | 2.21E-01 | 2.21E-01 | 0.00E+00 | 5.61E-01 | 0.00E+00 | 5.61E-01 | 0.00E+00 | 5.61E-01 | 0.00E+00 | 5.61E-01 |
| 'NADH-DEHYDROG-A-RXN' | 6.04E-01 | 6.04E-01 | 6.02E-01 | 6.02E-01 | 5.83E-02 | 1.12E+00 | 5.85E-02 | 1.12E+00 | 1.99E-01 | 1.12E+00 | 1.98E-01 | 1.12E+00 |
| 'NADH-DEHYDROG-A-RXN' | 6.04E-01 | 6.04E-01 | 6.02E-01 | 6.02E-01 | 5.83E-02 | 1.12E+00 | 5.85E-02 | 1.12E+00 | 1.99E-01 | 1.12E+00 | 1.98E-01 | 1.12E+00 |
| 'RXN-14107' | -9.99E+02 | 1.00E+03 | -9.99E+02 | 1.00E+03 | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 |
| '1.10.2.2-RXN' | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 | -1.00E+03 | 1.00E+03 |
| 'CYTOCHROME-C-OXIDASE-RXN' | 0.00E+00 | 2.05E-05 | 0.00E+00 | 6.81E-05 | 0.00E+00 | 3.04E-01 | 0.00E+00 | 3.04E-01 | 0.00E+00 | 3.04E-01 | 0.00E+00 | 3.04E-01 |
| 'RXN0-271' | 0.00E+00 | 4.09E-05 | 0.00E+00 | 1.36E-04 | 0.00E+00 | 6.43E-01 | 0.00E+00 | 6.43E-01 | 0.00E+00 | 6.43E-01 | 0.00E+00 | 6.43E-01 |
| '1.5.5.1-RXN' | 0.00E+00 | 1.19E-05 | 0.00E+00 | 3.91E-05 | -3.92E-29 | 1.61E-01 | 6.65E-27 | 1.61E-01 | 1.38E-23 | 2.14E-01 | 5.14E-25 | 2.14E-01 |