Genome Skimming of Symbiotic Communities – an *in silico* Evaluation of Assembly Approaches

Bastian Greshake†, Simonida Zehr†, Francesco Dal Grande*, Anjuli Meiser*, Imke Schmitt*, Ingo Ebersberger†

† Department for Applied Bioinformatics, Institute for Cell Biology and Neuroscience, Goethe University, Frankfurt am Main, Germany

* Biodiversity and Climate Research Centre, Senckenberg Gesellschaft für Naturforschung, Frankfurt am Main, Germany

Motivation

Lichens, an association of a filamentous fungus and one to several algal or cyanobacterial photobionts, are a hallmark for the success of mutualistic symbioses involving eukaryotes. They can colonize extreme ecological niches, frequently act as pioneering organisms, and are a promising resource for novel bioactive substances of medical and economical relevance. Still, the full potential of lichens for evolutionary and biotechnological studies has not been tapped, mainly since comprehensive genomic data are lacking. Extending the collection of lichen genomes is not trivial as a separate sequencing of the closely interacting symbionts is often not possible. Genome skimming of the lichen metagenome is an obvious and cost-effective solution to rapidly broaden the data basis for genomic research on lichens. Here we address the questions how to best assemble genome skimming data from a eukaryotic species mixture, what pitfalls can occur, and at what quality one can expect to reconstruct the individual genome sequences from a given experiment.

1. *in silico* Generation of Simulated Twin Sets

Starting from a real Illumina MiSeq sequencing experiment on the lichen *Lasallia pustulata*, consisting of 15 million read pairs (250 bp read length), we simulated Whole Genome Shotgun (WGS) reads with characteristics as close as possible to this data set. Reads were simulated using *ART*, parameterized with the read length, read number and insert length distribution of the *L. pustulata* sequencing experiment.

For the insert length distribution, we fitted a censored Weibull distribution to the observed insert sizes of the *L. pustulata* data set (Figure 1). The observed insert size distribution was generated by merging overlapping read pairs of the *L. pustulata* data set

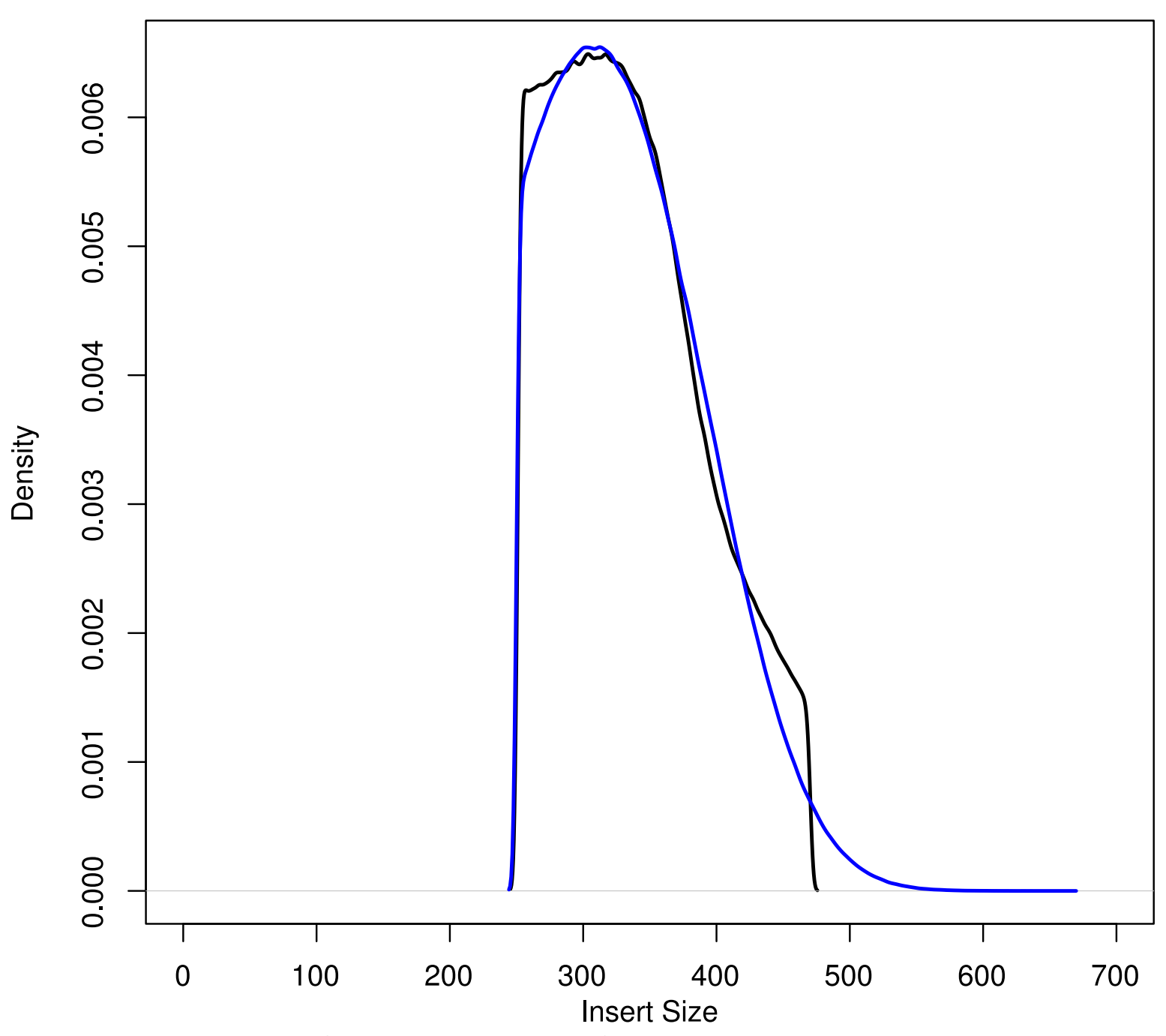


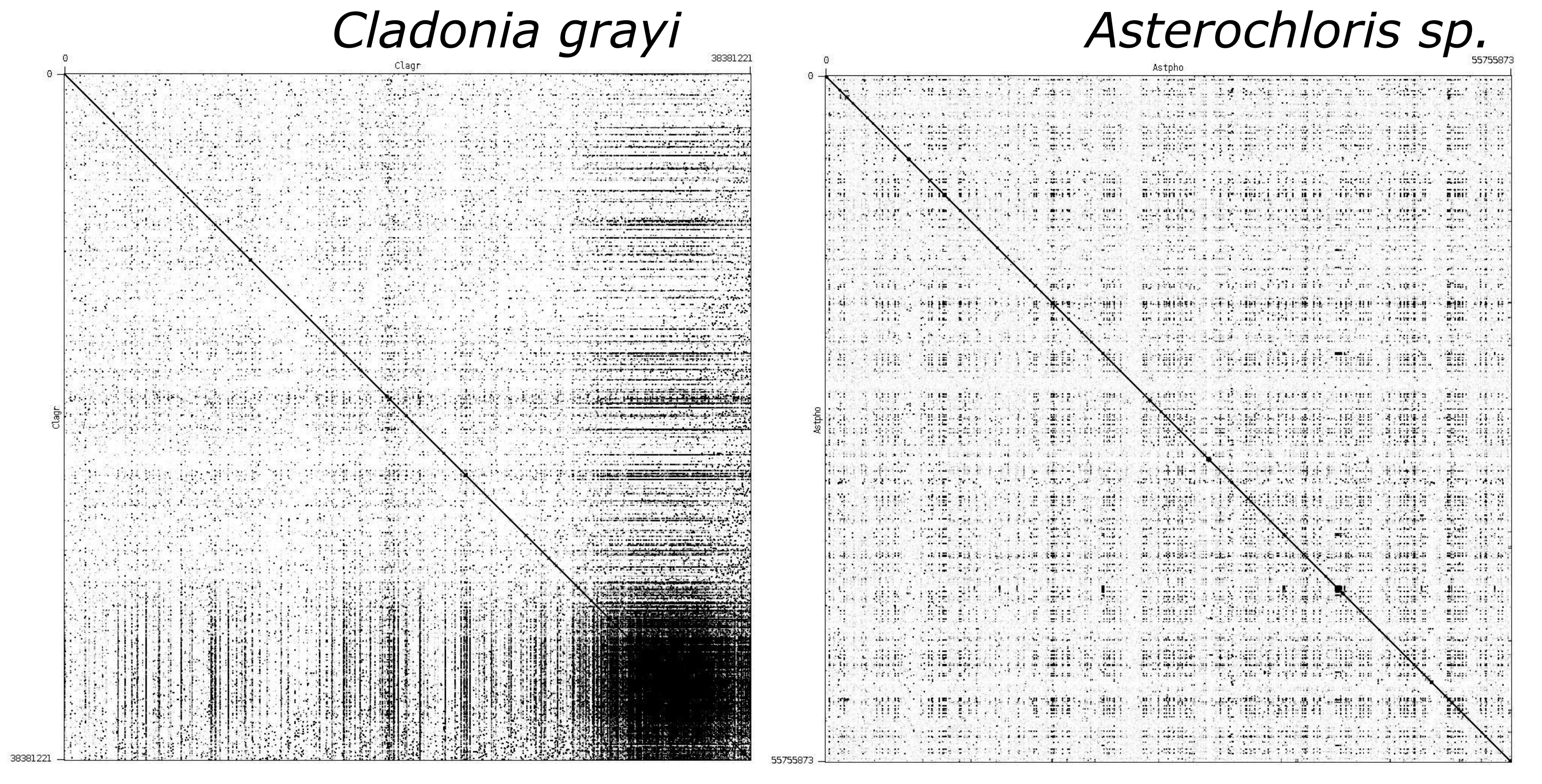
Figure 1: Insert Size distribution of the *L. pustulata* whole genome shotgun library (black: observed, blue: fitted Weibull distribution)

For our simulations we used the draft genomes of *Cladonia grayi* and *Asterochloris sp.* as templates for generating reads. We created single pseudo chromosomes for each of the two genomes by concatenating all scaffolds and removing ambiguous positions.

A RepeatMasker analysis (Box I), together with a dot plot analysis to display self similarities (Figure 2), revealed a substantial difference in repeat content between the two pseudo-chromosomes.

Box I Reference Genomes

	Cladonia grayi	*Asterochloris sp.*
Number of Scaffolds	1506	153
Total Length	38 Mbp	55 Mbp
GC content	44 %	58 %
% Repetitive	5 %	2.8 %


Cladonia grayi

Asterochloris sp.

Figure 2: Self-similarity of the pseudo-genomes of *C. grayi* and *Asterochloris sp.* visualized in a dot plot.

In total 11 different data sets with varying coverages for the two genomes were simulated (Table 1):
Two data sets only containing reads from *Cladonia grayi*, and *Asterochloris sp.* respectively, as well as 9 mixed species twin data sets. These contain varying coverage ratios for the two organisms, going from 9:1 to 1:9 in steps of 1.

Table 1: Relative and Absolute coverages for each organism per data set

Coverage Ratio *C. grayi* : *Asterochloris sp.*	Coverage *C. grayi*	Coverage *Asterochloris sp.*
10:0	182x	0x
9:1	157x	17x
8:2	134x	33x
7:3	112x	48x
6:4	92x	61x
5:5	74x	74x
4:6	56x	86x
3:7	40x	97x
2:8	26x	107x
1:9	13x	116x
0:10	0x	125x

2. Assembler Selection & Optimisation

Box II Assembler Choice

De Bruijn Graph based

- Velvet – Standard de Bruijn Graph
- MetaVelvet – Metagenome DBG Assembler
- SPAdes – Multisized de Bruijn Graph

Overlap Layout Consensus based

- MIRA – Regular Overlap Layout Consensus
- Omega – Metagenome OLC Assembler
- sga – String Graph Assembler

For Omega, sga, Velvet & MetaVelvet we explored the parameter space (overlap size and k-mer size respectively), using the maximization of the N50 size as the optimisation criterion.

3. Assembly Results

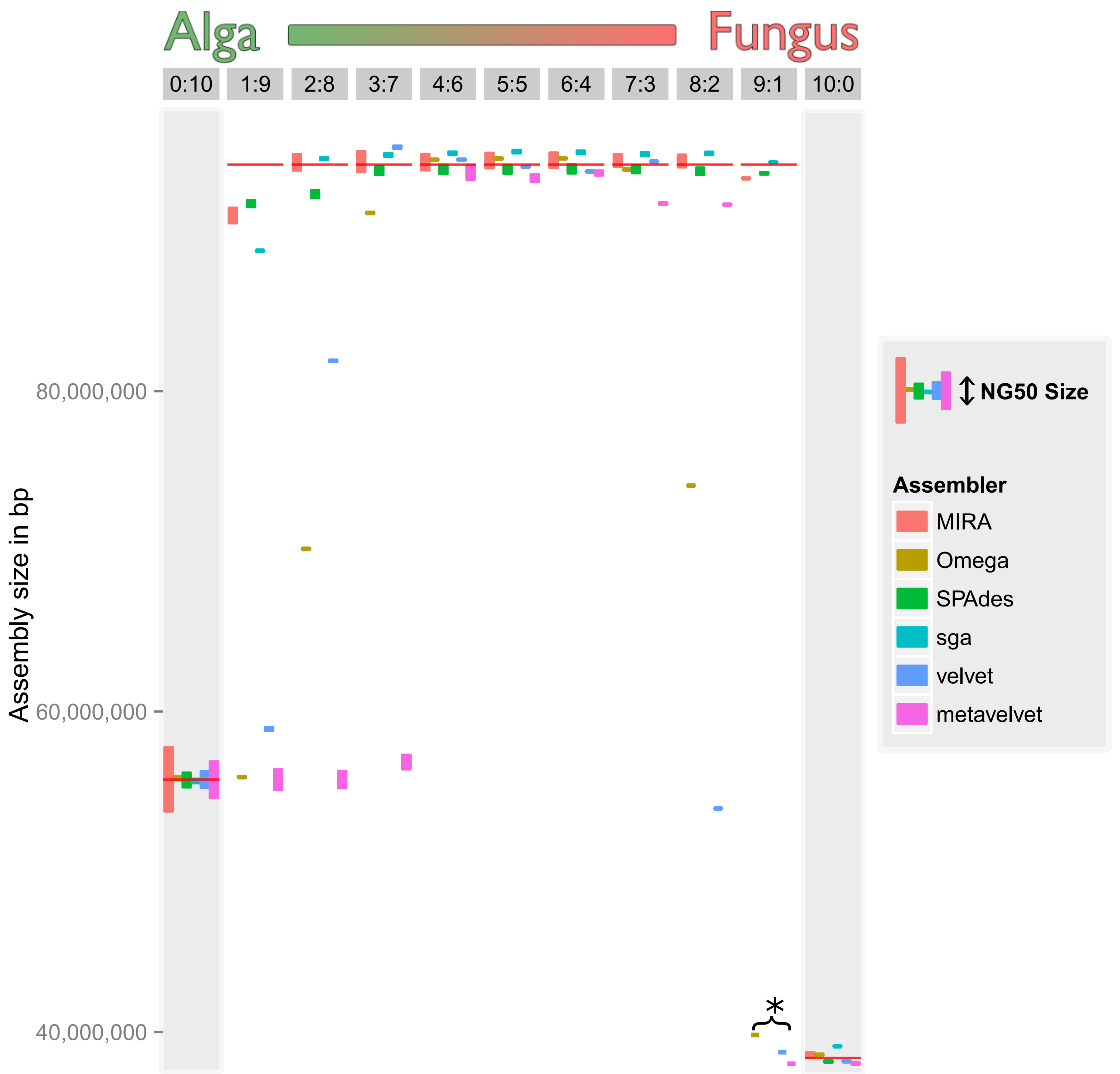


Figure 3: Assembly results over the 11 data sets. Bars are centered at total assembly length (red lines are reference lengths). Height of the bars shows the NG50 size. For the assemblies with the asterisk the total assembly length was less than 50% of the reference length. A default height was used in those instances.

For the single species data sets all assemblers succeeded in reconstructing the two genomes over the full length (Figure 3, column 0:10 and 10:0). Yet, the NG50 sizes varied substantially for the individual assemblies. In general the repeat-poor alga yielded much larger NG50 sizes compared to the repeat-rich fungus. For the alga the former contig boundaries of the reference genome can be spanned, while this is rarely possible for the fungus. (Fig. 4).

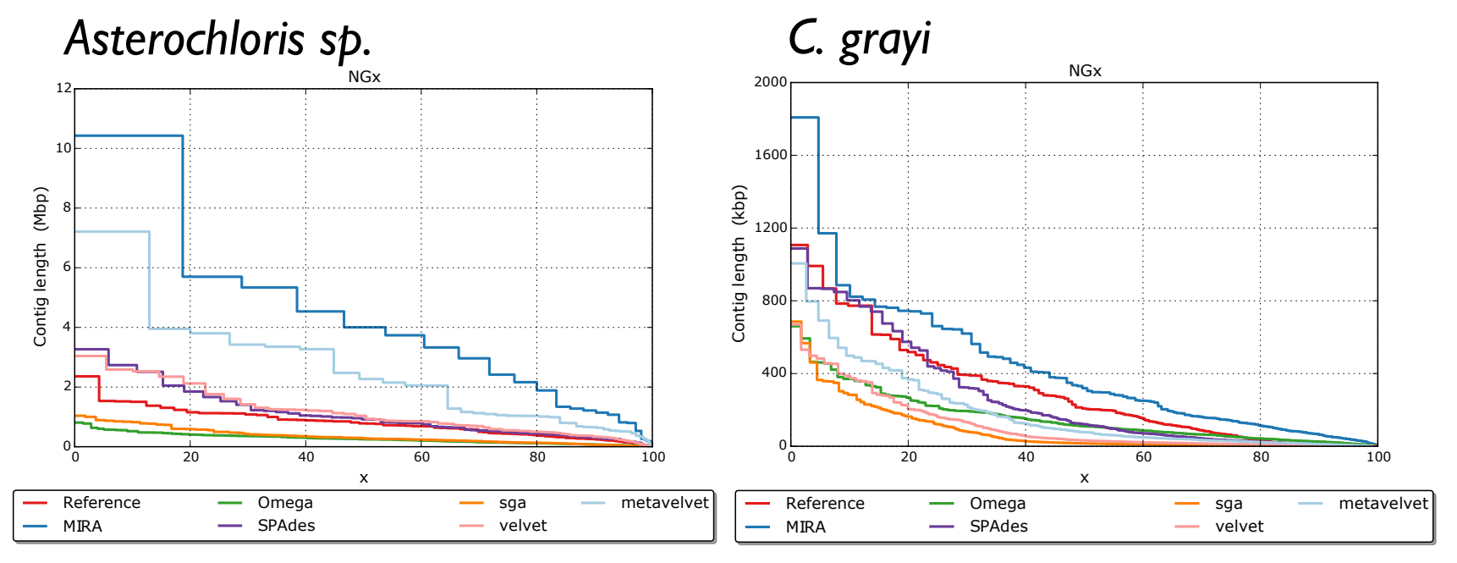


Figure 4: NGx distribution for *Asterochloris sp.* & *C. grayi*

For the mixed species data sets the full length genome reconstruction was not always possible, depending on the assembler and coverage ratio. For example, in some cases *MetaVelvet* failed to assemble both reference genomes, yielding only a single genome (Fig 3., Column 1:9 & 2:8).

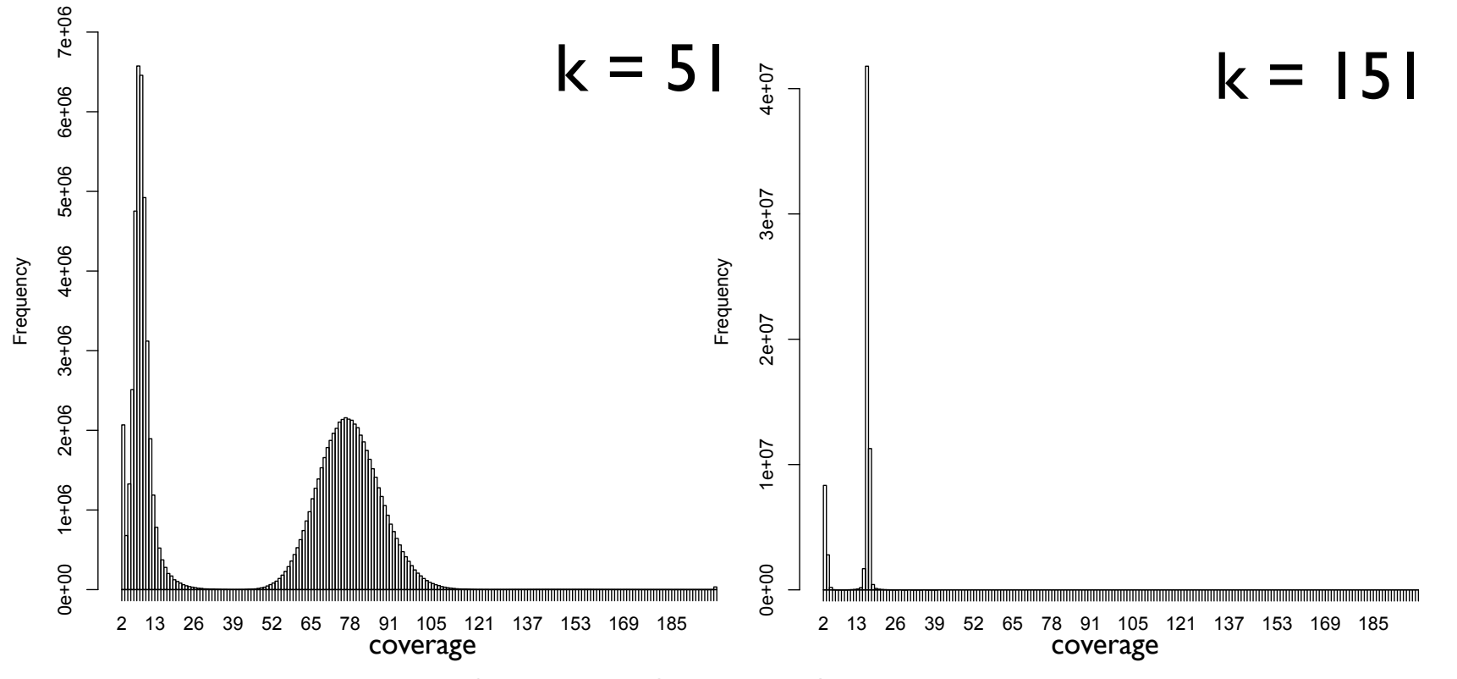


Figure 5: kmer-coverage frequency for the 1:9 data set.

This is a consequence of the assembly parameter optimisation strategy. By only optimizing for the N50 size, we ended up choosing a k-mer size which reduces the frequency of the fungal k-mers to an extend that they are no longer considered during assembly. (see Fig. 5). As an effect short contigs were not generated, and in turn the gene prediction suffered (Table 2).

Table 2: Number of fungal gene predictions mapping to the fungal reference genes.

Assembler	1:9	2:8	3:7
Reference	10740	10740	10740
MIRA	10348	10715	10718
Omega	72	5825	10302
SPAdes	10656	10666	10683
sga	10100	10674	10675
Velvet	2845	8817	10530
MetaVelvet	4	66	1657

As the NG50 could also be inflated by misassemblies, we checked for the NGA50. This metric does not use the contigs itself, but rather aligned blocks to calculate the NG metric. This does not change the relative ranking of the different assemblers (Fig. 6).

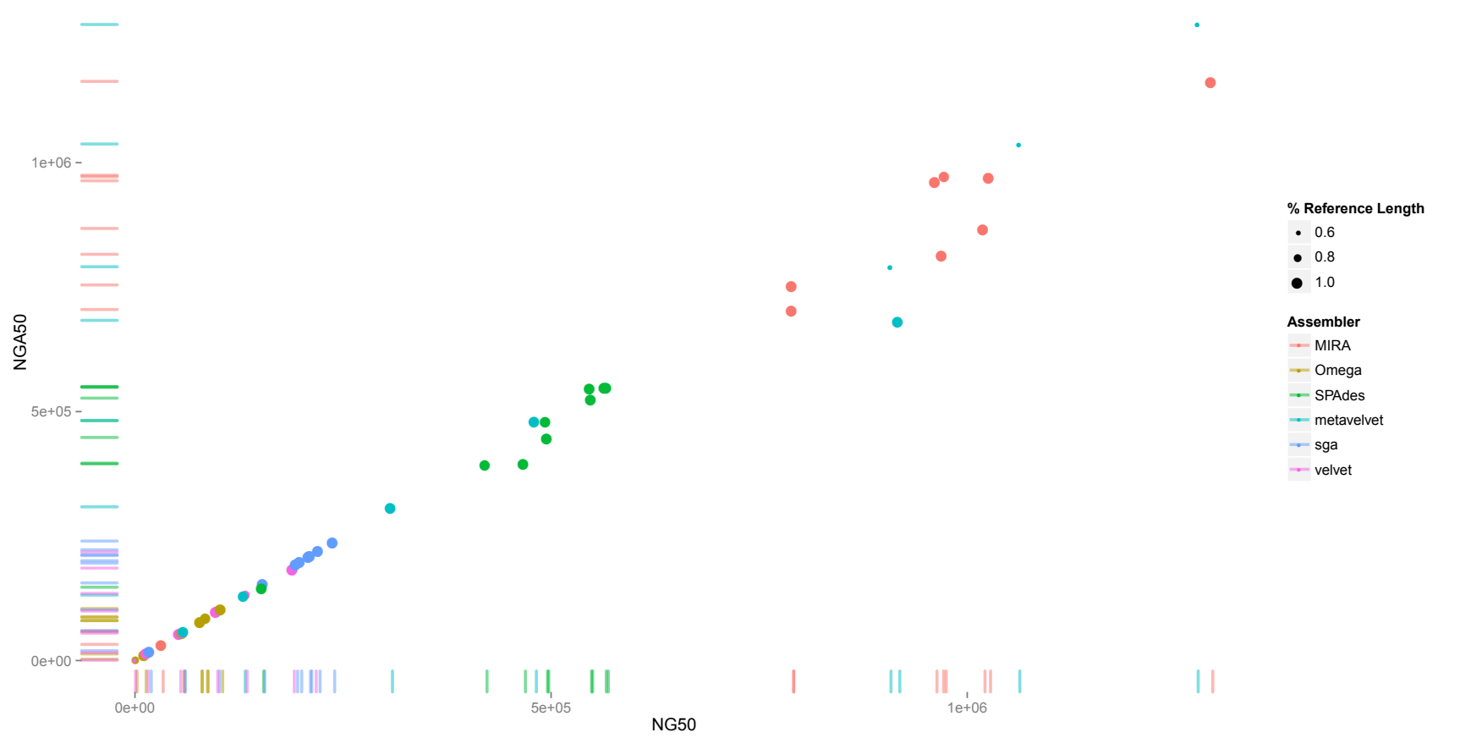


Figure 6: Comparing the NG50 (x-axis) to the NGA50 (y-axis).

Summary

- Already in single species data we see large differences in assembler performance
- In mixed species data sets the coverage distribution further inflates differences between the assemblers.
- Optimizing the N50 precludes sequences representing the low-coverage genome from assembly
- Excluding the formation of short contigs for the low coverage genome influences downstream analysis.
- Assembler performance in our data is not driven by misassemblies.

doi:10.6084/m9.figshare.1046678

Contact

Bastian Greshake
bgreshake@gmail.com
Goethe University, Frankfurt am Main, Germany
Max-von-Laue-Straße 13, 60438 Frankfurt am Main

References

[1] Huang W, Li L, Myers JR, and Marth GT. Bioinformatics (2012) 28 (4): 593-594
[2] http://sourceforge.net/projects/mira-assembler/
[3] Zerbino DR and Birney E. Genome Research (2008) 18:821-829.
[4] Simpson JT and Durbin R. Bioinformatics (2010) 26 (12): i367-i373
[5] Namiki T, Hachiya T, Tanaka H, Sakakibara Y. Nucleic Acids Res, (2012) 40(20), e155
[6] Huson DH, Auch AF, Qi J, et al. Genome Research (2007) 17: 000
[7| Stanke M, Steinkamp R, Waack S and Morgenstern B (2004) Nucleic Acids Research, Vol. 32, W309-W312
[8] Genis Parra, Keith Bradnam and Ian Korf (2007) Bioinformatics, 23: 1061-1067
[9] Trapnell C, Pachter L, Salzberg SL. Bioinformatics (2009) 25 (9): 1105-1111.
[10] http://sourceforge.net/projects/hamstr/
[11] Ebersberger I, de Matos Simoes R, Kupczok A, Gube M, Kothe E, Voigt K, and von Haeseler A. Mol Biol Evol (2012) 29 (5): 1319-1334
[12] Katoh, Standley (2013) Molecular Biology and Evolution 30:772-780
[13] Stamatakis A. Bioinformatics (2006) 22 (21): 2688-2690
[14] Conesa A, Götz S, Garcia-Gomez JM, Terol J, Talon M and Robles M. Bioinformatics, (2005) 21: 3674-3676.