

Architektura dużych projektów bioinformatycznych

Pakiety do obliczeń: naukowych,
Inżynierskich i statystycznych
Przegląd i porównanie

Bartek Wilczyński

10.4.2019

Plan na przyszły tydzień: quiz

- Kto używał debuggerów: pdb, pycharm, idle
- Kto używał profilerów: profile, cProfile, kernprofiler, runsnakerun
- Kto używał narzędzi do benchmarkowania: timeit, pystone, etc.
- Kto używał modułu inspect
- Kto używał systemów do logowania/raportowania błędów (logging, warnings)?

Plan na dziś

- Pakiety do obliczeń: przegląd zastosowań
- różnice w zapotrzebowaniu: naukowcy, inżynierowie, statystycy/medycy
- Matlab/octave/scipy
- S-Plus/SPSS/projekt R
- Mathematica/Maxima/Sage
- Pakiety komercyjne vs. Open Source
- Excel?

Typowi użytkownicy pakietów obliczeniowych

- Inżynierowie i projektanci (budownictwo, lotnictwo, motoryzacja, itp.)
- Naukowcy doświadczalni (fizycy, chemicy, materiałoznawcy, itp.)
- Statystycy (zastosowania w medycynie, ekonomii, biologii molekularnej, psychologii, socjologii, ubezpieczeniach, itp.)
- Matematycy (przede wszystkim matematyka stosowana)

Obliczenia naukowe

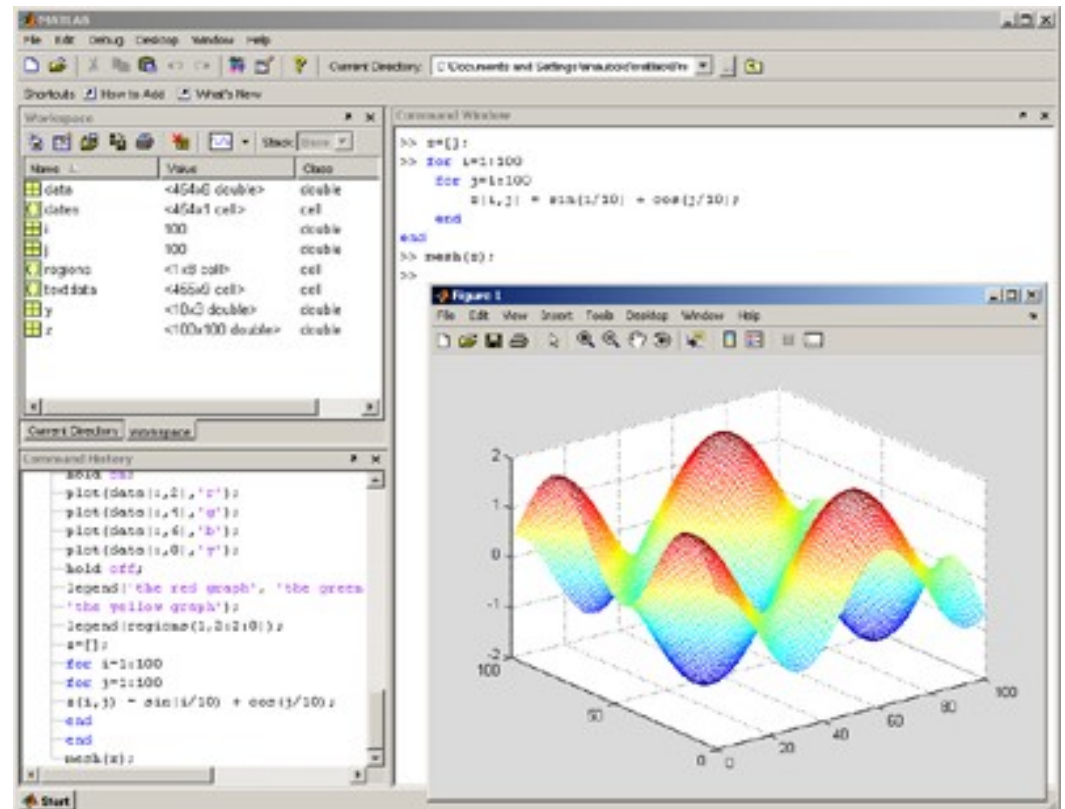
- Komputer jako “potężniejszy kalkulator”
- W zasadzie wszystko można zaprogramować samemu, ale każdemu mogą się przydać:
 - Interfejs użytkownika łatwiejszy niż typowego kompilatora
 - Możliwość zaawansowanej grafiki
 - Dobrze przetestowane standardowe procedury
 - Interfejsy do urządzeń
 - Wsparcie fachowców

Matlab i pakiety “inżynierskie”

- Rozwijany w latach 70'tych przez Clive Moler'a jako narzędzie dla studentów informatyki, aby nie musieli używać zaawansowanych bibliotek fortranu
- Firma mathworks powstaje w 1984 i wydaje pierwszą wersję Matlab'a
- Najpopularniejszy wśród inżynierów, dobre całki numeryczne, rozwiązywanie równań i wykresy (również 3d)
- Bardzo popularny także do przetwarzania sygnałów i symulacji (simulink)
- Licencja komercyjna – niedrogi dla studentów, droższy dla uczelni, bardzo drogi dla przemysłu

Toolbox'y Matlab'a

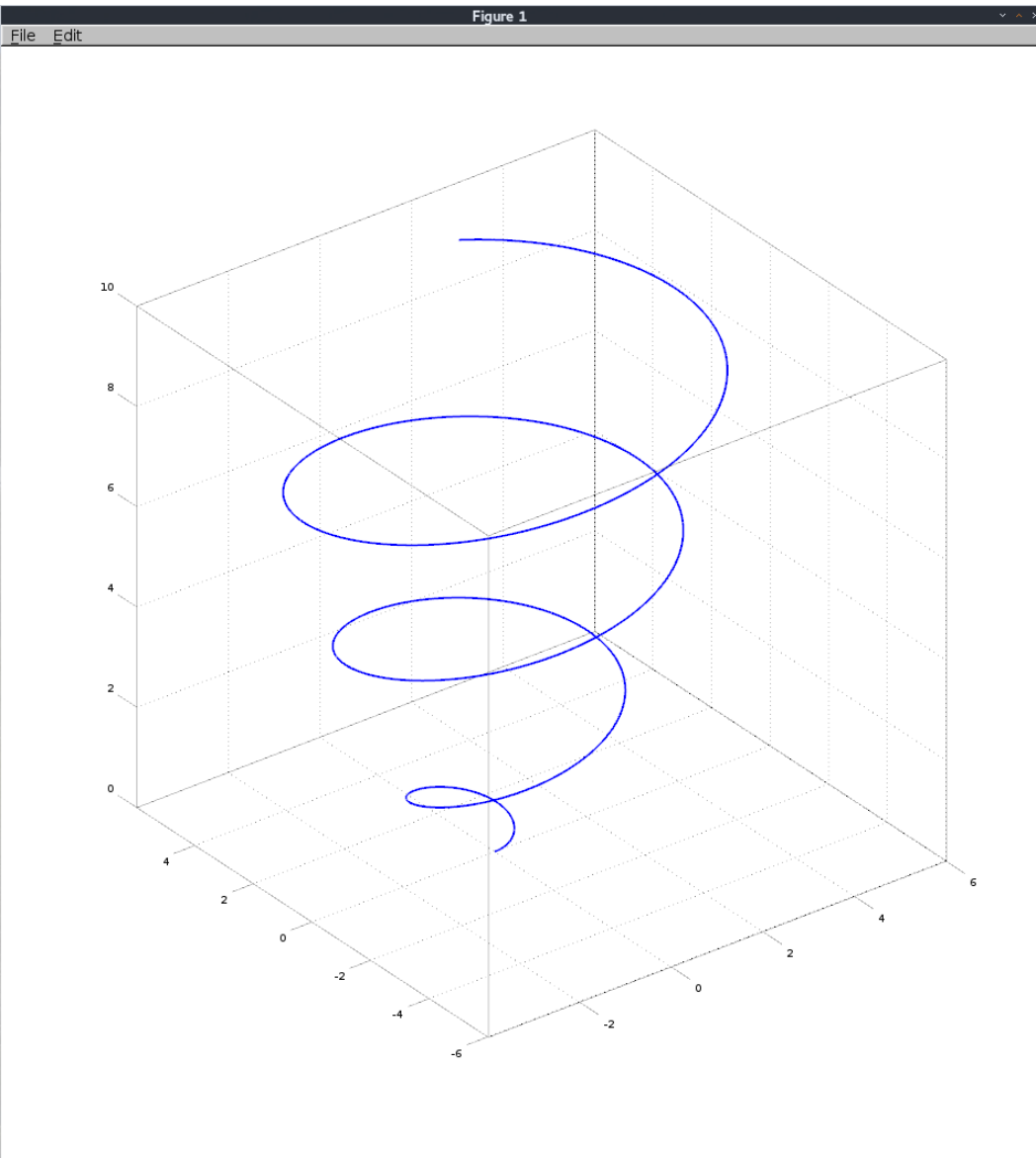
- Wiele dodatkowych (płatnych) bibliotek dla specjalistów
 - Symbolic math
 - Image processing
 - Financial toolbox
 - Bioinformatics
 - Optimization
 - SimBiology



Alternatywy openSource

- GNU Octave (rozpoczęty w 1988, wydania od 1992, rozwijany przez John'a W. Eatona, chemika z University of Wisconsin-Madison)
 - W zasadzie kompatybilny z Matlab'em
 - John W. Eaton Inc. - consulting, firma, która czerpie dochody z konsultingu Octave'a
- SciLab – dawniej psilab, rozwijany od lat 90tych w INRIA we Francji, przekształcony w spółkę Scilab enterprises, przejęty w 2017 przez ESI Group, dostępny na licencji GNU
- Scipy stack – zestaw bibliotek python'a do obliczeń naukowych
 - Wiele bibliotek, rozwijanych przez niezależne grupy
 - System pakietów, edytor i dystrybucja organizowana przez firmę Enthought, również komercyjne dystrybucje i consulting
 - Wiele konferencji tematycznych dla naukowców i pracowników przemysłu - także źródło dochodu

Interfejs Octave



```
octave
-----
~ » octave                                     asladic@lappy
GNU Octave, version 3.8.1
Copyright (C) 2014 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "x86_64-unknown-linux-gnu".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/get-involved.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

octave:1> t=[0:0.01:20];
octave:2> x=sqrt(t).*cos(t);
octave:3> y=sqrt(t).*sin(t);
octave:4> z=0.5*t;
octave:5> graph=plot3(x,y,z)
graph = -17.921
octave:6> set(graph(1), "linewidth", 2)
octave:7> □
```

Interfejs Enthought Canopy

The image displays the Canopy IDE interface, which is a Python development environment. The main window is titled "Editor - Canopy" and shows a code editor with Python code. The code includes comments and function calls, such as `num_vars = int` and `draw_circle_patch`. The interface is divided into several panels:

- File Browser:** Located at the top left, it shows a list of files and folders, with a filter set to "All Supported Files".
- Welcome to Canopy:** A central panel with the Canopy logo and a message: "Hi, welcome to Canopy! Login to your Enthought account or create one." Below this are three icons: "Editor", "Package Manager", and "Doc Browser".
- Recent files:** A list of recently opened files, including `application.py`, `application_launcher.py`, `util.py`, `prisma.py`, and `mac_metadata.py`. There are buttons for "Restore previous session" and "Open an existing file".
- Documentation Browser:** A window on the right side showing the "DOCUMENTATION" page. It includes a search bar with the URL `http://help.canopy/toplevel-docs.html` and a list of links for "Canopy User Guide" and "Online Help". The "Online Help" section lists various Python libraries and frameworks: Python Tutorial, Python Documentation, IPython, NumPy, SciPy, Traits, TraitsUI, Enaml, Envisage, Chaco, Mayavi, and Matplotlib Gallery. A "Tips" section provides instructions on how to use the documentation browser and code blocks within Canopy.
- Mayavi Scene 1:** A window in the bottom right corner displaying a 3D visualization of a complex, multi-colored, swirling structure, likely a visualization of a mathematical or scientific dataset.

The bottom status bar shows the cursor position (53 | 18) and the current language (Python).

S-Plus dla statystyków

- Język S zaprojektowany w laboratoriach Bell Labs przez Johna Chambers'a
- Implementacja przez R. Douglas'a Martina, profesora statystyki w Seattle
- Wydany komercyjnie w 1988 jako S-Plus, potem kolejno “przejmowany” przez różne korporacje aż do 2008, kiedy przejęła go firma TIBCO
- Adresowany do statystyków akademickich i przemysłowych
- Ogólny, bez specjalizacji w jakiejś dziedzinie zastosowań

Projekt R – Implementacja OpenSource języka S

- Rozpoczęty ok. 1995 projekt stworzenia darmowej implementacji języka S
- Ross Ithaka (obecnie Genentech) and Robert Gentleman (obecnie Univ. Auckland)
- W tej chwili zarządzany przez “R foundation”
- Wiele firm “wspierających” R
- Zachęcam do obejrzenia slajdów Chambers'a:
www.r-project.org/useR-2006/Slides/Chambers.pdf

Rstudio – interfejs do R'a

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for loading data, summarizing it, and creating a ggplot2 scatter plot titled "Diamond Pricing".
- Console:** Shows the execution output, including summary statistics for 'x', 'y', and 'z' (likely carat, price, and clarity), and the execution of the plot creation code.
- Workspace:** Lists the loaded data object 'diamonds' (53940 observations) and the 'p' object (ggplot object).
- Plots Panel:** Displays the resulting scatter plot of Price vs. Carat, colored by Clarity.

```
1 library(ggplot2)
2 source("plots/formatPlot.R")
3
4 view(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11 p <- qplot(carat, price,
12           data=diamonds, color=clarity,
13           xlab="Carat", ylab="Price",
14           main="Diamond Pricing")
15
```

Console Output:

```
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean : 5.731 Mean : 5.735 Mean : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max. :10.740 Max. :58.900 Max. :31.800
> summary(diamonds$price)
  Min. 1st Qu. Median Mean 3rd Qu. Max.
    326    950   2401  3933   5324 18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+           data=diamonds, color=clarity,
+           xlab="Carat", ylab="Price",
+           main="Diamond Pricing")
>
> format.plot(p, size=24)
>
```

Workspace Data:

Variable	Value
diamonds	53940 obs. of 10 variables
aveSize	0.7979
clarity	character [8]
p	ggplot [8]

Plots Panel: The plot is titled "Diamond Pricing" and shows Price on the y-axis (0 to 15000) and Carat on the x-axis (0.0 to 3.5). The data points are colored by Clarity, with a legend on the right showing categories: I1, SI2, SI1, VS2, VS1, VVS2, VVS1, and IF.

System pakietów w R

- Istotna jest możliwość rozwijania własnych “pakietów” w R (coś na kształt “toolbox'ów” Matlab'a)
- Jest to proces całkowicie demokratyczny, każdy może wysłać pakiet i umieścić go w repozytorium CRAN (Comprehensive R Archive Network)
- Możliwość automatycznej instalacji pakietów z CRAN
- Pewne standardy dokumentacji (winiety) zgodne z “literate programming”

Bioconductor

- Dystrybucja wybranych pakietów R do analizy danych biomedycznych
- Nacisk na łatwiejszą instalację i lepszą dokumentację pakietów
- Stabilny cykl wydań (2 razy do roku)
- Szkolenia adresowane do biologów i medyków
- Kompletnie not-for-profit
- Finansowany z grantów (ok 7-8 osób core team)

Obliczenia symboliczne - Mathematica

- Opracowana w latach 1980'tych przez Stephen'a Wolframa
- Jeden z pierwszych w historii pakietów umożliwiających obliczenia symboliczne
- Bardzo popularna wśród studentów amerykańskich, którzy muszą “zaliczyć” rachunek różniczkowy
- Obecnie także w wersji online: Wolfram Alpha
- Konkurencyjne pakiety: Maple, Mathcad

Mathematica - interfejs

The screenshot displays the Mathematica interface with several windows and components:

- Main Notebook Window:**
 - Code:**

```

aa = Table[A[[i, j]], {i, 1, 4}, {j, 1, 4}];
bb = Table[B[[i, j]], {i, 1, 4}, {j, 1, 4}];
ll = Table[L[[i]], {i, 1, 4}];
mm = Table[M[[i]], {i, 1, 4}];
ss = LinearSolve[aa, l];
(ss + mm)[[3]] / (aa + bb)[[3]]

```
 - Output:**

```

{0.}
{0.}
{1.}
{0.}
0.4

```
 - Text:**

■ p=2: Neumann-Randintegrale

Auf jedem Face kommen die grade. Daher: nur x = 1 wird be

```

nE[1][y_, z_] = y * (1 - z);
nE[2][y_, z_] = z * (1 - y);
nE[3][y_, z_] = z * (1 - y);
nE[4][y_, z_] = y * (1 - z);
For[i = 1, i <= 4, i++,
  Print[
    Integrate[Integrate[nF[y, z] * f[1, y, z], {y, 0, 1}], {z, 0, 1}]]

```
 - Output:**

```

0.0138889
0.0138889
0.0277778
0.0277778
0.00694444

```
- Help Window:**
 - Text:**

BesselJ[n, z] gives the Bessel function of the first kind $J_n(z)$.

$J_n(z)$ satisfies the differential equation $z^2 y'' + z y' + (z^2 - n^2) y = 0$.

BesselJ[n, z] has a branch cut discontinuity in the complex zplane running from $-\infty$ to 0.
 - Code:**

```

Plot[{BesselJ[0, x], BesselJ[1, x], BesselJ[2, x], BesselJ[9, x]},
  {x, 0, 20},
  PlotStyle -> {RGBColor[1, 0, 0], RGBColor[0, 1, 0], RGBColor[0, 0, 1],
    RGBColor[1, 0, 1]}]

```
 - Figure:** A 2D plot showing four Bessel functions: $J_0(x)$ (red), $J_1(x)$ (green), $J_2(x)$ (blue), and $J_9(x)$ (magenta) over the range $x \in [0, 20]$.
- Bottom Window:**
 - Code:**

```

In[1]= ListPlot3D[Table[Sin[x y] + Random[Real, {-0.15, 0.15}],
  {x, 0, 3 Pi / 2}, {y, 0, Pi / 15}]]];

```
 - Figure:** A 3D surface plot of a function with a grid, showing a wavy surface with peaks and valleys, colored with a gradient from blue to yellow.

Obliczenia symboliczne Open Source

- Maxima (1992-), a wcześniej Macsyma (1968-1982)
- Wydana w 1998 na licencji GPL
- Napisana w języku lisp
- Wiele konkurencyjnych interfejsów (WXMaxima, Gmaxima itp)
- Maxima skupiona na obliczeniach symbolicznych, bez większej funkcjonalności w numeryce

SAGE math cloud

(dawniej sage notebook)

- Stosunkowo nowy projekt (inny niż sage synapse)
- Połączenie wielu środowisk obliczeniowych
 - Python (Numpy, Scipy, Sympy, matplotlib, Networkx)
 - Maxima
 - R
 - GAP, FLINT, GD, JMOL, PALP, Singular
- Środowisko w przeglądarce, sesja na serwerze lub “w chmurze”

Interfejs SAGE

Use Sage to Solve Equations

last edited on April 11, 2011 05:45 PM by admin

[Save](#) [Save & quit](#) [Discard & quit](#)

File... Action... Data... sage Typeset

[Print](#) [Worksheet](#) [Edit](#) [Text](#) [Undo](#) [Share](#) [Publish](#)

```
var('a b c d e f x y')
```

```
(a, b, c, d, e, f, x, y)
```

```
show(solve(a*x^2 + b*x + c == 0, x)[0])
```

$$x = -\frac{b + \sqrt{-4ac + b^2}}{2a}$$

```
show(solve(x^3 + a*x + b == 0, x)[0])
```

$$x = \frac{(-i\sqrt{3}+1)a}{6\left(\frac{1}{18}\sqrt{4a^3+27b^2}\sqrt{3}-\frac{1}{2}b\right)^{\frac{1}{3}}} - \frac{1}{2}(i\sqrt{3}+1)\left(\frac{1}{18}\sqrt{4a^3+27b^2}\sqrt{3}-\frac{1}{2}b\right)^{\frac{1}{3}}$$

```
solve([a*x + b*y == c, d*x + e*y == f], x, y)
```

```
[[x == -(b*f - c*e)/(a*e - b*d), y == (a*f - c*d)/(a*e - b*d)]]
```

Excel?

- Najpopularniejszy pakiet do obliczeń
- Bardzo prosty interfejs
- Często stosowany również w bio-informatyce
- Ma spore ograniczenia (np. Maksymalna liczba linii w arkuszu), które utrudniają rozwój projektów prowadzonych w arkuszu
- Brak możliwości efektywnego testowania,
- brak debuggerów

**WHEN IT COMES TO DATA
ANALYSIS**

WE EXCEL

COMMENT

Open Access



Gene name errors are widespread in the scientific literature

Mark Ziemann¹, Yotam Eren^{1,2} and Assam El-Osta^{1,3*}

Abstract

The spreadsheet software Microsoft Excel, when used with default settings, is known to convert gene names to dates and floating-point numbers. A programmatic scan of leading genomics journals reveals that approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions.

Keywords: Microsoft Excel, Gene symbol, Supplementary data

Abbreviations: GEO, Gene Expression Omnibus; JIF, journal impact factor

The problem of Excel software (Microsoft Corp., Redmond, WA, USA) inadvertently converting gene symbols to dates and floating-point numbers was originally described in 2004 [1]. For example, gene symbols such as *SEPT2* (Septin 2) and *MARCH1* [Membrane-Associated Ring Finger (C3HC4) 1, E3 Ubiquitin Protein Ligase] are converted by default to ‘2-Sep’ and ‘1-Mar’, respectively. Furthermore, RIKEN identifiers were described to be automatically converted to floating point numbers (i.e. from accession ‘2310009E13’ to ‘2.31E+13’). Since that report, we have uncovered further instances where

frequently reused. Our aim here is to raise awareness of the problem.

We downloaded and screened supplementary files from 18 journals published between 2005 and 2015 using a suite of shell scripts. Excel files (.xls and .xlsx suffixes) were converted to tabular separated files (tsv) with *ssconvert* (v1.12.9). Each sheet within the Excel file was converted to a separate tsv file. Each column of data in the tsv file was screened for the presence of gene symbols. If the first 20 rows of a column contained five or more gene symbols, then it was suspected to be a list of gene symbols, and then a regular expression (regex) search of the entire column was applied to identify gene symbol errors. Official gene symbols from Ensembl version 82, accessed November 2015, were obtained for *Arabidopsis thaliana*, *Caenorhabditis elegans*, *Drosophila melanogaster*, *Danio rerio*, *Escherichia coli*, *Gallus gallus*, *Homo sapiens*, *Mus musculus*, *Oryza sativa* and *Saccharomyces cerevisiae* [2]. The regex search used was similar to that described previously by Zeeberg and colleagues [1], with the added screen for dates in other formats (e.g. DD/MM/YY and MM-DD-YY). To expedite analysis of supplementary files from multi-disciplinary journals, we limited the articles screened to those that have the keyword ‘genome’ in the title or abstract (*Science*, *Nature* and *PLoS One*). Excel files (.xls and .xlsx) deposited in NCBI Gene Expression Omnibus (GEO) [3] were also

Table 1 Results of the systematic screen of supplementary Excel files for gene name conversion errors

Journal ^a	Number of Excel files screened	Number of gene lists found	Number of papers with gene lists	Number of supplementary files affected	Number of papers affected	Number of gene names converted
<i>PLoS One</i>	7783	2202	994	220	170	4240
<i>BMC Genomics</i>	11464	1650	801	218	158	4932
<i>Genome Res</i>	2607	580	251	114	68	3180
<i>Nucleic Acids Res</i>	2117	540	315	88	67	1661
<i>Genome Biol</i>	2678	664	257	97	63	1878
<i>Genes Dev</i>	932	395	190	75	55	1593
<i>Hum Mol Genet</i>	980	372	168	48	27	1724
<i>Nature</i>	482	150	74	27	23	1375
<i>BMC Bioinformatics</i>	1790	235	152	26	21	534
<i>RNA</i>	569	127	77	20	15	1341
<i>Nat Genet</i>	264	70	37	12	9	178
<i>Bioinformatics</i>	731	112	67	11	6	339
<i>PLoS Comput Biol</i>	177	79	32	6	6	46
<i>PLoS Biol</i>	143	54	29	7	5	206
<i>Mol Biol Evol</i>	995	112	79	7	4	56
<i>Science</i>	172	36	19	7	3	451
<i>Genome Biol Evol</i>	490	32	25	2	2	121
<i>DNA Res</i>	801	57	30	2	2	6
<i>Total</i>	35175	7467	3597	987	704	23861

^aThe 18 journals investigated are ordered by the number of papers affected by gene name conversion errors