بحث حاسوبي يمكن إعادة نتائجه: عشر قواعد عملية

Ten Simple Rules for Reproducible Computational Research

غير كيتيل ساندفي، أنطون نيكروتينكو، جيمس تايلور، إيفيند هوفيغ



الملخص:

يقدم المقال عشر قواعد لتنظيم البحث الحاسوبي بحيث يمكن إعادة إنتاج نتائجه من البيانات الأصلية. وتشمل توثيق خطوات التحليل والبرامج ومعاملاتها، وتجنب المعالجة اليدوية، وحفظ إصدارات البرمجيات والشيفرات والنتائج الوسيطة، وتسجيل القيم الابتدائية لمولدات الأعداد العشوائية، وحفظ البيانات التي تستند إليها الرسوم. كما يوضح أهمية تنظيم النتائج في مستويات تفصيلية، وربط التفسيرات النصية بالأدلة، وإتاحة البيانات والشيفرات للآخرين. ويبيّن أن هذه الممارسات تعزز الشفافية والثقة، وتيسّر إعادة استخدام العمل وتوفر وقت الباحث على المدى البعيد.

الكلمات المفتاحية: قابلية إعادة الإنتاج، البحث الحاسوبي، سير العمل التحليلي، التحكم في الإصدارات، مشاركة البيانات والشيفرات

التكرار (replication) هو حجر الزاوية في العلم التراكمي [1]. غير أن الأدوات والتقنيات الجديدة، والكميات الهائلة من البيانات، والمقاربات متعددة التخصصات، وتعقّد الأسئلة المطروحة، كلها تُعقّد جهود التكرار، شأنها في ذلك شأن الضغوط المتزايدة على العلماء لدفع أبحاثهم إلى الأمام [2]. ولما كان التكرار الكامل للدراسات على بيانات جُمعت جمعًا مستقلًا غير ممكن في الغالب، فقد ظهرت مؤخرًا دعوة إلى البحث القابل لإعادة الإنتاج (reproducible research) بوصفه معيارًا أدنى قابلًا للتحقيق لتقييم قيمة الادعاءات العلمية [3]. ويتطلب ذلك أن تصف أوراق العلوم التجريبية النتائج، وأن تقدّم بروتوكولًا واضحًا وضوحًا كافيًا يتيح تكرار التحليلات المبنية على البيانات الأصلية وتوسيعها بنجاح [4].

وقد تجلّت أهمية التكرار وقابلية إعادة الإنتاج (reproducibility) مؤخرًا من خلال دراسات بيّنت أن الأوراق العلمية كثيرًا ما تُغفل تفاصيل تجريبية لا غنى عنها لإعادة الإنتاج [5]، ودراسات أظهرت صعوبات في تكرار نتائج تجريبية منشورة [6]، ومن خلال تزايد الأوراق المسحوبة [7]، والعدد الكبير من التجارب السريرية الفاشلة [8]، [9]. وقد

أفضى ذلك إلى نقاشات حول الكيفية التي يمكن بها للباحثين الأفراد والمؤسسات وجهات التمويل والمجلات أن ترسي إجراءات روتينية تزيد الشفافية وقابلية إعادة الإنتاج. وتعزيزًا لهذه الجوانب، طُرح أن المجتمع العلمي يحتاج إلى تطوير «ثقافة لإعادة الإنتاج» في العلوم الحاسوبية، وإلى اشتراطها في الادعاءات المنشورة [3].

ونودّ التأكيد على أن قابلية إعادة الإنتاج ليست مسؤولية أخلاقية تجاه المجال العلمي فحسب، بل إن غيابها قد يكون عبئًا عليك أيضًا بوصفك باحثًا فردًا. فعلى سبيل المثال، تُعدّ الممارسة الجيدة لإعادة الإنتاج ضرورية لكي يمكن تطبيق منهجية طُوّرت سابقًا تطبيقًا فعّالًا على بيانات جديدة، أو لكي يمكن إعادة استخدام الشيفرات والنتائج في مشروعات جديدة. وبعبارة أخرى، قد يتبيّن أن العادات الجيدة في إعادة الإنتاج توفّر الوقت فعلًا على المدى البعيد.

ونلاحظ كذلك أن قابلية إعادة الإنتاج تتعلق بالعادات التي تضمن بحثًا قابلًا لإعادة الإنتاج بقدر ما تتعلق بالتقنيات التي تجعل هذه العمليات فعّالة وواقعية. وتتناول كل قاعدة من القواعد العشر الآتية جانبًا محددًا من إعادة الإنتاج، وتناقش ما يلزم من حيث التعامل مع المعلومات وتتبّع الإجراءات. فإن كنت تتبع نهجًا يعتمد على أدوات منفصلة لتحليل بيانات المعلوماتية الحيوية، أي تشغيل نصوص برمجية مخصصة متنوعة من سطر الأوامر، فستحتاج على الأرجح إلى التعامل مع كل قاعدة تعاملًا صريحًا. أما إن كنت تُجري تحليلاتك عبر إطار عمل متكامل (مثل GenePattern) [10]، أو (Galaxy) [11]، أو (LONI pipeline) [12]، أو (Taverna) [13])، فقد يوفّر النظام بالفعل دعمًا كاملًا أو جزئيًا لمعظم القواعد. وكل ما يلزمك حينئذٍ هو معرفة كيفية استغلال هذه الإمكانات القائمة.

وفي الممارسة العملية، مع ضغوط النشر والمواعيد النهائية، قد يجد المرء نفسه مضطرًا إلى الموازنة بين مُثُل إعادة الإنتاج والحاجة إلى إخراج البحث ما دام لا يزال ذا أهمية. وتزداد أهمية هذه الموازنة إذا أخذنا في الحسبان أن جزءًا كبيرًا من التحليلات التي تُجرّب لا ينتهي أبدًا إلى نتائج. ومع ذلك، كثيرًا ما يتأمل المرء، بحكمة النظر إلى الوراء، الفرصة الضائعة لضمان إعادة الإنتاج، إذ قد يكون الأوان قد فات لتدوين الملاحظات اللازمة من الذاكرة (أو يكون ذلك على الأقل أصعب كثيرًا من تدوينها أثناء العمل). ونعتقد أن مكاسب إعادة الإنتاج ستعوّض عن احتمال إنفاق وقت ثمين في توثيق تحليلات يتبين لاحقًا أنها لا تفضي إلى نتائج.

وكحدّ أدنى، ينبغي أن تكون قادرًا على الأقل على إعادة إنتاج النتائج بنفسك. فهذا يفي بأبسط متطلبات البحث السليم، ويتيح مواجهة أي تشكيك جوهري في البحث مستقبلًا بتفسير دقيق. ومع أن هذا قد يبدو متطلبًا ضعيفًا جدًا، فإن هذا المستوى من إعادة الإنتاج نفسه كثيرًا ما يتطلب قدرًا من العناية لتحقيقه. إذ يوجد لأي تحليل عدد من التوليفات الممكنة يتزايد أُسّيًا لإصدارات البرامج، وقيم المعاملات، وخطوات المعالجة المسبقة، وما إلى ذلك، مما يعني أن الإخفاق في تدوين الملاحظات قد يجعل إعادة الإنتاج الدقيقة مستحيلة في جوهرها.

وبعد تحقيق هذا الحد الأساسي، يمكن التقدم إلى مستويات أفضل. فبدلًا من الاقتصار على القدرة على إعادة النتائج عند الضرورة، يمكنك تنظيم العمل السابق بحيث يسهل استخدامه باستمرار لرفع إنتاجيتك. ويمكنك كذلك تمكين زملائك من إعادة إنتاج النتائج عمليًا، مما قد يعزز الثقة بعملك والاهتمام به والاستشهاد به [6]، [14].

ونعرض هنا عشر قواعد بسيطة لقابلية إعادة إنتاج البحث الحاسوبي. ويمكن أن تكون هذه القواعد في متناولك متى أردت أن تجعل بحثك أيسر وصولًا، سواء أكان ذلك للأقران أم لنفسك في المستقبل.

القاعدة 1: تتبّع طريقة إنتاج كل نتيجة:

متى أمكن أن تكون لنتيجة ما أهمية محتملة، فتتبّع طريقة إنتاجها. وعند القيام بذلك، سيجد المرء كثيرًا أن الانتقال من البيانات الخام إلى النتيجة النهائية ينطوي على خطوات كثيرة مترابطة (أوامر مفردة، ونصوص برمجية، وبرامج). ونسمّي هذا التسلسل من الخطوات، سواء أكان مؤتمتًا أم يُنفّذ يدويًا، سيرَ عمل تحليليًا (analysis workflow). ومع أن الجزء الجوهري من التحليل كثيرًا ما تمثله خطوة واحدة فقط من الخطوات، فإن التسلسل الكامل لخطوات المعالجة المسبقة واللاحقة كثيرًا ما يكون حاسمًا للوصول إلى النتيجة المتحققة. ولكل خطوة مشمولة، ينبغي أن تضمن تسجيل كل تفصيل قد يؤثر في تنفيذها. وإذا كانت الخطوة تُنفّذ ببرنامج حاسوبي، فإن التفاصيل الحاسمة تشمل اسم البرنامج وإصداره، إضافة إلى المعاملات والمدخلات الدقيقة التي استُخدمت.

ومع أن التدوين اليدوي للتسلسل الدقيق للخطوات المتخذة يتيح إعادة إنتاج التحليل، فإن التوثيق قد يفقد تزامنه بسهولة مع الطريقة التي أُجري بها التحليل فعلًا في نسخته النهائية. أما بتحديد سير العمل التحليلي كاملًا في صيغة تسمح بالتنفيذ المباشر، فيمكن للمرء أن يضمن تطابق الوصف مع التحليل الذي أُجري (لاحقًا)، وأن يمكن إعادة إنتاج التحليل بطريقة مؤتمتة على يدك أو على يد غيرك. وقد تأتي هذه الأوصاف القابلة للتنفيذ [10] في صورة نصوص برمجية لمفسّر الأوامر (shell scripts) بسيطة أو ملفات (makefile) [15]، [16] في سطر الأوامر، أو في صورة مسارات عمل مخزنة في نظام لإدارة سير العمل [10]، [11]، [13]، [17]، [18].

وكحدّ أدنى، ينبغي أن تسجّل على الأقل تفاصيل كافية عن البرامج والمعاملات والإجراءات اليدوية تتيح لك، بعد سنة أو نحوها، إعادة إنتاج النتائج على وجه التقريب.

القاعدة 2: تجنّب خطوات المعالجة اليدوية للبيانات:

اعتمد، متى أمكن، على تنفيذ البرامج بدلًا من الإجراءات اليدوية لتعديل البيانات. فهذه الإجراءات اليدوية ليست غير فعّالة ومعرّضة للأخطاء فحسب، بل يصعب كذلك إعادة إنتاجها. وإذا كنت تعمل في سطر أوامر يونكس (UNIX)، فيمكن عادةً الاستعاضة عن التعديل اليدوي للملفات باستخدام أوامر يونكس القياسية أو نصوص برمجية مخصصة صغيرة. وإذا كنت تعمل بأطر عمل متكاملة، فستتوافر عادةً مجموعة غنية جدًا من المكونات لمعالجة البيانات. فعلى سبيل المثال، ينبغي الاستعاضة عن التعديل اليدوي لملفات البيانات لتحقيق التوافق في الصيغ بمحوّلات صيغ يمكن إعادة تشغيلها وإدراجها في مسارات عمل قابلة للتنفيذ. كما ينبغي تجنّب العمليات اليدوية الأخرى، مثل استخدام النسخ واللصق بين المستندات. وإذا تعذّر تجنّب العمليات اليدوية، فينبغي لك كحدّ أدنى أن تدوّن ملفات البيانات التي عُدّلت أو نُقلت، والغرض من ذلك.

القاعدة 3: أرشِف الإصدارات الدقيقة لجميع البرامج الخارجية المستخدمة:

لإعادة إنتاج نتيجة ما إعادةً دقيقة، قد يكون من الضروري استخدام البرامج بالإصدارات الدقيقة التي استُخدمت في الأصل. كما أن صيغ المدخلات والمخرجات قد تتغير بين الإصدارات، ومن ثم قد لا يعمل إصدار أحدث من برنامج ما أصلًا دون تعديل مدخلاته. وحتى مع تدوين الإصدار المستخدم من برنامج ما، فليس من اليسير دائمًا الحصول على البرنامج بأي إصدار غير الإصدار الحالي. ولذلك قد توفّر أرشفة الإصدارات الدقيقة من البرامج المستخدمة فعلًا كثيرًا من العناء في المراحل اللاحقة. وفي بعض الحالات، كل ما يلزم هو تخزين ملف تنفيذي واحد أو ملف شيفرة مصدرية واحد. وفي حالات أخرى، قد تكون لبرنامج ما بدوره متطلبات محددة من برامج أو حزم أخرى مثبتة، أو

اعتماديات على مكونات محددة من نظام التشغيل. ولضمان الإتاحة مستقبلًا، قد يكون الحل الوحيد القابل للتطبيق حينئذٍ هو تخزين صورة كاملة لآلة افتراضية تضم نظام التشغيل والبرنامج. وكحدّ أدنى، ينبغي أن تدوّن الأسماء والإصدارات الدقيقة للبرامج الرئيسة التي تستخدمها.

القاعدة 4: أخضِع جميع النصوص البرمجية المخصصة للتحكم في الإصدارات:

إن أدنى تغيير في برنامج حاسوبي قد تترتب عليه عواقب كبيرة مقصودة أو غير مقصودة. فحين تُستخدم شيفرة يجري تطويرها باستمرار (وهي عادةً نص برمجي صغير) لتوليد نتيجة معينة، فقد لا تستطيع إنتاج ذلك المخرج بعينه إلا تلك الحالة الدقيقة من النص البرمجي، حتى مع البيانات المدخلة والمعاملات نفسها. وكما نوقش أيضًا في القاعدتين 3 و6، قد تكون إعادة الإنتاج الدقيقة للنتائج أساسية في حالات معينة. وإذا لم تُؤرشف الشيفرة الحاسوبية أرشفة منهجية على امتداد تطورها، فقد يكون الرجوع إلى حالة الشيفرة التي أعطت نتيجة معينة مهمة ميؤوسًا منها. وقد يلقي ذلك بظلال من الشك على النتائج السابقة، إذ قد يستحيل معرفة ما إذا كانت ناتجة جزئيًا عن خلل برمجي أو عن سلوك غير مرغوب فيه بطريقة أخرى.

والحل القياسي لتتبّع تطور الشيفرة هو استخدام نظام للتحكم في الإصدارات (version control system) [15]، مثل (Subversion) أو (Git) أو (Mercurial). وهذه الأنظمة سهلة الإعداد والاستخدام نسبيًا، ويمكن استخدامها لتخزين حالة الشيفرة تخزينًا منهجيًا طوال التطوير بأي درجة زمنية مرغوبة من الدقة.

وكحدّ أدنى، ينبغي أن تؤرشف نسخًا من نصوصك البرمجية من حين لآخر، لكي تحتفظ بسجلّ تقريبي للحالات المختلفة التي مرّت بها الشيفرة أثناء التطوير.

القاعدة 5: سجّل جميع النتائج الوسيطة، وبصيغ معيارية متى أمكن:

من حيث المبدأ، ما دامت العملية الكاملة المستخدمة في إنتاج نتيجة ما متتبَّعة، فيمكن أيضًا توليد جميع البيانات الوسيطة من جديد. أما من الناحية العملية، فإن توافر النتائج الوسيطة بسهولة قد يكون ذا قيمة كبيرة. فالتصفح السريع للنتائج الوسيطة قد يكشف عن تباينات مع ما هو مفترض، ويمكنه بذلك أن يكشف أخطاء برمجية أو تفسيرات خاطئة لا تظهر في النتائج النهائية. وثانيًا، فهو يكشف على نحو أكثر مباشرة عواقب البرامج البديلة واختيارات المعاملات في الخطوات المفردة. وثالثًا، حين لا تكون العملية الكاملة قابلة للتنفيذ بسهولة، فإنه يتيح إعادة تشغيل أجزاء منها. ورابعًا، عند إعادة إنتاج النتائج، يتيح تتبّع أي تناقضات تُصادف إلى الخطوات التي تنشأ فيها المشكلات. وخامسًا، يتيح الفحص النقدي للعملية الكاملة وراء نتيجة ما، دون الحاجة إلى أن تكون جميع الملفات التنفيذية جاهزة للعمل. واحفظ هذه النتائج الوسيطة بصيغ معيارية متى أمكن. وكحدّ أدنى، أرشِف أي ملفات نتائج وسيطة تُنتج عند تشغيل تحليل ما (ما لم تكن متطلبات التخزين مرتفعة التكلفة).

القاعدة 6: في التحليلات التي تتضمن العشوائية، دوّن البذور العشوائية المستخدمة:

تتضمن تحليلات وتنبؤات كثيرة عنصرًا من العشوائية، مما يعني أن البرنامج نفسه سيعطي عادةً نتائج مختلفة اختلافًا طفيفًا في كل مرة يُنفّذ فيها (حتى عند تلقّي مدخلات ومعاملات متطابقة). غير أنه مع البذرة الابتدائية نفسها، ستكون جميع الأعداد العشوائية المستخدمة في التحليل متساوية، مما يعطي نتائج متطابقة في كل مرة يُشغّل فيها. وثمة فرق كبير بين ملاحظة أن نتيجة ما قد أُعيد إنتاجها إعادة دقيقة أو تقريبية فحسب. فبينما يُعدّ

الحصول على نتائج متساوية مؤشرًا قويًا على أن الإجراء قد أُعيد إنتاجه بدقة، فكثيرًا ما يصعب استنتاج أي شيء عند الحصول على نتائج متساوية تقريبًا فقط. وفي التحليلات التي تنطوي على أعداد عشوائية، يعني هذا أنه ينبغي تسجيل البذرة العشوائية (random seed). فذلك يتيح إعادة إنتاج النتائج بدقة بتزويد مولّد الأعداد العشوائية بالبذرة نفسها في عمليات التشغيل المستقبلية. وكحدّ أدنى، ينبغي أن تدوّن خطوات التحليل التي تنطوي على العشوائية، لكي يمكن توقع قدر معين من التباين عند إعادة إنتاج النتائج.

القاعدة 7: احفظ دائمًا البيانات الخام التي تقوم عليها الرسوم البيانية:

قد يتغير الرسم البياني مرات عدة بين إعداده الأول ونشره، سواء لتحسين وضوحه أو لتوحيد مظهره مع بقية الرسوم. وإذا حُفظت البيانات الخام التي يستند إليها بطريقة منظمة يسهل استرجاعها، أمكن تعديل أوامر الرسم بدلًا من إعادة التحليل كاملًا. ويتيح ذلك أيضًا الرجوع إلى القيم الدقيقة متى لزم الأمر. وإذا تضمّن الرسم معالجة إضافية للبيانات، فمن المفيد حفظ البيانات الأصلية والقيم المعالجة معًا؛ ففي المدرج التكراري مثلًا، يمكن حفظ القيم قبل تجميعها في فئات، وعدد القيم في كل فئة، وهو ما يحدد ارتفاع الأعمدة. وعند استخدام نظام قائم على الأوامر مثل (R)، احفظ الشيفرة التي أنشأت الرسم لتتمكن من تعديلها بسهولة. والحد الأدنى هو توثيق البيانات التي بُني عليها كل رسم وكيفية إعادة تكوينها.

القاعدة 8: نظّم المخرجات في مستويات تتيح الانتقال إلى التفاصيل:

كثيرًا ما تمثّل النتائج النهائية التي تصل إلى المقال، سواء أكانت رسومًا بيانية أم جداول، بيانات ملخّصة تلخيصًا شديدًا. فعلى سبيل المثال، قد تمثّل كل قيمة على امتداد منحنى ما بدورها متوسطات مأخوذة من توزيع أساسي. وللتحقق من النتيجة الرئيسة وفهمها فهمًا كاملًا، كثيرًا ما يكون من المفيد فحص القيم التفصيلية التي تقوم عليها الملخصات. ومن الطرق الشائعة لكن غير العملية للقيام بذلك إدراج مخرجات تصحيح متنوعة في الشيفرة المصدرية للنصوص البرمجية والبرامج. وحين يسمح سياق التخزين، فمن الأفضل ببساطة إدراج مخرجات دائمة لجميع البيانات الأساسية عند توليد نتيجة رئيسة، باستخدام اصطلاح منهجي للتسمية يتيح العثور بسهولة على البيانات الكاملة التي تقوم عليها قيمة ملخّصة معينة. ونجد أن النص الفائق (أي المخرجات في صورة ملفات (html)) مفيد على نحو خاص لهذا الغرض. فهو يتيح توليد النتائج الملخّصة مصحوبة بروابط يمكن تتبّعها بيسر كبير (بنقرة بسيطة) وصولًا إلى البيانات الكاملة التي تقوم عليها كل قيمة ملخّصة. وعند التعامل مع نتائج ملخّصة، ينبغي لك كحدّ أدنى أن تولّد القيم التفصيلية التي تقوم عليها الملخصات وتفحصها وتتحقق منها مرة واحدة على الأقل.

القاعدة 9: اربط العبارات النصية بالنتائج التي تستند إليها:

على امتداد مشروع بحثي نموذجي، تُجرّب مجموعة من التحليلات المختلفة وتُفسَّر نتائجها. ومع أن نتائج التحليلات وتفسيراتها النصية المقابلة مترابطة بوضوح على المستوى المفاهيمي، فإنها غالبًا ما تُحفظ كل منها بمعزل عن الأخرى: فالنتائج تُحفظ عادةً في مساحة لتخزين البيانات على خادم أو حاسوب شخصي، بينما تُحفظ التفسيرات في مستندات نصية في صورة ملاحظات شخصية أو رسائل بريد إلكتروني إلى المتعاونين. وليست هذه التفسيرات النصية عمومًا مجرد إعادة وصف للنتائج؛ إذ كثيرًا ما تنطوي على النظر إلى النتائج في ضوء نظريات ونتائج أخرى. وهي بذلك تحمل معلومات إضافية، بينما تستند في الوقت نفسه إلى نتيجة محددة.

وإذا أردت إعادة تقييم تفسيراتك السابقة، أو إتاحة الفرصة للأقران لإجراء تقييمهم الخاص للادعاءات التي تطرحها في ورقة علمية، فسيتعين عليك ربط عبارة نصية معينة (تفسير، أو ادعاء، أو استنتاج) بالنتائج الدقيقة التي تقوم عليها العبارة. وقد يكون إجراء هذا الربط عند الحاجة إليه صعبًا ومعرّضًا للأخطاء، إذ قد يصعب تحديد النتيجة الدقيقة التي تقوم عليها العبارة وتدعمها من بين مجموعة كبيرة من التحليلات المختلفة ذات الإصدارات المتنوعة.

ولإتاحة الاسترجاع الفعّال للتفاصيل الكامنة وراء العبارات النصية، نقترح أن تُربط العبارات بالنتائج التي تستند إليها منذ وقت صياغتها الأولى (في الملاحظات أو رسائل البريد الإلكتروني مثلًا). ويمكن أن يكون هذا الربط على سبيل المثال مسارًا بسيطًا لملف يضم النتائج التفصيلية، أو معرّف نتيجة في إطار تحليلي، يُدرج في النص نفسه. ولتكامل أوثق، تتوافر أدوات تساعد على دمج التحليلات القابلة لإعادة الإنتاج مباشرة في المستندات النصية، مثل (Sweave) [19]، والوظيفة الإضافية (GenePattern) لبرنامج (Word) [4]، و(Galaxy Pages) [20]. ويمكن استخدام هذه الحلول لاحقًا أيضًا فيما يتصل بالمنشورات، كما سيُناقش في القاعدة التالية.

وكحدّ أدنى، ينبغي أن تقدّم مع تفسيراتك النصية تفاصيل كافية تتيح تتبّع النتائج الدقيقة التي تقوم عليها، أو بعض النتائج ذات الصلة على الأقل، في المستقبل.

القاعدة 10: أتِح الشيفرات وخطوات التشغيل والنتائج للجمهور:

وأخيرًا وليس آخرًا، ينبغي إتاحة جميع البيانات المدخلة والنصوص البرمجية والإصدارات والمعاملات والنتائج الوسيطة للعموم وبسهولة. وقد أصبحت تتوافر الآن حلول متنوعة تجعل مشاركة البيانات أيسر وأكثر معيارية وإتاحة في مجالات بعينها، كما في بيانات التعبير الجيني [21]-[23]. وتسمح معظم المجلات بإرفاق مواد إلكترونية تكميلية بالمقالات، وقد بدأت بعض المجلات جهودًا إضافية لجعل البيانات والشيفرات أكثر تكاملًا مع المنشورات [3]، [24]. وكحدّ أدنى، ينبغي أن تقدّم البيانات الرئيسة والشيفرة المصدرية بوصفها مواد تكميلية، وأن تكون مستعدًا للرد على أي طلبات من الأقران للحصول على مزيد من البيانات أو التفاصيل المنهجية.

إن جعل إعادة إنتاج عملك على يد الأقران إمكانية واقعية يرسل إشارة قوية على الجودة والجدارة بالثقة والشفافية. وقد يزيد ذلك من جودة عملية مراجعة عملك وسرعتها، ومن فرص نشره، ومن فرص أن يبني عليه باحثون آخرون ويستشهدوا به بعد نشره [25].

المراجع:

1. Crocker J, Cooper ML (2011) Addressing scientific fraud. Science 334: 1182.
2. Jasny BR, Chin G, Chong L, Vignieri S (2011) Data replication & reproducibility. Again, and again, and again.... Introduction. Science 334: 1225.
3. Peng RD (2011) Reproducible research in computational science. Science 334: 1226-1227.
4. Mesirov JP (2010) Computer science. Accessible reproducible research. Science 327: 415-416.

5. Nekrutenko A, Taylor J (2012) Next-generation sequencing data interpretation: enhancing reproducibility and accessibility. Nat Rev Genet 13: 667-672.

6. Ioannidis JP, Allison DB, Ball CA, Coulibaly I, Cui X, et al. (2009) Repeatability of published microarray gene expression analyses. Nat Genet 41: 149-155.

7. Steen RG (2011) Retractions in the scientific literature: is the incidence of research fraud increasing? J Med Ethics 37: 249-253.

8. Prinz F, Schlange T, Asadullah K (2011) Believe it or not: how much can we rely on published data on potential drug targets? Nat Rev Drug Discov 10: 712.

9. Begley CG, Ellis LM (2012) Drug development: raise standards for preclinical cancer research. Nature 483: 531-533.

10. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, et al. (2006) GenePattern 2.0. Nat Genet 38: 500-501.

11. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, et al. (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15: 1451-1455.

12. Rex DE, Ma JQ, Toga AW (2003) The LONI Pipeline Processing Environment. Neuroimage 19: 1033-1048.

13. Oinn T, Addis M, Ferris J, Marvin D, Senger M, et al. (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20: 3045-3054.

14. Piwowar HA, Day RS, Fridsma DB (2007) Sharing detailed research data is associated with increased citation rate. PLoS ONE 2: e308

15. Heroux MA, Willenbring JM (2009) Barely sufficient software engineering: 10 practices to improve your cse software. In: 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering. pp. 15-21.

16. Schwab M, Karrenbach M, Claerbout J (2000) Making scientific computations reproducible. Comput Sci Eng 2: 61-67.

17. Goble CA, Bhagat J, Aleksejevs S, Cruickshank D, Michaelides D, et al. (2010) myExperiment: a repository and social network for the sharing of bioinformatics workflows. Nucleic Acids Res 38: W677-682.

18. Deelman E, Singh G, Su M-H, Blythe J, Gil Y, et al. (2005) Pegasus: a framework for mapping complex scientific workflows onto distributed systems. Scientific Programming Journal 13: 219-237.

19. Leisch F (2002) Sweave: dynamic generation of statistical reports using literate data analysis. In: Härdle W, Rönz B, editors. Compstat: proceedings in computational statistics. Heidelberg, Germany: Physika Verlag. pp. 575–580.

20. Goecks J, Nekrutenko A, Taylor J (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11: R86.

21. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, et al. (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat Genet 29: 365–371.

22. Brazma A, Parkinson H, Sarkans U, Shojatalab M, Vilo J, et al. (2003) ArrayExpress–a public repository for microarray gene expression data at the EBI. Nucleic Acids Res 31: 68–71.

23. Edgar R, Domrachev M, Lash AE (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res 30: 207–210.

24. Sneddon TP, Li P, Edmunds SC (2012) GigaDB: announcing the GigaScience database. Gigascience 1: 11.

25. Prlić A, Procter JB (2012) Ten simple rules for the open development of scientific software. PLoS Comput Biol 8: e1002802